

Network Working Group  
Internet Draft  
Intended status: Informational  
Expires: December 12, 2012

Maria Napierala  
AT&T  
Luyuan Fang  
Dennis Cai  
Cisco Systems

June 12, 2012

IP-VPN Data Center Problem Statement and Requirements  
draft-fang-vpn4dc-problem-statement-01.txt

Abstract

Network Service Providers commonly use BGP/MPLS VPNs [RFC 4364] as the control plane for virtual networks. This technology has proven to scale to a large number of VPNs and attachment points, and it is well suited for Data Center connectivity, especially when supporting all IP applications.

The Data Center environment presents new challenges and imposes additional requirements to IP VPN technologies, including multi-tenancy support, high scalability, VM mobility, security, and orchestration. This document describes the problems and defines the new requirements.

Status of this Memo

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

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

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

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

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

Napierala, Fang, Cai    Expire December 12, 2012

[Page 1]

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1.	Introduction	3
2.	Terminology	4
3.	IP-VPN in Data Center Network	4
3.1.	Data Center Connectivity Scenarios	5
4.	Data Center Virtualization Requirements	6
5.	Decoupling of Virtualized Networking from Physical Infrastructure	6
6.	Encapsulation/Decapsulation Device for Virtual Network Payloads	7
7.	Decoupling of Layer 3 Virtualization from Layer 2 Topology	8
8.	Requirements for Optimal Forwarding of Data Center Traffic	9
9.	Virtual Network Provisioning Requirements	9
10.	Application of BGP/MPLS VPN Technology to Data Center Network	10
10.1.	Data Center Transport Network	12
10.2.	BGP Requirements in a Data Center Environment	12
11.	Virtual Machine Migration Requirement	14
12.	IP-VPN Data Center Use Case: Virtualization of Mobile Network	15
13.	Security Considerations	17
14.	IANA Considerations	17
15.	Normative References	17
16.	Informative References	17
17.	Authors' Addresses	17
18.	Acknowledgements	18

## Requirements Language

Although this document is not a protocol specification, 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 [RFC 2119].

## 1. Introduction

Data Centers are increasingly being consolidated and outsourced in an effort, both to improve the deployment time of applications as well as reduce operational costs. This coincides with an increasing demand for compute, storage, and network resources from applications. In order to scale compute, storage, and network resources, physical resources are being abstracted from their logical representation. This is referred as server, storage, and network virtualization. Virtualization can be implemented in various layers of computer systems or networks.

The compute loads of many different customers are executed over a common infrastructure. Compute nodes are often executed as Virtual Machines in an "Infrastructure as a Service" (IaaS) Data Center. The set of virtual machines corresponding to a particular customer should be constrained to a private network.

New network requirements are presented due to the consolidation and virtualization of Data Center resources, public, private, or hybrid. Large scale server virtualization (i.e., IaaS) requires scalable and robust Layer 3 network support. It also requires scalable local and global load balancing. This creates several new problems for network connectivity, namely elasticity, location independence (referred to also as Virtual Machine mobility), and extremely large number of virtual resources.

In the Data Center networks, the VMs of a specific customer or application are often configured to belong to the same IP subnet. Many solutions proposed for large Data Center networks rely on the assumption that the layer-2 inter-server connectivity is required, especially to support VM mobility within a virtual IP subnet. Given that VM mobility consists in moving VMs anywhere within (and even across) Data Centers, the virtual subnet locality associated with small scale deployments cannot be preserved. A Data Center solution should not prevent grouping of virtual resources into IP subnets but the virtual subnets have no benefits of locality across a large data-center.

While some applications may expect to find other peers in a particular user defined IP subnet, this does not imply the need to provide a Layer 2 service that preserves MAC addresses. A network virtualization solution should be able to provide IP unicast connectivity between hosts in the same and different subnets without any assumptions regarding the underlying media layer. A solution should also be able to provide a multicast service that implements IP subnet broadcast as well as IP multicast.

One of the main goals in designing a Data Center network is to minimize the cost and complexity of its core/"fabric" network. The cost and complexity of Data Center network is a function of the number of virtualized resources, that is, the number of "closed user-groups". Data Centers use VPNs to isolate compute resources associated with a specific "closed user-group". Some use VLANs as a VPN technology, others use Layer 3 based solutions often with proprietary control planes. Service Providers are interested in interoperability and in openly documented protocols rather than in proprietary solutions.

## 2. Terminology

AS	Autonomous Systems
DC	Data Center
DCI	Data Center Interconnect
EPC	Evolved Packet Core
End-System	A device where Guest OS and Host OS/Hypervisor reside
IaaS	Infrastructure as a Service
LTE	Long Term Evolution
PCEF	Policy Charging and Enforcement Function
RT	Route Target
ToR	Top-of-Rack switch
VM	Virtual Machine
Hypervisor	Virtual Machine Manager
SDN	Software Defined Network
VPN	Virtual Private Network

## 3. IP-VPN in Data Center Network

In this document, we define the problem statement and requirements for Data Center connectivity based on the assumption that applications require IP connectivity but no Layer 2 direct adjacencies. Applications do not send or receive Ethernet frames directly. They are restricted to IP services due to several reasons such as privileges, address discovery, portability, APIs, etc. IP service can be unicast, VPN broadcast, or multicast.

An IP-VPN DC solution is meant to address IP-only Data Center, defined by a Data Center where VMs, applications, and appliances require only IP connectivity and the underlying DC core infrastructure is IP only. Non-IP applications are addressed by other solutions and are not in scope of this document.

It is also assumed that both IPv4 and IPv6 unicast communication is to be supported. Furthermore, the multicast transmission, i.e., allowing IP applications to send packets to a group of IP addresses should also be supported. The most typical multicast applications are service, network, device discovery applications and content

distribution. While there are simpler and more effective ways to provide discovery services or reliable content delivery, a Data Center solution should support multicast transmission to applications. A Data Center solution should cover the case where the Data Center transport network does not support IP multicast transmission service.

The Data Center multicast service should also support a delivery of traffic to all endpoints of a given VPN even if those endpoints have not sent any control messages indicating the need to receive that traffic. In other words, the multicast service should be capable of delivering the IP broadcast traffic in a virtual topology.

### 3.1. Data Center Connectivity Scenarios

There are three different cases of Data Center (DC) network connectivity:

1. Intra-DC connectivity: Private network connectivity between compute resources within a public (or private) Data Center.
2. Inter-DC connectivity: Private network connectivity between different Data Centers, either public or private.
3. Client-to-DC connectivity: Connectivity between client and a private or public Data Center. The later includes interconnection between a service provider and a public Data Center (which may belong to the same or different service provider).

Private network connectivity within the Data Center requires network virtualization solution. In this document we define Layer 3 VPN requirements to Data Center network virtualization. The Layer 3 VPN technology (i.e., MPLS/BGP VPN) also applies to the interconnection of different data-centers.

When private networks interconnect with public Data Centers, the VPN provider must interconnect with the public Data Center provider. In this case we are in the presence of inter-provider VPNs. The Inter-AS MPLS/BGP VPN Options A, B, or C [RFC 4364] provide network-to-network interconnection service and they constitute the basis of SP network to public Data Center network connectivity. There might incremental improvements to the existing inter-AS solutions, pertaining to scalability and security, for example.

Service Providers can leverage their existing Layer 3 VPN services and provide private VPN access from client's branch sites to client's own private Data Center or to SP's own Data Center. The service provider-based VPN access can provide additional value compared with public internet access, such as security, QoS, OAM, and troubleshooting.

#### 4. Data Center Virtualization Requirements

Private network connection service in a Data Center must provide traffic isolation between different virtual instances that share a common physical infrastructure. A collection of compute resources dedicated to a process or application is referred to as a "closed user-group". Each "closed user-group" is a VPN in the terminology used by IP VPNs.

Any DC solution needs to assure network isolation among tenants or applications sharing the same Data Center physical resources. A DC solution should allow a VM or application end-point to belong to multiple closed user-groups/VPNs. A closed user-group should be able to communicate with other closed-user groups according to specified routing policies. A customer or tenant should be able to define multiple closed user-groups.

Typically VPNs that belong to different tenants do not communicate with each other directly but they should be allowed to access common appliances such as storage, database services, security services, etc. It is also common for tenants to deploy a VPN per "application tier" (e.g. a VPN for web front-ends and a different VPN for the logic tier). In that scenario most of the traffic crosses VPN boundaries. That is also the case when "network attached storage" (NAS) is used or when databases are deployed as-a-service.

Another reason for the Data Center network virtualization is the need to support VM move. Since the IP addresses used for communication within or between applications may be anywhere across the data-center, using a virtual topology is an effective way to solve this problem.

#### 5. Decoupling of Virtualized Networking from Physical Infrastructure

The Data Center switching infrastructure (access, aggregation, and core switches) should not maintain any information that pertains to the virtual networks. Decoupling of virtualized networking from the physical infrastructure has the following advantages: 1) provides

better scalability; 2) simplifies the design and operation; 3) reduces the cost of a Data Center network. It has been proven (in Internet and in large BGP IP VPN deployments) that moving complexity associated with virtual entities to network edge while keeping network core simple has very good scaling properties.

There should be a total separation between the virtualized segments (virtual network interfaces that are associated with VMs) and the physical network (i.e., physical interfaces that are associated with the data-center switching infrastructure). This separation should include the separation of the virtual network IP address space from the physical network IP address space. The physical infrastructure addresses should be routable in the underlying Data Center transport network, while the virtual network addresses should be routable on the VPN network only. Not only should the virtual network data plane be fully decoupled from the physical network, but its control plane should be decoupled as well. In order to decouple virtual and physical networks, the virtual networking should be treated as an "infrastructure" application. Only the solutions that meet those requirements would provide a truly scalable virtual networking.

MPLS labels provide the necessary information to implement VPNs. When crossing the Data Center infrastructure the virtual network payloads should be encapsulated in IP or GRE [RFC 4023], or native MPLS envelopes.

## 6. Encapsulation/Decapsulation Device for Virtual Network Payloads

In order to scale a virtualized Data Center infrastructure, the encapsulation (and decapsulation) of virtual network payloads should be implemented on a device as close to virtualized resources as possible. Since the hypervisors in the end-systems are the devices at the edge of a Data Center network they are the most optimal location for the VPN encap/decap functionality. Data-plane device that implements the VPN encap/decap functionality acts as the first-hop router in the virtual topology.

The IP-VPN solution for Data Center should also support deployments where it is not possible or not desirable to implement VPN encapsulation in the hypervisor/Host OS. In such deployments encap/decap functionality may be implemented in an external physical switch such as aggregation switch or top-of-rack switch. The external device implementing VPN tunneling functionality should be as close as possible to the end-system itself. The same DC solution should support deployments with both, internal (in a hypervisor) and external (outside of a hypervisor) encap/decap devices.

Whenever the VPN forwarding functionality (i.e., the data-plane device that encapsulates packets into, e.g., MPLS-over-GRE header) is implemented in an external device, the VPN service itself must be delivered to the virtual interfaces visible to the guest OS. However, the switching elements connecting the end-system to the encap/decap device should not be aware of the virtual topology. Instead, the VPN endpoint membership information might be, for example, communicated by the end-system using a signaling protocol. Furthermore, for an all-IP solution, the Layer 2 switching elements connecting the end-system to the encap/decap device should have no knowledge of the VM/application endpoints. In particular, the MAC addresses known to the guest OS should not appear on the wire.

## 7. Decoupling of Layer 3 Virtualization from Layer 2 Topology

The IP-VPN approach to Data Center network design dictates that the virtualized communication should be routed, not bridged. The Layer 3 virtualization solution should be decoupled from the Layer 2 topology. Thus, there should be no dependency on VLANs or Layer 2 broadcast.

In solutions that depend on Layer 2 broadcast domains, the VM-to-VM communication is established based on flooding and data plane MAC learning. Layer 2 MAC information has to be maintained on every switch where a given VLAN is present. Even if some solutions are able to eliminate data plane MAC learning and/or unicast flooding across Data Center core network, they still rely on VM MAC learning at the network edge and on maintaining the VM MAC addresses on every (edge) switch where the Layer 2 VPN is present.

The MAC addresses known to guest OS in end-system are not relevant to IP services and introduce unnecessary overhead. Hence, the MAC addresses associated with virtual machines should not be used in the virtual Layer 3 networks. Rather, only what is significant to IP communication, namely the IP addresses of the VMs and application endpoints should be maintained by the virtual networks. An IP-VPN solution should forwards VM traffic based on their IP addresses and not on their MAC addresses.

From a Layer 3 virtual network perspective, IP packets should reach the first-hop router in one-hop, regardless of whether the first-hop router is a hypervisor/Host OS or it is an external device. The VPN first-hop router should always perform an IP lookup on every packet it receives from a VM or an application. The first-hop router should encapsulate the packets and route them towards the destination end-system. Every IP packet should be forwarded along the shortest path towards a destination host or appliance,



regardless of whether the packet's source and destination are in the same or different subnets.

#### 8. Requirements for Optimal Forwarding of Data Center Traffic

The Data Center solutions that optimize for the maximum utilization of compute and storage resources require that those resources may be located anywhere in the data-center. The physical and logical spreading of appliances and computations implies a very significant increase in data-center infrastructure bandwidth consumption. Hence, it is important that DC solutions are efficient in terms of traffic forwarding and assure that packets traverse Data Center switching infrastructure only once. This is not possible in DC solutions where a virtual network boundary between bridging (Layer 2) and routing (Layer 3) exists anywhere within the Data Center transport network. If a VM can be placed in an arbitrary location, mixing of the Layer 2 and the Layer 3 solutions may cause the VM traffic traverse the Data Center core multiple times before reaching the destination host.

It must be also possible to send the traffic directly from one VM to another VM (within or between subnets) without traversing through a midpoint router. This is important given that most of the traffic in a Data Center is within the VPNs.

#### 9. Virtual Network Provisioning Requirements

IP-VPN DC has to provide fast and secure provisioning (with low operational complexity) of VPN connectivity for a VM within a Data Center and across Data Centers. This includes interconnecting VMs within and across physical Data Centers in the context of a virtual networking. It also includes the ability to connect a VM to a customer VPN outside the Data Center, thus requiring the ability to provision the communication path within the Data Center to the customer VPN.

The VM provisioning should be performed by an orchestration system. The orchestration system should have a notion of a closer user-group/tenant and the information about the services the tenant is allowed to access. The orchestration system should allocate an IP address to a VM. When the VM is provisioned, its IP address and the closed user-group/VPN identifier (VPN-ID) should be communicated to the host OS on the end-system. There should a centralized database system (possibly with a distributed implementation) that will contain the provisioning information regarding VPN-IDs and the services the corresponding VPNs could

access. This information should be accessible to the virtual network control plane.

The orchestration system should be able to support the specification of fine grain forwarding policies (such as filtering, redirection, rate limiting) to be injected as the traffic flow rules into the virtual network.

Common APIs can be a simple and a useful step to facilitate the provisioning processes. Authentication is required when a VM is being provisioned to join an IP VPN.

An IP-VPN Data Center networking solution should seamlessly support VM connectivity to other network devices (such as service appliances or routers) that use the traditional BGP/MPLS VPN technology.

#### 10. Application of BGP/MPLS VPN Technology to Data Center Network

BGP IP VPN technologies (based on [RFC 4364]) have proven to be able to scale to a large number of VPNs (tens of thousands) and customer routes (millions) while providing for aggregated management capability. Data Center networks could use the same transport mechanisms as used today in many Service Provider networks, specifically the MPLS/BGP VPNs that often overlay huge transport areas.

MPLS/BGP VPNs use BGP as a signaling protocol to exchange VPN routes. IP-VPN DC solution should consider that it might not be feasible to run BGP protocol on a hypervisor or external switch such as top-of-rack. This includes functions like BGP route selection and processing of routing policies, as well as handling MP-BGP structures like Route Distinguishers and Route Targets. Rather, it might be preferable to use a signaling mechanism that is more familiar and compatible with the methods used in the application software development. While network devices (such as routers and appliances) may choose to receive VPN signaling information directly via BGP, the end-systems/switches may choose other type of interface or protocol to exchange virtual end-point information. The IP VPN solution for Data Center should specify the mapping between the signaling messages used by the hypervisors/switches and the MP-BGP routes used by MP-BGP speakers participating in the virtual network.

In traditional WAN deployments of BGP IP VPNs [RFC 4364], the forwarding function and control function of a Provider Edge (PE) device have co-existed within a single physical router. In a Data Center network, the PE plays a role of the first-hop router, in a

virtual domain. The signaling exchanged between forwarding and control planes in a PE has been proprietary to a specific PE router/vendor. When BGP IP VPNs are applied to a Data Center network, the signaling used between the control plane and forwarding should be open to provisioning and standardization. We explore this requirement in more detail below.

When MPLS/BGP VPNs [RFC 4364] are used to connect VMs or application endpoints, it might be desirable for a hypervisor's host or an external switch (such as TOR) to support only the forwarding aspect of a Provider Edge (PE) function. The VMs or applications would act as Customer Edges (CEs) and the virtual networks interfaces associated with the VMs/applications as CE interfaces. More specifically, a hypervisor/first-hop switch would support only the creation and population of VRF tables that store the forwarding information to the VMs and applications. The forwarding information should include 20-bit label associated with a virtual interface (i.e., a specific VM/application endpoint) and assigned by the destination PE. This label has only a local significance within a destination PE. A hypervisor/first-hop switch would not need to support BGP, a protocol familiar to network devices.

When a PE forwarding function is implemented on an external switch, such as aggregation or top-of-rack switch, the end-system must be able to communicate the endpoint and its VPN membership information to the external switch. It should be able to convey the endpoint's instantiation as well as removal events.

An IP-VPN Data Center networking solution should be able to support a mixture of internal PEs (implemented in hypervisors/Host OS) and external PEs (implemented on external to the end-system devices).

The IP-VPN DC solution should allow BGP/MPLS VPN-capable network devices, such as routers or appliances, to participate directly in a virtual network with the Virtual Machines and applications. Those network devices can participate in isolated collections of VMs, i.e., in isolated VPNs, as well as in overlapping VPNs (called "extranets" in BGP/MPLS VPN terminology).

The device performing PE forwarding function should be capable of supporting multiple Virtual Routing and Forwarding (VRF) tables representing distinct "close user groups". It should also be able to associate a virtual interface (corresponding to a VM or application endpoint) with a specific VRF.

The first-hop router has to be capable of encapsulating outgoing traffic (end-system towards Data Center network) in IP/GRE or MPLS envelopes, including the per-prefix 20-bit VPN label. The first-hop router has to be also capable of associating incoming packets from

a Data Center network with a virtual interface, based on the 20-bit VPN label contained in the packets.

The protocol used by the VPN first-hop routers to signal VPNs should be independent of the transport network protocol as long as the transport encapsulation has the ability to carry a 20-bit VPN label.

### 10.1. Data Center Transport Network

MPLS/VPN technology based on [RFC 4364] specifies several different encapsulation methods for connecting PE routers, namely Label Switched Paths (LSPs), IP tunneling, and GRE tunneling. If LSPs are used in the transport network they could be signaled with LDP, in which case host (/32) routes to all PE routers must be propagated throughout the network, or with RSVP-TE, in which case a full mesh of RSVP-TE tunnels is required, generating a lot of state in the network core. If the number of LSPs is expected to be high, due to a large size of Data Center network, then IP or GRE encapsulation can be used, where the above mentioned scalability is not a concern due to route aggregation property of IP protocols.

### 10.2. BGP Requirements in a Data Center Environment

#### 10.2.1. BGP Convergence and Routing Consistency

BGP was designed to carry very large amount of routing information but it is not a very fast converging protocol. In addition, the routing protocols, including BGP, have traditionally favored convergence (i.e., responsiveness to route change due to failure or policy change) over routing consistency. Routing consistency means that a router forwards a packet strictly along the path adopted by the upstream routers. When responsiveness is favored, a router applies a received update immediately to its forwarding table before propagating the update to other routers, including those that potentially depend upon the outcome of the update. The route change responsiveness comes at the cost of routing blackholes and loops.

Routing consistency across Data Center is important because in large Data Centers thousands of Virtual Machines can be simultaneously moved between server racks due to maintenance, for example. If packets sent by the Virtual Machines that are being moved are dropped (because they do not follow a live path), the active network connections on those VMs will be dropped. To minimize the disruption to the established communications during VM migration, the live path continuity is required.

### 10.2.2. VM Mobility Support

To overcome BGP convergence and route consistency limitations, the forwarding plane techniques that support fast convergence should be used. In fact, there exist forwarding plane techniques that support fast convergence by removing from the forwarding table a locally learned route and instantaneously using already installed new routing information to a given destination. This technique is often referred to as "local repair". It allows to forward traffic (almost) continuously to a VM that has migrated to a new physical location using an indirect forwarding path or tunnel via VM's old location (i.e., old VM forwarder). The traffic path is restored locally at the VM's old location while the network converges to the new location of the migrated VM. Eventually, the network converges to optimal path and bypasses the local repair. BGP should assist in the local repair techniques by advertizing multiple and not only the best path to a given destination.

### 10.2.3. Optimizing Route Distribution

When virtual networks are triggered based on the IP communication (as proposed in this document), the Route Target Constraint extension [RFC 4684] of BGP should be used to optimize the route distribution for sparse virtual network events. This technique ensures that only those VPN forwarders that have local participants in a particular data plane event receive its routing information. This also decreases the total load on the upstream BGP speakers.

### 10.2.4. Inter-operability with MPLS/BGP VPNs

As was stated in section 10, the IP-VPN DC solution should be fully inter-operable with MPLS/BGP VPNs. MPLS/BGP VPN technology is widely supported on routers and other appliances. When connecting a Data Center virtual network with other services/networks, it is not necessary to advertize the specific VM host routes but rather the aggregated routing information. A router or appliance within a Data Center can be used to aggregate VPN's IP routing information and advertize the aggregated prefixes. The aggregated prefixes would be advertized with the router/appliance IP address as BGP next-hop and with locally assigned aggregate 20-bit label. The aggregate label will trigger a destination IP lookup in its corresponding VRF on all the packets entering the virtual network.

## 11. Virtual Machine Migration Requirement

The "Virtual Machine live migration" (a.k.a. VM mobility) is highly desirable for many reasons such as efficient and flexible resource sharing, Data Center migration, disaster recovery, server redundancy, or service bursting. VM live migration consists in moving a virtual machine from one physical server to another, while preserving the VM's active network connections (e.g., TCP and higher-level sessions).

VM live mobility primarily happens within the same physical Data Center but VM live mobility between Data Centers might be also required. The IP-VPN Data Center solutions need to address both intra-Data Center and inter-Data Center VM live mobility.

Traditional Data Center deployments have followed IP subnet boundary, i.e., hosts often stayed in the same IP subnet and a host had to change its IP address when it moved to a different location. Such architecture have worked well when hosts were dedicated to an application and resided in physical proximity to each other. These assumptions are not true in the IaaS environment where compute resources associated with a given application can be spread and dynamically move across a large Data Center.

Many DC design proposals are trying to address the VM mobility with data-center wide VLANs using Data Center-wide Layer 2 broadcast domains. With data-center wide VLANs, a VM move is handled by generating gratuitous ARP reply to update all ARP caches and switch learning tables. Since a virtual subnet locality cannot be preserved in a large Data Center, a virtual subnet (VLAN) must be present on every Data Center switch, limiting the number of virtual networks to 4094. Even if a Layer 2 Data Center solution is able to minimize or eliminate the ARP flooding across Data Center core, all edge switches still have to perform dynamic VM MAC learning and maintain VM's MAC-to-IP mappings.

Since in large Data Centers physical proximity of computing resources cannot be assumed, grouping of hosts into subnets does not provide any VM mobility benefits. Rather, VM mobility in a large Data Center should be based on a collection of host routes spread randomly across a large physical area.

When dealing with IP-only applications it is not only sufficient but optimal to forward the traffic based on Layer 3 rather than on Layer 2 information. The MAC addresses of Virtual Machines are irrelevant to IP services and introduce unnecessary overhead (i.e., maintaining ARP caches of VM MACs) and complications when VMs move (e.g., when VM's MAC address is changed in its new location). IP-based VPN connectivity solution is a cost effective and scalable approach to

solve VM mobility problem. In IP-VPN DC a VM move is handled by a route advertisement.

To accommodate live migration of Virtual Machines, it is desirable to assign a permanent IP address to a VM that remains with the VM after it moves. Typically, a VM/application reaches the off-subnet destinations via a default gateway, which should be the first-hop router (in the virtual topology). A VM/application should reach the on-subnet destinations via an ARP proxy which again should be the VPN first-hop router. A VM/application cannot change the default gateway's IP and MAC addresses during live migration, as it would require changes to TCP/IP stack in the guest OS. Hence, the first-hop VPN router should use a common, locally significant IP address and a common virtual MAC address to support VM live mobility. More specifically, this IP address and the MAC address should be the same on all first-hop VPN routers in order to support the VM moves between different physical machines. Moreover, in order to preserve virtual network and infrastructure separation, the IP and MAC addresses of the first-hop routers should be shared among all virtual IP-subnets/VPNs. Since the first-hop router always performs an IP lookup on every packet destination IP address, the VM traffic is forwarded on the optimal path and traverses the Data Center network only once.

The VM live migration has to be transparent to applications and any external entity interacting with the applications. This implies that the VM's network connectivity restoration time is critical. The transport sessions can typically survive over several seconds of disruption, however, applications may have sub-second latency requirement for their correct operation.

To minimize the disruption to the established communications during VM migration, the control plane of a DC solution should be able to differentiate between VM activation in a new location from advertising its host route to the network. This will enable the VPN first-hop routers forwarders to install a route to VM's new location prior to its migration, allowing the traffic to be tunneled via the first-hop router at the VM's old location. There are techniques available in BGP as well as in forwarding plane that support fast convergence due to withdrawal or replacement of current or less preferred forwarding information (see section 10.2 for more detailed description of such technique).

## 12. IP-VPN Data Center Use Case: Virtualization of Mobile Network

Application access is being done increasingly from clients such as cell phones or tablets connecting via private or public WiFi access

points, or 3G/LTE wireless access. Enterprises with a mobile workforce need to access resources in the enterprise VPN while they are traveling, e.g., sales data from a corporate database. The mobile workforce might also, for security reasons, be equipped with disk-less notebooks which rely on the enterprise VPN for all file accesses. The mobile workforce applications may occasionally need to utilize the compute resources and other functions (e.g., storage) that the enterprise hosts on the infrastructure of a cloud computing provider. The mobile devices might require simultaneous access to resources in both, the cloud infrastructure as well as the enterprise VPN.

The enterprise wide area network may use a provider-based MPLS/BGP VPN service. The wireless service providers already use MPLS/BGP VPNs for enterprise customer isolation in the mobile packet core elements. Using the same VPN technology in the service provider Data Center network (or in a public Data Center network) is a natural extension.

Furthermore, there is a need to instantiate mobile applications themselves as virtual networks in order to improve application performance (e.g., latency, Quality-of-Service) or to enable new applications with specialized requirements. In addition it might be required that the application's computing resource is made to be part of the mobility network itself and placed as close as possible to a mobile user. Since LTE data and voice applications use IP protocols only, the IP-VPN solution to virtualization of compute resources in mobile networks would be the optimal approach.

The infrastructure of a large scale mobility network could itself be virtualized and made available in the form of virtual private networks to organizations that do not want to spend the required capital. The Mobile Core functions can be realized via software running on virtual machines in a service-provider-class compute environment. The functional entities such as Service-Gateways (S-GW), Packet-Gateways (P-GW), or Policy Charging and Enforcement Function (PCEF) of the LTE system can be run as applications on virtual machines, coordinated by an orchestrator and managed by a hypervisor. Virtualized packet core network elements (PCEF, S-GW, P-GW) could be placed anywhere in the mobile network infrastructure, as long as the IP connectivity is provided. The virtualization of the Mobile Core functions running on a private computing environment has many benefits, including faster service delivery, better economies of scale, simpler operations. Since the LTE (Long Term Evolution) and Evolved Packet Core (EPC) system are all-IP networks, the IP-VPN solution to mobile network virtualization is the best fit.



13. Security Considerations

The document presents the problems need to be addressed in the L3VPN for Data Center space. The requirements and solutions will be documented separately.

The security considerations for general requirements or individual solutions will be documented in the relevant documents.

14. IANA Considerations

This document contains no new IANA considerations.

15. Normative References

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

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

[RFC 4684] 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.

16. Informative References

[RFC 2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

17. Authors' Addresses

Maria Napierala  
AT&T  
200 Laurel Avenue  
Middletown, NJ 07748  
Email: mnapierala@att.com

Luyuan Fang  
Cisco Systems  
111 Wood Avenue South

Iselin, NJ 08830, USA  
Email: lufang@cisco.com

Dennis Cai  
Cisco Systems  
725 Alder Drive  
Milpitas, CA 95035, USA  
Email: dcai@cisco.com

## 18. Acknowledgements

The authors would like to thank Pedro Marques for his helpful comments and input.

Internet Engineering Task Force  
Internet-Draft  
Intended status: Standards Track  
Expires: January 16, 2013

D. Freedman  
Claranet  
July 15, 2012

OSPF Version 2 as the Customer Edge/Customer Protocol for BGP/MPLS IP  
VPNs  
draft-freedman-l3vpn-ospf2-4364-ce-01

Abstract

RFC4577 (OSPF as the Provider/Customer Edge Protocol for BGP/MPLS IP VPNs) proposes a mechanism for the use of the Open Shortest Path First V2 ("OSPF", RFC2328) protocol between the Provider Edge ("PE") and Customer Edge ("CE") routers within a BGP/MPLS IP Virtual Private Network ("IPVPN", RFC4364).

The standard provides for use of such a provider VPN to join discontinuous locations together, preserving the OSPF area and domain behaviour.

This document describes a technique for utilising the same, IPVPN network infrastructure without the requirement to enable the OSPF protocol on the PE/CE interface and thus relieve the PE router of OSPF duties.

Status of this Memo

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

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

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

This Internet-Draft will expire on January 16, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Problem Statement . . . . .	5
3. Solution Statement . . . . .	6
4. Domains . . . . .	7
5. Sham Links . . . . .	8
6. Behavioral Considerations . . . . .	9
7. Security Considerations . . . . .	10
8. Acknowledgements . . . . .	11
9. Normative References . . . . .	12
Author's Address . . . . .	13

## 1. Introduction

[RFC4577] describes a mechanism whereby discontinuous locations belonging to the same OSPF area and domain are connected by use of an [RFC4364] IPVPN network.

The premise (see Figure 1) is that OSPF [RFC2328] routing information from a site is learned by the attached PE router through an OSPF adjacency with the site's CE router. This OSPF routing information is learned in the context of a Virtual Routing and Forwarding ("VRF") instance intended to trigger redistribution into the provider BGP as a VPN-IPv4 route through the addition of various Extended Communities [RFC4360] such as "Route Target" (to select the desired destination VRFs when imported by other PE routers), "OSPF Domain Identifier" (to determine if the route should be treated as internal or external to the CE OSPF domain) and "OSPF Route Type" (to encode the LSA type as it was received from the OSPF neighbor).

When a remote PE router, importing such routes into a VRF (due to a matching Route-Target in a VRF import policy), locates the OSPF extended communities, it uses them to originate OSPF LSAs to its attached CE. Providing the OSPF domain ID is the same, BGP routes can be redistributed back into the CE-attached OSPF area using the information encoded in the BGP update, fooling the attached site into thinking that there is a contiguous OSPF domain.

"Sham Links" may then be created between VPN residing endpoints on all involved PE routers, to provide simulated intra-area links, ensuring that any "Backdoor" links between C routers are not automatically selected by OSPF in preference to the provider network links (which would normally be treated as inter-area had the Sham Link not been present).

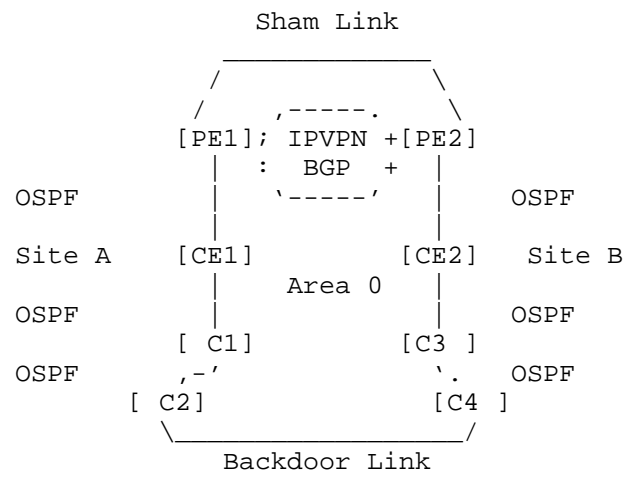


Figure 1

## 2. Problem Statement

The author infers from the title and language in RFC4577 that the original intention of the document was to provide OSPF functionality over an IPVPN network through use of the Provider's PE routers. Since the Provider may use the PE router for multiple customers, and OSPF is based on repeated execution of the Shortest Path First ("SPF") algorithm, this approach may create computation scaling considerations for the PE as the number or complexity of customer topologies using this technology on the PE increases.

### 3. Solution Statement

This document proposes a mechanism for providing this OSPF functionality over IPVPN networks, using only CE routers in a single OSPF domain. A CE router that performs this function is to be known as an O-CE router. Only IP and BGP are therefore required between the O-CE and PE, this is illustrated below in Figure 2

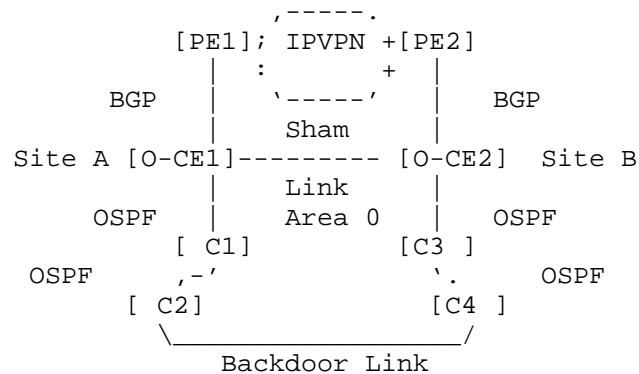


Figure 2

By removing the OSPF from the PE router and placing the responsibility on the O-CE, the provider's existing IPVPN PE routers are no longer forced to run the SPF algorithm since this task can be delegated to the O-CE which does not have the same scaling concerns (it does not share this task with multiple customer domains).



#### 4. Domains

An O-CE is intended to only operate in one OSPF domain, known as the O-CD (O-CE Domain). Though the O-CD is intended to be operator configured on the O-CE, it may instead be automatically discovered (but such mechanisms are outside the scope of this document). It is assumed that the reachability signalled in the O-CD reflects the reachability inside the corresponding attached provider VRF.

An O-CE receiving reachability information via BGP from the IPVPN network from the provider VRF should interact with the C router domain with respect to the O-CD in line with [RFC4577] Section 4.1. In these cases, the O-CE MAY choose to accept reachability concerning a domain other than the O-CD, in such case the O-CE must flood this information as extra-area (type 5/7).

## 5. Sham Links

RFC4577 makes the following requirement of creating Sham Links (Sec 4.2.7.3):

An OSPF protocol packet is regarded as having been received on a particular Sham Link if and only if the following three conditions hold:

- The packet arrives as an MPLS packet, and its MPLS label stack causes it to be "delivered" to the local Sham Link endpoint address.
- The packet's IP destination address is the local Sham Link endpoint address.
- The packet's IP source address is the remote Sham Link endpoint address.

Although RFC4577 marks the use of Sham Links as "OPTIONAL", creation of such links, with respect to the above stated, require that the implementation transmit the OSPF protocol packets over MPLS transport.

Since the intention of this document is to ensure that only IP and BGP are required between O-CE routers, this document relaxes the requirements stated in this RFC section, by removing the requirement for the packet to arrive as an MPLS packet. Since the routing information is redistributed into the BGP and labelled by the PE router for use within the Provider's IPVPN network, an additional MPLS LSP is not required.

This document adds the requirement that BGP should be used as a PE/O-CE protocol and that Extended Communities be made available to both peers through mutual negotiation of the relevant BGP capability [RFC3392].

## 6. Behavioral Considerations

This document makes the following summary recommendations in respect to behavior:

1. That the requirement stated in Section 3 (Requirements) of RFC4577, that OSPF is used as a PE/CE routing protocol be relaxed, such that BGP is used as a PE/O-CE routing protocol and that BGP extended communities are enabled between PE and O-CE. A mechanism to filter said communities SHOULD be made available to the operator to ensure that no other (unwanted) extended communities are injected to or from the provider space.
2. That the requirement stated in Section 4.2.7.3 (OSPF Protocol on Sham Links) of RFC4577 with regards to the requirement that the OSPF protocol packet be received as an MPLS packet, be relaxed in an O-CE router implementation. In its place, the O-CE router MUST verify that the OSPF protocol packet is valid based on the operator configuration of valid Sham Link endpoints.

## 7. Security Considerations

The Behavioral Considerations (Section 6) specify that particular behavioral patterns of RFC4577 be relaxed, references to ensuring appropriate security of these modified behaviors can be found here.

It is important to note that the O-CE operates only in the context of the O-CD, this means that the RFC4577 requirements supporting multiple domain/instance behaviors are not relevant in the scope of the O-CE.

## 8. Acknowledgements

The author would like to thank Paul Wells for his valuable input.

## 9. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, April 1998.
- [RFC3392] Chandra, R. and J. Scudder, "Capabilities Advertisement with BGP-4", RFC 3392, November 2002.
- [RFC4360] Sangli, S., Tappan, D., and Y. Rekhter, "BGP Extended Communities Attribute", RFC 4360, February 2006.
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.
- [RFC4577] Rosen, E., Psenak, P., and P. Pillay-Esnault, "OSPF as the Provider/Customer Edge Protocol for BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4577, June 2006.

Author's Address

David Freedman  
Claranet  
London  
UK

Phone: +44 20 7685 8000

Email: david.freedman@uk.clara.net





L3VPN Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 31, 2013

L. Han, Ed.  
R. Li  
Huawei Technologies  
July 30, 2012

Multicast VPN Support by Receiver-Driven Multicast Extensions to RSVP-TE  
draft-hlj-l3vpn-mvpn-mrsvp-te-01

## Abstract

This document describes a method to support multicast VPN (MVPN) by a receiver-driven multicast extension to RSVP-TE (mRSVP-TE). This method is desirable and applicable to MVPN applications when QoS assurance and traffic-engineered tunnels are desired.

## Status of this Memo

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

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

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

This Internet-Draft will expire on January 31, 2013.

## Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

## Table of Contents

1. Introduction . . . . .	4
2. Terminology . . . . .	5
2.1. Definitions . . . . .	5
3. Overview . . . . .	6
3.1. Multicast LSP . . . . .	7
3.2. PIM States, PIM Interfaces and PMSI . . . . .	7
3.3. Reverse-Path Forwarding . . . . .	8
3.4. Default mLSP . . . . .	9
3.4.1. Establishment of Default mLSP . . . . .	9
3.4.2. Virtual PIM Interface for Default mLSP . . . . .	10
3.4.3. PIM signaling over Default mLSP . . . . .	10
3.4.4. PIM state with Default mLSP . . . . .	12
3.4.5. Multicast data over default mLSP . . . . .	13
3.5. Data mLSP . . . . .	13
3.5.1. Establishment of Data mLSP . . . . .	13
3.5.2. Virtual PIM interface for Data mLSP . . . . .	14
3.5.3. mLSP ID and data mLSP mapping . . . . .	14
3.5.4. Switching of Data mLSP . . . . .	15
3.5.5. PIM Prune Impact to Data mLSP . . . . .	15
3.5.6. Deletion of Data mLSP . . . . .	16
3.5.7. PIM (S,G) signaling after Data mLSP is created . . . . .	16
4. PIM-SSM Solutions . . . . .	16
4.1. Option 1 . . . . .	16
4.2. Option 2 . . . . .	17
4.3. Option 3 . . . . .	17
5. PIM-SM solutions . . . . .	18
5.1. RP-PE mLSP . . . . .	18
5.2. Source-PE mLSP . . . . .	18
5.3. PIM register process . . . . .	18
5.3.1. Scenario 1: The multicast source and RP are behind the same PE . . . . .	18
5.3.2. Scenario 2: The multicast source and RP are behind the different PE . . . . .	19

5.4.	SPT switching	21
5.5.	RPT prune	21
5.6.	Data mLSP switching	21
5.7.	PIM state at Receiver-PE	22
6.	Aggregation	22
6.1.	Aggregation by Default mLSP	23
6.2.	Aggregation by Data mLSP	23
7.	Non-VPN multicast support	23
8.	Message Format and Constants	24
8.1.	Path session object for PIM-SSM option 1 (IPv4)	24
8.2.	Path session object for PIM-SSM option 1 (IPv6)	25
8.3.	Path session object for other PIM modes (IPv4)	26
8.4.	Path session object for other PIM modes (IPv6)	27
8.5.	mLSP TLV Message format for IPv4	27
8.6.	mLSP TLV Message format for IPv6	28
9.	Acknowledgements	29
10.	IANA Considerations	29
11.	Security Considerations	29
12.	References	29
12.1.	Normative References	29
12.2.	Informative References	30
	Authors' Addresses	30

## 1. Introduction

A L3VPN service that supports multicast is known as a Multicast VPN, or MVPN for short. There have been different proposed messages, procedures and mechanisms to support MVPN. These methods differ in protocols used in the service provider's network, for example, the mGRE-based MVPN, BGP extensions to transport customer's PIM signaling and P2MP RSVP-TE extensions to transport multicast data streams, and mLDP-based MVPN, as summarized as follows:

Type	Data Plane	Protocols for core	Standard
1	mGRE	PIM, BGP(with MDT_SAFI)	RFC6037
2	MPLS	P2MP RSVP-TE, BGP(with extension)	RFC6513
3	MPLS	mLDP	No

Table 1. Existing mVPN Solutions

Type 1 solution requires to run PIM in the service provider's network.

Both Type 2 and Type 3 require an MPLS data forwarding plane, but they differ in protocols used in the service provider's network. Type 2 uses RSVP-TE with a P2MP extension [RFC4875] for multicast data streams and BGP extensions with multicast encodings and procedures [RFC6514] for PIM signaling, or use mLDP [RFC6388] for both control plane signaling and multicast data streams. Type 3 is simpler than type 2 in terms of required protocols and provisioning.

With Type 2 solution, multicast traffic is carried over MPLS-TE tunnels, QoS and traffic engineering are supported for mVPN applications. It is an advantage of Type 2 over Type 3.

However, for Type 2 solution, BGP has to be extended with seven (7) types of MCAST-VPN NLRI's together with four (4) new BGP attributes. In some scenarios, multiple P2MP RSVP-TE tunnels are used. And therefore, it requires to upgrade both BGP and RSVP-TE, which brings more complexity and operational inconvenience.

In addition to the above-mentioned three methods, do we have any alternative method which is expected to be simpler and more scalable, but can still provide QoS assurance and traffic-engineered transport?

This document specifies a new method to implement multicast VPN by

receiver-driven multicast extensions to RSVP-TE (mRSVP-TE) specified in [I-D.lzj-mpls-receiver-driven-multicast-rsvp-te]. mRSVP-TE is a new extension to RSVP-TE for multicast applications in MPLS networks, whose behavior is closer to IP PIM since both of them work by sending control messages from the data receivers to the data senders. The receiver-driven nature of the mRSVP-TE makes it more adaptive and easier to be integrated with PIM for multicast applications including multicast VPN.

As an extension to RSVP-TE, mRSVP-TE inherits all the desirable features from RSVP-TE such as QoS assurance and traffic-engineered paths, which makes it to distinguish from mLDP used in Type 3.

By using an MP2MP tunnel created by mRSVP-TE to carry the customer's PIM signaling, we do not need to use BGP multicast extension to signal customer's multicast information.

The MVPN method described in the document supports both PIM-SSM and PIM-SM. For PIM-SM, this method supports multicast source, receiver, Rendezvous Point (RP) located at any place including PE, CE or any device connected to CE. It can also support Bootstrap Router (BSR) Mechanism [RFC5059].

## 2. Terminology

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] and indicate requirement levels for compliant MVPN implementations.

### 2.1. Definitions

In what follows we describe some terminologies which are widely used in this document.

#### Source-PE

Source-PE is a PE which is connected to a MVPN CE and the multicast source is on or behind the CE.

#### Receiver-PE

Receiver-PE is a PE which is connected to a MVPN CE and the multicast receiver is on or behind the CE

#### RP-PE

RP PE is a PE which is connected to a MVPN CE and the multicast Rendezvous Point for PIM-SM is on or behind the CE

**PE type**

A PE can be either Source-PE, Receiver-PE or RP-PE for different MVPN and different (S,G). Its type can also be the mixture of any combination of the three PE type.

**MDT**

Multicast Distribution Tree, introduced in [RFC6037] for the IP backbone based MVPN. MDT is composed of mGRE tunnels. There are default MDT and data MDT

**mLSP**

Multicast-Label-Switched-Path. It is the equivalent of MDT in MPLS networks, and sometimes we will use mLSP and MDT interchangeably. The same as MDT, we also have default mLSP and data mLSP.

**Source-PE mLSP**

mLSP whose header-end is a Source-PE.

**RP-PE mLSP**

mLSP whose header-end is a RP-PE.

**MI-PMSI(MVPN\_ID)**

It corresponds to the MI-PMSI [RFC6513] for a MVPN (ID is MVPN\_ID), it is the Multidirectional Inclusive P-Multicast Service interface for the default mLSP or the MP2MP tunnel at either tail-end or header-end.

**S-PMSI(MVPN\_ID, mLSP\_ID)**

It corresponds to a S-PMSI [RFC6513] for a MVPN (ID is MVPN\_ID) and the mLSP ID is mLSP\_ID, it is the Selective P-Multicast Service interface for the P2MP tunnel for (S,G) at either tail-end or header-end.

**OIF**

Outgoing Interface for PIM state

**IIF**

Incoming Interface for PIM state

**Olist**

Outgoing Interface list for PIM state

### 3. Overview

### 3.1. Multicast LSP

Multicast-Label-Switched-Path (mLSP) is an MPLS tree in MPLS network to distribute multicast data to different receivers who are interested in particular multicasted data stream(s). An mLSP is composed of multiple Sub-Label-Switched-Paths (sub-LSP) which connect different Label Switch Routers (LSRs) to form an MPLS multicast network. There are two basic types of mLSPs: P2MP LSP and MP2MP LSP. In the case of P2MP LSP, the header-end of an mLSP is the Source-PE which connects the source device of multicast traffic, and its tail-ends are the Receiver-PEs which connect the destination device of multicast traffic. The joint points on an mLSP are called "Branch LSR" where the MPLS packets are replicated and then forwarded to different downstream LSRs. In the case of MP2MP LSP, there is a special LSR which serves as the root of the mLSP, and all the leaf nodes are both the Source-PE and Receiver-PEs.

For mVPN multicast traffic, it travels on a multicast tree which spans over two different networks: MPLS network operated by service providers and IP network on the customer's sites. The mVPN multicast traffic always starts from one customer's site as IP multicast, and then is transported over the MPLS network to other customer's sites. The traffic on customer's sites is distributed over a PIM multicast distribution tree, while in service provider's MPLS network it is distributed by mLSP tunnels. The mLSP and the PIM distribution tree SHOULD be seamlessly integrated. The IP multicast data received from a CE is encapsulated as MPLS packet at the Source-PE of an mLSP tree, and then transported over the mLSP. The MPLS packet is replicated at the branch LSRs and delivered to the different Receiver-PEs, where the MPLS packet is de-capsulated to IP multicast packet and forwarded to the connected IP multicast tree, then it is distributed to the particular receivers.

### 3.2. PIM States, PIM Interfaces and PMSI

It is assumed that PIM is used on customer's sites and mRSVP-TE is used in service provider's network without PIM being enabled in service provider's network. In order to set up customer's multicast distribution trees across a service provider's MPLS network, it is desired that the customer's PIM SHOULD inter-work with service provider's mRSVP-TE, which brings up some new requirements about PIM states and interfaces.

The most important factors for PIM states are the IIF and OIF, both of which are PIM-enabled interfaces. A PIM interface appearing in an PIM state is characterized as follows:

- o It has an IP address configured
- o It has the PIM protocol enabled and running
- o It has one or more PIM adjacencies

Since the customer's PIM adjacencies MUST be established between PEs, virtual interfaces associated with the MPLS tunnels connecting PEs are introduced. Such virtual interfaces are also called PMSI for Provider Multicast Service Interface. In this document we will use two types of PMSI: MI-PMSI and S-PMSI.

### 3.3. Reverse-Path Forwarding

For multicast forwarding by a PIM state (S,G), we need to check if the packet is coming from the expected interface which is the egress interface to reach the source S based on the unicast routing table. The expected interface is called RPF interface, or the IIF (Incoming interface). In this document, we will consider the following two modes:

- o PIM-SSM mode: the state to forward traffic is (S,G), so there is only one IIF for a (S,G).
- o PIM-SM mode: RP will have (\*,G) before the traffic is received and (S,G) after the register processing is finished. Other routers have (\*,G) before the SPT switching and (S,G) after the SPT switching. For the (\*,G), the RPF is the interface to reach RP by unicast routing.

In the context of mVPN, PE is the boundary router between customer's IP network and service provider's MPLS network, and thus needs to handle the RPF issue as follows:

- o For a Source-PE, the RPF checking for any (S,G) does not have any change since the IIF is still a normal IP interface.
- o For a Receiver-PE, the RPF interface for any (S,G) or (\*,G) is not derived from the unicast routing table for the multicast source S or RP for a multicast stream. Instead, we MUST force the RPF interface to be the PIM interface which is associated with either an MI-PMSI or an S-PMSI.
- o For a RP-PE, before traffic starts, the RPF interface is not set for (\*,G) since the multicast source is unknown. After the PIM register process is completed, the (S,G) state will be created. Then we MUST force the RPF interface to be the PIM interface which is associated with the MI-PMSI for the MVPN.



RPF does not apply to the multicast forwarding in MPLS network by mLSP. mLSP established by mRSVP-TE protocol can guarantee the loop-free for packet forwarding which is the whole purpose of RPF checking.

### 3.4. Default mLSP

To each mVPN, we associate an mLSP, called its default mLSP. Given an mVPN, its default mLSP is a multi-directional shared tree with all the PEs as its leaf nodes. Default mLSP is a MP2MP tunnel which can provide a multi-directional transportation for any data. The default mLSP is used for the following two purposes:

- o Customer's PIM signaling: Customer's PIM signaling is transported over such default mLSP
- o Default customer's multicast data distribution: Customer's multicast data are transported and distributed over such mLSP by default.

#### 3.4.1. Establishment of Default mLSP

The construction of default mLSP does not depend on the existence of multicast traffic for a MVPN; it is built up before any such multicast traffic is seen.

Default mLSP is established when a VPN attached to a PE enables MVPN service. After the MVPN is enabled, the mRSVP-TE stack MUST send the mRSVP-TE path message continuously. The time interval to send the path message at each PE could be default value or configurable. If it is configurable, different PE's interval value MUST be proper to guarantee the mLSP state is steady without any flapping.

To enable MVPN service on a PE, root node(s) IP address MUST be given. Root node is normally a P router inside the backbone network.

The location of root node may impact the efficiency of a MP2MP tunnel. How to choose a root node to establish a MP2MP tunnel to obtain the efficient multicast replication in MPLS network is out of scope of the document.

In addition to the root node, explicit nodes from any PE to the root node P MAY be applied as an option if user wants the path from the root node P to a PE goes through some expected routers.

For the details of root node and explicit node in a MP2MP tunnel, please refer to the [I-D.lzj-mpls-receiver-driven-multicast-rsvp-te].

If the redundancy for the root node is desired to protect the failure of root node, multiple root nodes may be given to construct multiple default mLSP. The redundancy for root node is out of scope of this document.

With the method herein, there is only one default mLSP for each MVPN, or two for root redundancy case,

#### 3.4.2. Virtual PIM Interface for Default mLSP

A MI-MPSI interface SHOULD be created at both Source-PE and Receiver-PE when the default mLSP for a MVPN is established. This is a PIM enabled interface for a MVPN. It is used for PIM adjacency, PIM state keeping, and PIM IIF and OIF representation for the MPLS packet forwarding over MPLS network.

MI-PMSI is a joint point of IP multicast tree and mLSP. If a MI-PMSI is one OIF in the Olist for a multicast forwarding entry (S,G), it means the IP multicast stream (S,G) will be replicated for the MI-PMSI and sent to the interface. If a MI-PMSI is the IIF for a multicast forwarding entry (S,G), it means the MPLS packet received from MI-PMSI will be forwarded by the forwarding entry (S,G) if the de-capsulated MPLS packet is IP packet, and source and group are S and G respectively.

MI-PMSI is a PIM interface and its IP address will be the address for the PIM peering. This address on one PE MUST be reachable from all other PEs. When PIM adjacency are established between PEs, one PE can see all its PIM adjacency's MI-PMSI are up. For the convenience, it is RECOMMENDED to use the BGP source address for the BGP session between PEs for MI-PMSI. The BGP session here refers to the BGP for unicast VPN service.

All PIM hello variables for a virtual interface MI-PMSI, such as timer, are default value when the interface is created. But they could be configurable and this is up to the implementation.

#### 3.4.3. PIM signaling over Default mLSP

For MVPN attached PE, PIM is enabled for the interfaces connecting different CEs which may belong to the same or different VPNs. Each interface has a MVPN instance associated with it. After a MI-PMSI(MVPN\_ID) is created for a default mLSP, it MUST join the same PIM domain for the particular MVPN.

The default mLSP SHOULD support all PIM multicast messages:

- o HELLO message
- o JOIN/PRUNE message
- o ASSERT
- o BOOTSTRAP

For the following PIM unicast message, they SHOULD NOT be sent to the default mLSP, instead, they SHOULD be sent over a unicast MPLS tunnel to the destination PE.

- o REGISTER message
- o REGISTER-STOP message
- o GRAFT
- o GRAFT-ACK
- o CANDIDATE-RP-ADV

For one MVPN at a PE, PIM signaling (multicast) messages SHOULD be multicasted to all PIM interfaces for that particular MVPN including MI-PMSI. PIM messages are sent to a MI-PMSI(MVPN\_ID) immediately after the interface is created. The messages are encapsulated to MPLS packets and will be distributed to all other receiver-PEs in the same MVPN through the default mLSP.

At Receiver-PE, the MPLS packets are de-capsulated and delivered to a particular MVPN, the MVPN ID is determined by the MPLS label which was allocated locally on Receiver-PE when the PE initializes the default mLSP by sending mRSVP-TE path message to the root node. The popped MPLS label from the received MPLS packet can associate the packet with a MI-PMSI(MVPN\_ID) interface as incoming interface, So, the MI-PMSI(MVPN\_ID) interface at Receiver-PE can be used for RPF checking of multicast forwarding.

Receiver-PE SHOULD punt PIM signaling message to PIM protocol stack for the particular MVPN. After all PIM HELLO messages are exchanged between PEs, PIM adjacencies are established between multiple PEs through each PE's MI-PMSI(MVPN\_ID) which is associated with a particular MVPN.

As the result of PIM adjacency over the default mLSP, the details of MPLS backbone network topology is hidden for PIM. It will make the MVPN PIM virtually run over a virtual multi-access interface which is composed of multiple MI-PMSI(MVPN\_ID) from different PEs.

#### 3.4.4. PIM state with Default mLSP

Since the MI-PMSI interface is a PIM enabled interface, all PIM multicast messages, Hello, Join, Prune, Bootstrap and Assert, can be sent to or received from the MI-PMSI interface. PIM protocol can create and update the appropriate state for a MVPN accordingly. MI-PMSI can behavior as a normal PIM interface to join or exit the Olist for PIM state.

Below is the detail of the PIM processing for different PIM modes and join messages. All behaviors are based on the PIM protocol and some PIM changes are required for MVPN solution described in this document.

##### (S,G) join for PIM-SSM

When a Receiver-PE receives PIM (S,G) join message from attached CE, it SHOULD send the join message through MI-PMSI(MVPN\_ID) interface to the default mLSP. Meanwhile, if PIM (S,G) state was not created on the Receiver-PE, PIM MUST create a (S,G) state for which the MI-PMSI(MVPN\_ID) is IIF. As a result of sending PIM join message to MI-PMSI(MVPN\_ID) interface, the message will be populated to all PEs connected to the same default mLSP. However, only Source-PE SHOULD process the PIM join message. The Source-PE is derived from the BGP next hop of source address S. All other PEs SHOULD ignore the join message. After the Source-PE receives the (S,G) join from a default mLSP, if the PIM (S,G) state was not created, PIM SHOULD create a PIM (S,G) state for multicast routing table, the entry SHOULD add MI-PMSI(MVPN\_ID) to its Olist. After the 1st PIM join message is processed at both Receiver-PE and Source-PE, the subsequent PIM join message will only reset the PIM timer and will not change the PIM (S,G) state. This behavior is same as PIM in IP network.

##### (\* ,G,RP) join for PIM-SM

PIM-SM (\* ,G) join message will be populated through default mLSP to all PEs attached to the same mLSP. The behavior for PIM (\* ,G) join is the same as PIM-SSM. The only difference is that (\* ,G) join is sent to RP (Rendezvous Points). As a result, only RP-PE SHOULD process the PIM join message. The RP-PE is derived from the BGP next hop of RP address. All other PEs SHOULD ignore the join message. After the PIM (\* ,G) join message is sent from Receiver-PE and received by RP-PE. PIM SHOULD create a (\* ,G) state on Receiver-PE, for which the MI-PMSI(MVPN\_ID) is IIF. PIM SHOULD create a (\* ,G) state at RP-PE and add the MI-PMSI(MVPN\_ID) to its Olist.

PIM prune message processing has no change on PE, it may lead to the

interface state change for a PIM state, or a PIM state deletion. When a PIM state is deleted on a receiver-PE, it MUST send the PIM prune message to the default mLSP to forward the prune message to source-PE or RP-PE. When a Source-PE or RP-PE receives a prune message from the default mLSP, it MUST prune the MI-PMSI from the PIM state's Olist.

#### 3.4.5. Multicast data over default mLSP

If a default mLSP is used to carry user's multicast data, it will send the multicast data to all PEs connected to the default mLSP, no matter if a PE is intended or not to receive the particular multicast traffic. The PIM join or prune does not start or stop the traffic over the default mLSP. This is normally used for the beginning of the multicast traffic flowing when the traffic rate is low. Obviously, there are two drawbacks for it

- o Some PE may not want to receive some multicast traffic, this will be wasteful for the bandwidth and resource for routers.
- o Too much multicast data shares one tree can congest the MP2MP tunnel.

To overcome above problems, data mLSP is used to offload the data traffic from the default mLSP.

#### 3.5. Data mLSP

Data mLSP is used to offload some data stream from the default mLSP. It is a P2MP tunnel corresponding to a (S,G) entry in a MVPN. Data mLSP is built up either statically or dynamically depending on the solutions for different PIM modes. Section 4 and 5 will discuss details.

##### 3.5.1. Establishment of Data mLSP

Static data mLSP establishment is triggered by a Receiver-PE to send the mRSVP-TE P2MP path message to a Source-PE. It will be described in section 4.1.

Dynamical data mLSP establishment is driven by the multicast traffic. The mechanism is similar to the data MDT described in [RFC6037].

Source-PE MUST monitor the rate of all multicast streams passing through it. As for how to monitor the traffic rate, it is out of the scope of the document.

When a Source-PE detects the rate of a MVPN multicast stream (S,G)

exceeds the pre-configured threshold, it MUST send a data mLSP join TLV to the default data mLSP. The format of data mLSP join TLV is defined in section 8.5 and 8.6.

The data mLSP join TLV will be flooded to all PE connected to the same default mLSP. When a PE receives the data mLSP join TLV and if the PE has joined the group G, it MUST initialize the setup of P2MP tunnel by sending the mRSVP-TE P2MP path message to the Source-PE for (S,G). Source-PE address is derived from the BGP nexthop of the VPN address S.

The periodically sending of mRSVP-TE path message from receiver-PE to Source-PE is driven by the periodically received mLSP join TLV message at receiver-PE

The operation of data mLSP is similar to the operation of data MDT for mGRE based mVPN. It has four timer defined to govern the data mLSP: MDT\_DATA\_DELAY, MDT\_DATA\_TIMEOUT, MDT\_DATA\_HOLDDOWN, MDT\_INTERVAL. For the detailed definition of those timers and operations, please refer to [RFC6037].

Since the interval to receive mLSP join TLV message will determine the interval to send mRSVP-TE path message, we SHOULD make sure the interval of mLSP join TLV is less than the timeout value of sub-LSP created by the mRSVP-TE path message.

### 3.5.2. Virtual PIM interface for Data mLSP

After a data mLSP is created, the S-PMSI(MVPN\_ID,mLSP\_ID) MUST be instantiated. S-PMSI(MVPN\_ID,mLSP\_ID) is only used for Incoming Interface (IIF) at Receiver-PE and Outgoing Interface (OIF) at Source-PE for the multicast forwarding, it is not used for PIM signaling.

The mLSP\_ID is "mLSP ID" shown in Fig.3 which is assigned at Source-PE.

S-PMSI(MVPN\_ID,mLSP\_ID) points to the same PIM interface as MI-PMSI(MVPN\_ID). It only adds extra L2 rewriting information block to the PIM interface for the packet forwarding purpose.

### 3.5.3. mLSP ID and data mLSP mapping

Data mLSP is identified by "mLSP ID" which is defined in section 8.3 and 8.4. mLSP ID is a 4 byte value starting from 1 for data mLSP. mLSP ID 0 is reserved for the default mLSP. mLSP ID is to distinguish different data mLSP (P2MP tunnel) at Source-PE side. During the data mLSP building, the mLSP ID allocated at a Source-PE MUST be notified

to all Receiver-PE by the mLSP join TLV.

When a Source-PE detects the rate of a MVPN multicast stream (S,G) exceeds the pre-configured threshold, it MUST assign a mLSP ID from its mLSP pool for the (S,G). And the mLSP join TLV message binds (S,G) with mLSP ID. The Receiver-PE receiving the mLSP join TLV will know the binding relationship. As a result, both Source-PE and Receiver-PE will have a mapping for the mLSP ID and data mLSP, this is used for the switching of data MDT for a stream (S,G).

After a data mLSP is deleted, the associated mLSP ID MUST be returned to the mLSP pool.

#### 3.5.4. Switching of Data mLSP

The Source-PE SHOULD switch the traffic from default mLSP to data mLSP after it created the data mLSP for a multicast stream (S,G). The mLSP ID and data mLSP mapping information will tell which data mLSP is used for which stream (S,G). From the PIM state point of view, at Source-PE, the PIM state (S,G) SHOULD change the OIF from MI-PMSI(MVPN\_ID) to S-PMSI(MVPN\_ID, mLSP\_ID). Since MI-PMSI(MVPN\_ID) and S-PMSI(MVPN\_ID, mLSP\_ID) share the same PIM interface, the switching essentially means the MPLS forwarding is switched from the MP2MP tunnel to P2MP tunnel. There is no PIM interface changing for PIM signaling during and after the data mLSP switching

After switching, Receiver-PE MUST use the correct data mLSP associated S-PMSI(MVPN\_ID, mLSP\_ID) for the RPF checking for a stream (S,G).

The data mLSP switching is associated with the change of forwarding state for (S,G) as following

- o Source-PE MUST modify the OIF from MI-PMSI(MVPN\_ID) to S-PMSI(MVPN\_ID, mLSP\_ID).
- o Receiver-PE MUST modify the IIF from MI-PMSI(MVPN\_ID) to S-PMSI(MVPN\_ID, mLSP\_ID).

#### 3.5.5. PIM Prune Impact to Data mLSP

When the multicast data is transported over a data mLSP, the PIM prune may cause the prune of the data mLSP tree. After a Receiver-PE receives PIM prune message and if the prune message leads to the IIF prune for a PIM state, it MUST update the PIM state in such that the IIF represented by the S-PMSI(MVPN\_ID, mLSP\_ID) is pruned. And the Receiver-PE MUST send the mRSVP-TE PATHTEAR message to the upstream LSR to prune the data mLSP tree. If a Source-PE receives the

mRSVP-TE PATHTEAR message, the whole data mLSP is deleted and Source-PE MUST stop flooding the mLSP join TLV to the default mLSP.

#### 3.5.6. Deletion of Data mLSP

Data mLSP join TLV will be flooded through default mLSP periodically by the interval of MDT\_INTERVAL [RFC6037], if during the timeout period defined by MDT\_DATA\_TIMEOUT [RFC6037], there is no mLSP join TLV received for a receiver-PE, the receiver-PE will start to delete the P2MP leaf from the data mLSP. This is done by sending mRSVP-TE PATHTEAR message to the upstream LSR. After the whole data mLSP is deleted, the traffic will be switched back to the default mLSP.

#### 3.5.7. PIM (S,G) signaling after Data mLSP is created

When a data mLSP is created for a particular multicast stream (S,G), the PIM signaling is not changed. PIM join, prune for (S,G) is still going through the default mLSP.

### 4. PIM-SSM Solutions

To support PIM-SSM by mRSVP, we have three options.

#### 4.1. Option 1

PIM (S,G) join message received at Receiver-PE MUST trigger the data mLSP setup by sending a mRSVP-TE P2MP path message to the Source-PE, if the data mLSP was not created before. Source-PE address is the BGP next hop of the address S. The mRSVP-TE path message MUST embed the (S,G) information as shown in Fig. 1.

PIM join message is sent to default mLSP and received by the Source-PE. This SHOULD trigger the PIM (S,G) state created at Source-PE and Receiver-PE. The (S,G) state at Source-PE MUST add the S-PMSI(MVPN\_ID,mLSP\_ID) for the data mLSP to its Olist; The (S,G) state at receiver-PE SHOULD set the S-PMSI(MVPN\_ID,mLSP\_ID) as IIF.

After the PIM (S,G) state created at Source-PE, the traffic can be sent to data mLSP immediately.

The P2MP mRSVP-TE path message for data mLSP MUST include the ERO objects when the explicit path is given for the source S.

There is no default mLSP to data mLSP switching for this option.



#### 4.2. Option 2

PIM (S,G) join message received at Receiver-PE MUST be sent to the default mLSP and received by the Source-PE. This SHOULD trigger the PIM (S,G) state created at Source-PE and Receiver-PE, if the PIM state was not created before. The PIM (S,G) state at Source-PE SHOULD add the MI-PMSI(MVPN\_ID) as OIF; The (S,G) state at receiver-PE SHOULD add the MI-PMSI(MVPN\_ID) as IIF.

After the (S,G) state created at source-PE, the traffic can be sent to the default mLSP.

Source-PE MUST detects the rate for the multicast stream (S,G) in a MVPN. If the traffic rate for (S,G) exceeds the configured threshold, the Source-PE MUST flood the mLSP join TLV to all PEs. Each PE, if it is interested to receive the traffic for (S,G), MUST initialize a mRSVP-TE P2MP path message to the Source-PE.

The P2MP path message MUST include the ERO objects when the explicit path is given for the source S.

After the Source-PE creates a data mLSP for (S,G), it MUST switch the traffic from default mLSP to data mLSP.

#### 4.3. Option 3

PIM (S,G) join message received at Receiver-PE MUST be sent to the default mLSP and received by the Source-PE. This SHOULD trigger the PIM (S,G) state created at Source-PE and Receiver-PE, if the PIM state was not created before. Unlike the option 2, PIM does not add the default mLSP interface MI-PMSI(MVPN\_ID) as the IIF and OIF for (S,G) state. In stead, Source-PE MUST trigger a mLSP join TLV flooded to all PEs. Each PE, if it is interested to receive the traffic for (S,G), MUST initialize a mRSVP-TE P2MP path message to the Source-PE to build up a data mLSP.

As the result of data mLSP setup, The PIM (S,G) state at receiver-PE MUST add the S-PMSI(MVPN\_ID, mLSP\_ID) as IIF. At the Source-PE, after the data mLSP is created. The PIM (S,G) state MUST add the S-PMSI(MVPN\_ID, mLSP\_ID) as OIF;

The P2MP path message MUST include the ERO objects when the explicit path is given for the source S.

There is no default mLSP to data mLSP switching for this option.

## 5. PIM-SM solutions

PIM-SM supporting is different to the PIM-SSM. It involves some extra process like PIM register, register stop, RPT and SPT switching, etc [RFC4601]. Following describes the details of different scenarios for MVPN PIM-SM.

### 5.1. RP-PE mLSP

RP-PE mLSP is a mLSP whose header-end is at the RP-PE, and multiple tail-ends at different Receiver-PEs. RP-PE mLSP is the equivalence of RPT (RP tree or shared tree) of IP PIM in MPLS network. RP-PE mLSP will use the default mLSP in the method specified in this document.

PIM (\*,G,RP) join message received at Receiver-PE MUST be sent to the default mLSP and finally reach the RP-PE. Then, the Source-PE and RP-PE can create the PIM state for (\*,G). The (\*, G) state at RP-PE MUST have the MI-PMSI(MVPN\_ID) as its OIF, and the (\*, G) state at Receiver-PE MUST have the MI-PMSI(MVPN\_ID) as its IIF.

### 5.2. Source-PE mLSP

Source-PE mLSP is a mLSP whose header-end is at a Source-PE. Depending on the location of tail-end, we have Source-PE to RP-PE mLSP, and Source-PE to Receiver-PE mLSP. Source-PE to RP-PE mLSP is the tree whose header-end is at the source-PE, and the tail-ends at RP-PE. It is constructed after the PIM register process is finished but before the PIM SPT switching or data mLSP switching. Source-PE to receiver-PE mLSP is the tree whose header-end is at the source-PE, and the tail-ends at receiver-PEs. It is built after SPT switching or data mLSP switching. By the method specified in this document, Source-PE to RP-PE mLSP also use the default mLSP like RP-PE mLSP. source-PE to receiver-PE mLSP will use the data mLSP.

When the Source-PE and RP-PE are same (scenario 1 in section 6.3.1), there is no Source-PE to RP-PE mLSP.

### 5.3. PIM register process

PIM register process is between multicast source and RP. Depending on the PR location, we can have different scenarios.

#### 5.3.1. Scenario 1: The multicast source and RP are behind the same PE

For this scenario, both RP and the multicast source are behind the same PE for the same MVPN. In another words, the unicast path from the multicast source to the multicast RP for a particular MVPN does

not need to go through one PE to another PE and cross the MPLS network. So, the RP-PE is also the Source-PE. The PIM register process does not cross different PEs in the core MPLS network. Both RP-PE and source-PE are not aware of the PIM register process. There is no particular design consideration for MPLS tunnels.

Before PIM register process, the PIM (\*,G) join message from different Receiver-PE MUST be forwarded to the RP-PE. As a result, the PIM (\*,G) state MUST be created on both RP-PE and Receiver-PE. The state (\*,G) at RP-PE has the MI-PMSI(MVPN\_ID) as its OIF, and the state (\*,G) at Receiver-PE has the MI-PMSI(MVPN\_ID) as its IIF.

After the PIM register process is finished, PIM state on RP-PE will be changed to (S,G) which inherits all OIF from its parent (\*,G). There is no change for PIM state for (\*,G) at Receiver-PE. The multicast traffic will be flooded to all Receiver-PE through the RP-mLSP, or default mLSP. the Receiver-PE SHOULD drop the traffic if it does not have the (\*,G) state created before.

#### 5.3.2. Scenario 2: The multicast source and RP are behind the different PE

For this scenario, RP and the multicast source are behind the different PEs for the same MVPN. The unicast path from the multicast source to the multicast RP MUST go through source-PE to RP-PE and cross the MPLS network.

After the PIM(\*,G) join is forwarded to the RP-PE through the default mLSP from different Receiver-PE, Only the RP-PE and receiver-PE have the state (\*,G) created. The state at RP-PE has the default mLSP as its OIF, and the interface connecting to a CE as the IIF, which CE is the nexthop to reach RP from the RP-PE. The state at receiver-PE has the default mLSP as IIF.

At Source-PE, there is no forwarding state. Multicast source S MUST start the register process by sending data packet to RP. The PIM register message is IP unicast message (encapsulated multicast data) which destination is to RP from source S, it SHOULD go through a unicast MPLS tunnel from Source-PE to RP-PE. The creation of unicast MPLS tunnel is out of scope of this document.

When RP-PE receives register message which is encapsulated in MPLS format, following things SHOULD happen:

- o RP-PE MUST de-capsulate the MPLS packet and forward the PIM register message to RP behind the RP-PE. RP then MUST forward the multicast packet (de-capsulated from PIM register message) to all Receiver-PE. This is done through the (\*,G) state created before

PIM register process. The traffic from RP will be forwarded back to RP-PE from the interface connecting to CE, and then RP-PE will forward the traffic to RP-mLSP, or the default mLSP.

- o All PEs attached to the default mLSP SHOULD receive the traffic. Source-PE and receiver-PE which did not join the group G SHOULD drop the traffic.
- o RP initialize a PIM (S,G) join to source S. S address is retrieved from the received data traffic from PIM register message. The (S,G) join message MUST be forwarded from RP to RP-PE, and then RP-PE MUST forward the join through the default mLSP to the Source-PE. The address of Source-PE is determined by the BGP next hop of the VPN address S.
- o The Source-PE and RP-PE MUST create a PIM (S,G) state as a result of PIM (S,G) join message processing, PIM (S,G) state at Source-PE MUST have the MI-PMSI(MVPN\_ID) as OIF, PIM (S,G) state at RP-PE MUST inherit all OIF from the previous (\*,G) state, and adds the MI-PMSI(MVPN\_ID) as IIF. Note, the OIF for old (\*,G) state has had the MI-PMSI(MVPN\_ID) as OIF, this OIF MUST NOT be inherited for (S,G).
- o At Source-PE, the multicast traffic received from a multicast source behind Source-PE, MUST be forwarded through the source-PE to RP-PE mLSP represented by the OIF of MI-PMSI(MVPN\_ID). The "source-PE to RP-PE mLSP" is the default mLSP. Meanwhile, multicast source S still embeds the traffic as the PIM register message and send it to RP through the unicast MPLS tunnel.
- o After the RP-PE receives the traffic from the source-PE to RP-PE mLSP (default mLSP) during the PIM register process, following events SHOULD happen
  1. RP-PE SHOULD forward the traffic to RP.
  2. After RP receives the native multicast traffic from the interface which was used to forward the PIM (S,G) join message to multicast source S, RP SHOULD stop de-capsulating the PIM register message. All received PIM register message will be discarded.
  3. RP Sends a PIM register-stop (unicast) message to multicast source S.
- o After the multicast source receives register-stop message, it MUST stop to send PIM register message to RP, and all multicast data is natively forwarded by the (S,G) state to flood to the source-PE to

RP-PE mLSP, or the default mLSP.

#### 5.4. SPT switching

After Receiver-PE receive multicast traffic from the default mLSP. Each Receiver-PE SHOULD forward the traffic to some attached CEs by the PIM state (\*,G) created when the PIM (\*,G) join was received from the attached CEs.

After the traffic reaches the Last Hop Router (LHR), LHR can initialize the Shortest Path Tree (SPT) switching by checking the traffic rate. If the rate exceeds the pre-configured threshold, LHR SHOULD send the PIM (S,G) join to the multicast source.

With the above solution for SPT switching, the Receiver-PE MUST still forward the PIM (S,G) join to the default mLSP. And the PIM (S,G) state SHOULD be created at the Receiver-PE and the state SHOULD inherit all Olist from the previously created (\*,G) state.

As a result of this SPT switching solution, only Receiver-PE has the PIM state change. The traffic will be forwarded by (S,G) instead of (\*,G). Source-PE has no change to the PIM state (S,G). There is no MPLS LSP changes for the traffic forwarding path in MPLS core network. The traffic is still forwarded to the default mLSP at source-PE.

#### 5.5. RPT prune

Using the above method, the SPT switching does not lead to the traffic receiving interface change on the receiver PE, so, there is no RPT prune message triggered.

#### 5.6. Data mLSP switching

As described in above section, the SPT switching does not change the MPLS path for multicast forwarding. Some receiver-PEs still receive the traffic even there is no intention to join the specific group G. We will use data mLSP switching to serve the similar purpose for MPLS network as SPT switching in IP network. By data mLSP switching, the multicast forwarding path in MPLS network can be changed from a shared tree (default mLSP) to a

- o Shortest MPLS path from receiver to source, if the explicit path is not configured for the source S, and the QoS is not required.
- o User defined path, if the explicit path is configured for the source S, or the QoS reservation is required.

If the traffic rate for the stream (S,G) exceeds the threshold, the Source-PE MUST flood the mLSP join TLV to all PEs. Each PE, if it has already created the PIM state for group G, MUST initialize a mRSVP-TE P2MP path message to the Source-PE. The Source-PE is found by the BGP next hop address for S.

The Receiver-PE MUST update its IIF for state (S,G) from MI-PMSI(MVPN\_ID) to S-PMSI(MVPN\_ID, mLSP\_ID).

After the data mLSP is constructed at Source-PE, the PIM state (S,G) MUST add S-PMSI(MVPN\_ID, mLSP\_ID) to its Olist and prune the old OIF MI-PMSI(MVPN\_ID). Note, at this moment, the traffic is still sent to the default mLSP from the Source-PE.

As a result of OIF updating for (S,G) at Source-PE, the traffic is switched from the default mLSP to the data mLSP for (S,G).

#### 5.7. PIM state at Receiver-PE

PIM state at Receiver-PE may be different due to the rate threshold configuration of SPT switching and data mLSP switching.

- o If the rate threshold for mLSP data switching is less than the rate threshold for SPT switching, the data mLSP will be switched earlier than the SPT switching in IP. The multicast distribution tree in MPLS could be switched to a shortest path tree but the tree in IP network is still a shared tree. As a result, the traffic is carried by a P2MP tunnel in MPLS network. But at the receiver-PE, the de-capsulated MPLS traffic MUST be still forwarded by a PIM state (\*,G) which is corresponding to a shared tree.
- o If the rate threshold for mLSP data switching is greater than the rate threshold for SPT switching, the data mLSP will be switched later than the SPT switching in IP. The tree in IP network is switched to a shortest path tree but the multicast distribution tree in MPLS is still a default mLSP. So, the traffic is carried by a MP2MP tunnel in MPLS network, and at the receiver-PE, the de-capsulated MPLS traffic will be forwarded by a PIM state (S,G) which is corresponding to a shortest path tree.

#### 6. Aggregation

With the method described above, there is one data mLSP per multicast stream (S,G). This may not be feasible if the stream number is big, or, there is limit for MPLS label for multicast in a network. Under those scenarios, traffic aggregation in MPLS network is desired.

Aggregation can save the MPLS tunnel, but always with trade off. When multiple MPLS multicast trees are not completely overlapped, to aggregate them will lead to some sub-LSP waste the bandwidth. For example, if two trees have different set of receiver-PEs, some traffic has to be dropped on a PE if it does not have the (S,G) state created before.

#### 6.1. Aggregation by Default mLSP

The aggregation by the default mLSP is straightforward. If we do not set the data mLSP for any (S,G), the traffic of (S,G) will be kept in the default mLSP forever. The aggregation for selective (S,G) can be done at CLI level by ACL (Access List) or any other kind of tool to make the streams which satisfy some conditions to stay in the default mLSP.

#### 6.2. Aggregation by Data mLSP

The aggregation by the data mLSP can be achieved by the following ways

- o Source-PE assigns the same mLSP ID for the streams expected to be aggregated, and keeps the mapping for the mLSP ID to different (S,G).
- o Source-PE floods the mLSP join TLV for each (S,G) with the same mLSP ID to default mLSP.
- o Receiver-PE receives the mLSP join TLV and checks if the data mLSP corresponding to the mLSP ID is created already. If yes, receiver-PE SHOULD only update its mapping for the mLSP ID to an new (S,G) but SHOULD NOT initialize the new path message to source-PE.
- o Source-PE SHOULD switch multiple streams which were assigned with the same mLSP ID to the same data mLSP after it is created.

The aggregation by data mLSP essentially aggregates multicast streams which share the same source-PE even they have different multicast source.

### 7. Non-VPN multicast support

The method specified in this document can also apply to the non-VPN multicast support.

For the non-VPN multicast case, we will take the same approach as VPN

multicast case. Basically, we will treat the public table with a special VPN ID = 0 (see section 9 for VPN ID). By such treatment, the multicast in public domain becomes the multicast in a special table, and it can be supported as normal mVPN.

## 8. Message Format and Constants

Two types of new message format have to be introduced to support mVPN by mRSVP-TE. One is the SESSION-object message format for mRSVP-TE. Another is the mLSP join TLV message format.

The SESSION-object defined in mRSVP-TE is a opaque value (Fig. 7 in [I-D.lzj-mpls-receiver-driven-multicast-rsvp-te]). So, we can define the details of the opaque value based on the mVPN requirement. For the PIM-SSM support option 1 (Fig. 1, Fig. 2), we can embed the "c-source" and "c-group" directly into the SESSION-object. But for PIM-SM support, and PIM-SSM support option 2/3, we can embed the "mLSP ID" into the SESSION-object (Fig. 3). "mLSP ID" can map to different "c-source" and "c-group" at PEs.

The mLSP join TLV message format is similar to the MDT defined in section 7.2 and 7.3 [RFC6037]. Both use UDP to encapsulate the join TLV message, and the IP destination address are also same, it is ALL-PIM-ROUTERS (224.0.0.13) for IPv4 and ALL-PIM-ROUTERS (ff02::d) for IPv6. But there are some difference for MDT and mLSP. The 1st difference is that we have to use a value (mLSP ID) other than "P Group" for MPLS case since we do not have "P Group" for MPLS core network. The 2nd difference is that we have to use different port number instead of 3232 assigned to mGRE based MDT. The exact UDP port number is TBD.

### 8.1. Path session object for PIM-SSM option 1 (IPv4)

The MVPN mRSVP path message for PIM-SSM option 1 for IPv4 has the following format.

Class = SESSION, mRSVP\_TE\_MVPN\_IPv4 C-Type = TBD



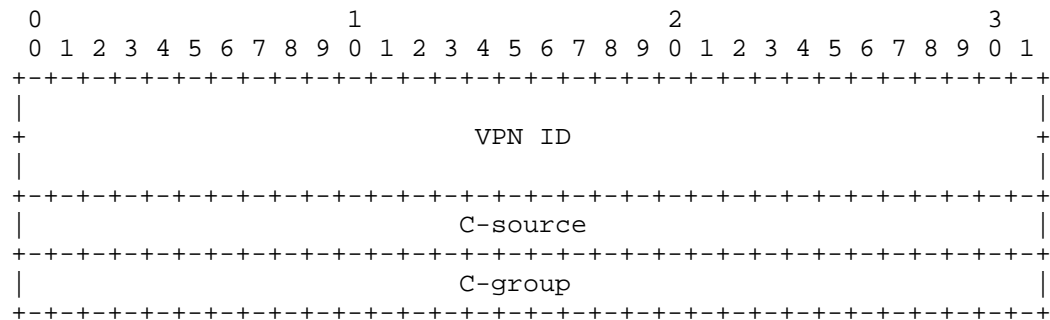


Fig. 1 mVPN mRSVP-TE path session object for PIM-SSM option 1 (IPv4)

VPN ID:

This is the ID for MVPN, it MUST be same for the same VPN cross a MPLS network. VRF RD (Route Distinguisher) or any other unique value in a MPLS network can be used for VPN ID. 0 is reserved for the global MPLS multicast or non MVPN case

C-source (32 bits):

the IPv4 address of the traffic source in the VPN.

C-group (32 bits):

the IPv4 address of the multicast traffic destination address in the VPN.

## 8.2. Path session object for PIM-SSM option 1 (IPv6)

The MVPN mRSVP path message for PIM-SSM option 1 for IPv6 has the following format.

Class = SESSION, mRSVP\_TE\_MVPN\_IPv6 C-Type = TBD

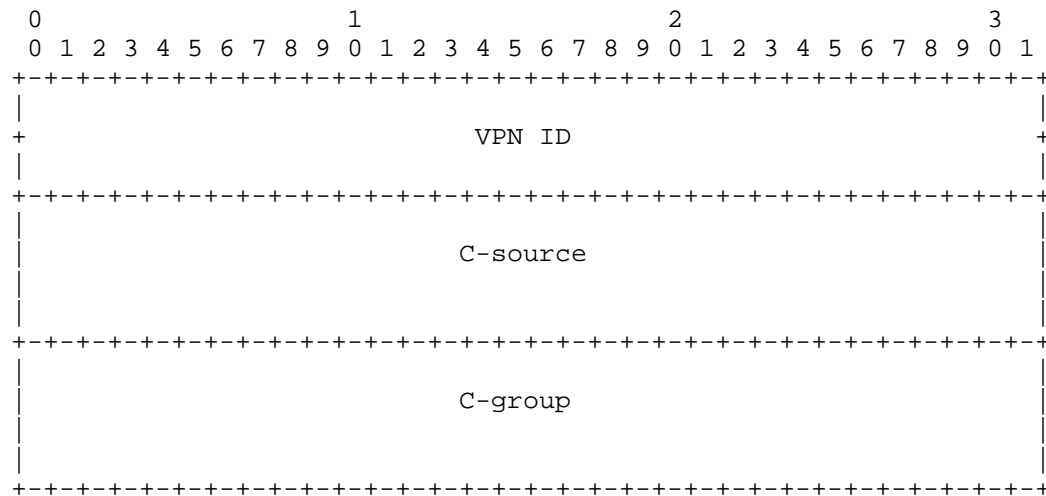


Fig. 2 mVPN mRSVP-TE path session object for PIM-SSM option 1 (IPv6)

VPN ID:

Same definition as in Fig. 1

C-source (128 bits):

the IPv6 address of the traffic source in the VPN.

C-group (128 bits):

the IPv6 address of the multicast traffic destination address in the VPN.

### 8.3. Path session object for other PIM modes (IPv4)

The MVPN mRSVP-TE path message for PIM-SM, PIM-SSM (Option 2 and 3) for IPv4 has the following format.

Class = SESSION, mRSVP-TE-MVPN-IPv4 C-Type = TBD

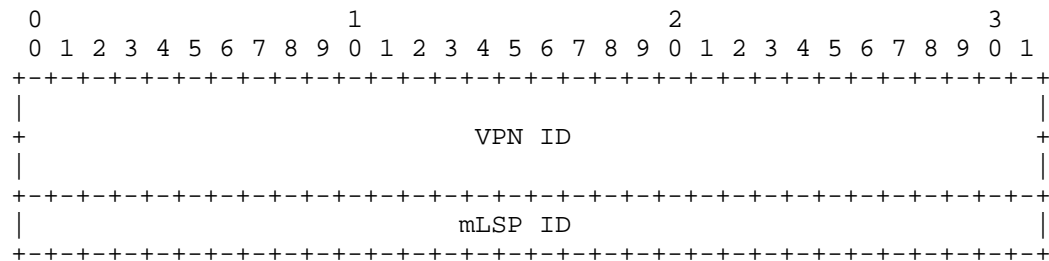


Fig. 3 mVPN mRSVP-TE path session object for PIM-SM, PIM-SSM (Option 2 and 3)

VPN ID:

Same definition as in Fig. 1

mLSP ID:

the mLSP ID corresponding to a tunnel, 0 is reserved for default mLSP. For all non-zero mLSP ID, it SHOULD come from the mLSP join TLV message, see Fig. 4 and Fig. 5

#### 8.4. Path session object for other PIM modes (IPv6)

The MVPN mRSVP-TE path message for PIM-SM, PIM-SSM (Option 2 and 3) for IPv6 has the following format.

Class = SESSION, mRSVP\_TE\_MVPN\_IPv6 C-Type = TBD

The session object format is the same as for IPv4 shown in Fig 3.

#### 8.5. mLSP TLV Message format for IPv4

The mLSP Join TLV for IPv4 streams has the following format.

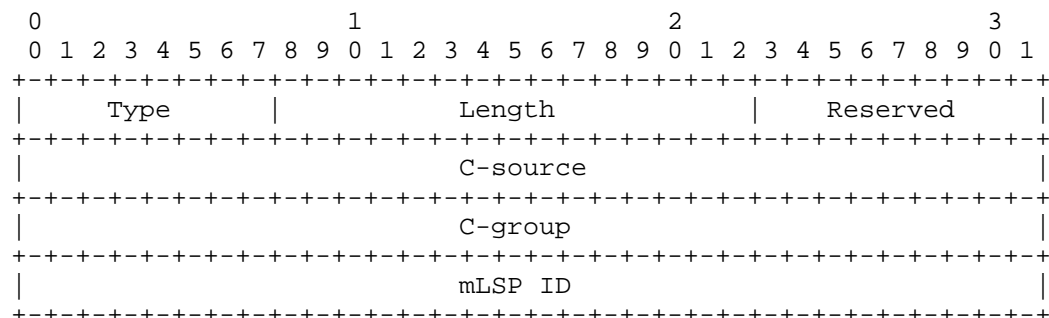


Fig. 4 mLSP join TLV message format for IPv4

Type (8 bits):  
MUST be set to 1.

Length (16 bits):  
MUST be set to 16.

Reserved (8 bits):  
for future use.

C-source (32 bits):  
the IPv4 address of the traffic source in the VPN.

C-group (32 bits):  
the IPv4 address of the multicast traffic destination address in the VPN.

mLSP ID (32 bits):  
the mLSP ID corresponding to the data mLSP carrying the flow (C-source, C-group).

#### 8.6. mLSP TLV Message format for IPv6

The mLSP Join TLV for IPv6 streams has the following format.

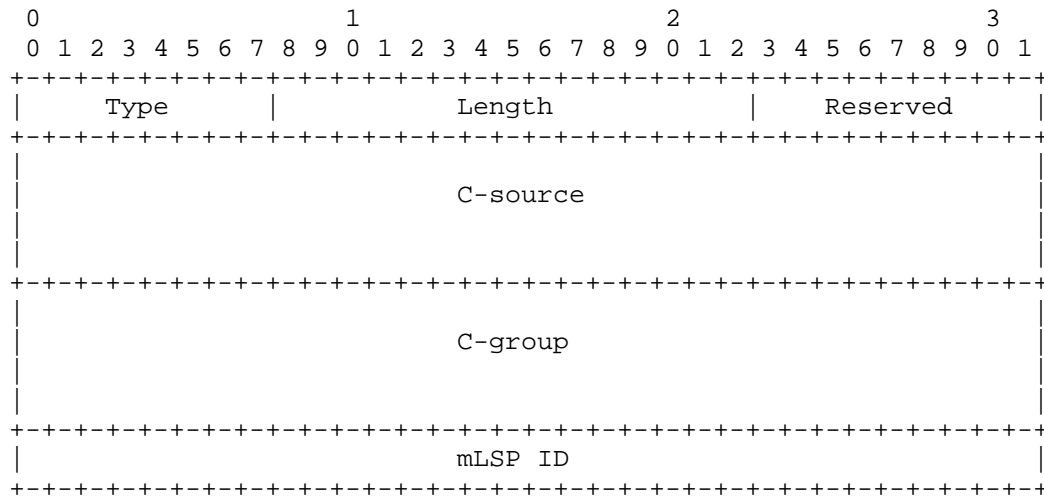


Fig. 5 mLSP join TLV message format for IPv6

Type (8 bits):  
MUST be set to 4.

Length (16 bits):  
MUST be set to 40.

Reserved (8 bits):  
for future use.

C-source (128 bits):  
the IPv6 address of the traffic source in the VPN.

C-group (128 bits):  
the IPv6 address of the multicast traffic destination address  
in the VPN.

mLSP ID (32 bits):  
the mLSP ID corresponding to the data mLSP carrying the flow  
(C-source, C-group).

## 9. Acknowledgements

We would like to thank Katherine Zhao and Quintin Zhao for comments on early drafts of this work.

## 10. IANA Considerations

There is no change with regards to IANA

## 11. Security Considerations

There is no change with regards to the security for PIM protocol and mRSVP-TE after the MVPN is provided

## 12. References

### 12.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4875] Aggarwal, R., Papadimitriou, D., and S. Yasukawa, "Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label

Switched Paths (LSPs)", RFC 4875, May 2007.

- [RFC6388] Wijnands, IJ., Minei, I., Kompella, K., and B. Thomas, "Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths", RFC 6388, November 2011.
- [RFC6513] Rosen, E. and R. Aggarwal, "Multicast in MPLS/BGP IP VPNs", RFC 6513, February 2012.
- [RFC4601] Fenner, B., Handley, M., Holbrook, H., and I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601, August 2006.
- [RFC5059] Bhaskar, N., Gall, A., Lingard, J., and S. Venaas, "Bootstrap Router (BSR) Mechanism for Protocol Independent Multicast (PIM)", RFC 5059, January 2008.
- [RFC6514] Aggarwal, R., Rosen, E., Morin, T., and Y. Rekhter, "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", RFC 6514, February 2012.
- [I-D.lzj-mpls-receiver-driven-multicast-rsvp-te]  
Li, R., Zhao, Q., and C. Jacquenet, "Receiver-Driven Multicast Traffic-Engineered Label-Switched Paths", draft-lzj-mpls-receiver-driven-multicast-rsvp-te-01 (work in progress), July 2012.

## 12.2. Informative References

- [RFC6037] Rosen, E., Cai, Y., and IJ. Wijnands, "Cisco Systems' Solution for Multicast in BGP/MPLS IP VPNs", RFC 6037, October 2010.

## Authors' Addresses

Lin Han (editor)  
Huawei Technologies  
2330 Central Expressway  
Santa Clara, CA 95050  
USA

Phone: +10 408 330 4613  
Email: lin.han@huawei.com

Renwei Li  
Huawei Technologies  
2330 Central Expressway  
Santa Clara, CA 95050  
USA

Phone:  
Email: renwei.li@huawei.com





L3VPN Working Group  
Internet Draft  
Intended Status: Standards Track  
Expires: December 2, 2014  
Updates: 6513,6625

Eric C. Rosen (Editor)  
IJsbrand Wijnands  
Cisco Systems, Inc.

Yiqun Cai  
Microsoft

Arjen Boers

June 2, 2014

## MVPN: Using Bidirectional P-Tunnels

draft-ietf-l3vpn-mvpn-bidir-08.txt

### Abstract

A set of prior RFCs specify procedures for supporting multicast in BGP/MPLS IP VPNs. These procedures allow customer multicast data to travel across a service provider's backbone network through a set of multicast tunnels. The tunnels are advertised in certain BGP multicast "auto-discovery" routes, by means of a BGP attribute known as the "Provider Multicast Service Interface (PMSI) Tunnel attribute". Encodings have been defined that allow the PMSI Tunnel attribute to identify bidirectional (multipoint-to-multipoint) multicast distribution trees. However, the prior RFCs do not provide all the necessary procedures for using bidirectional tunnels to support multicast VPNs. This document updates RFCs 6513 and 6625 by specifying those procedures. In particular, it specifies the procedures for assigning customer multicast flows (unidirectional or bidirectional) to specific bidirectional tunnels in the provider backbone, for advertising such assignments, and for determining which flows have been assigned to which tunnels.

### Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

#### Copyright and License Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1	Introduction .....	4
1.1	Terminology .....	4
1.2	Overview .....	9
1.2.1	Bidirectional P-tunnel Technologies .....	10
1.2.2	Reasons for Using Bidirectional P-tunnels .....	10
1.2.3	Knowledge of Group-to-RP and/or Group-to-RPA Mappings ..	11
1.2.4	PMSI Instantiation Methods .....	12
2	The All BIDIR-PIM Wild Card .....	14
3	Using Bidirectional P-Tunnels .....	15
3.1	Procedures Specific to the Tunneling Technology .....	15
3.1.1	BIDIR-PIM P-Tunnels .....	15
3.1.2	MP2MP LSPs .....	16
3.2	Procedures Specific to the PMSI Instantiation Method ..	16
3.2.1	Flat Partitioning .....	17
3.2.1.1	When an S-PMSI is a 'Match for Transmission' .....	18
3.2.1.2	When an I-PMSI is a 'Match for Transmission' .....	19
3.2.1.3	When an S-PMSI is a 'Match for Reception' .....	20
3.2.1.4	When an I-PMSI is a 'Match for Reception' .....	21
3.2.2	Hierarchical Partitioning .....	21
3.2.2.1	Advertisement of PE Distinguisher Labels .....	23
3.2.2.2	When an S-PMSI is a 'Match for Transmission' .....	24
3.2.2.3	When an I-PMSI is a 'Match for Transmission' .....	25
3.2.2.4	When an S-PMSI is a 'Match for Reception' .....	25
3.2.2.5	When an I-PMSI is a 'Match for Reception' .....	26
3.2.3	Unpartitioned .....	27
3.2.3.1	When an S-PMSI is a 'Match for Transmission' .....	29
3.2.3.2	When an S-PMSI is a 'Match for Reception' .....	29
3.2.4	Minimal Feature Set for Compliance .....	30
4	IANA Considerations .....	30
5	Security Considerations .....	30
6	Acknowledgments .....	31
7	Authors' Addresses .....	31
8	Normative References .....	32
9	Informative References .....	32

## 1. Introduction

The RFCs that specify multicast support for BGP/MPLS IP VPNs ([MVPN], [MVPN-BGP], [MVPN-WILDCARDS]) allow customer multicast data to be transported across a service provider's network through a set of multicast tunnels. These tunnels are advertised in BGP multicast "auto-discovery" (A-D) routes, by means of a BGP attribute known as the "Provider Multicast Service Interface (PMSI) Tunnel" attribute. The base specifications allow the use of bidirectional (multipoint-to-multipoint) multicast distribution trees, and describe how to encode the identifiers for bidirectional trees into the PMSI Tunnel attribute. However, those specifications do not provide all the necessary detailed procedures for using bidirectional tunnels; the full specification of these procedures was considered to be outside the scope of those documents. The purpose of this document is to provide all the necessary procedures for using bidirectional trees in a service provider's network to carry the multicast data of VPN customers.

### 1.1. Terminology

This document uses terminology from [MVPN] and, in particular, uses the prefixes "C-" and "P-", as specified in Section 3.1 of [MVPN], to distinguish addresses in the "customer address space" from addresses in the "provider address space". The following terminology and acronyms are particularly important in this document:

- MVPN

Multicast Virtual Private Network -- a VPN [L3VPN] in which multicast service is offered.

- VRF

VPN Routing and Forwarding table [L3VPN].

- PE

A Provider Edge router, as defined in [L3VPN].

- LSP

An MPLS Label Switched Path.

- P2MP Point-to-Multipoint.

- MP2MP

Multipoint-to-multipoint.

- Unidirectional

Adjective for a multicast distribution tree in which all traffic travels downstream from the root of the tree. Traffic can enter a unidirectional tree only at the root. A P2MP LSP is one type of unidirectional tree. Multicast distribution trees set up by PIM-SM [PIM] are also unidirectional trees.

Data traffic traveling along a unidirectional multicast distribution tree is sometimes referred to in this document as "unidirectional traffic".

- Bidirectional

Adjective for a multicast distribution tree in which traffic may travel both upstream (towards the root) and downstream (away from the root). Traffic may enter a bidirectional tree at any node. A MP2MP LSP is one type of bidirectional tree. Multicast distribution trees created by BIDIR-PIM [BIDIR-PIM] are also bidirectional trees.

Data traffic traveling along a bidirectional multicast distribution tree is sometimes referred to in this document as "bidirectional traffic".

- P-tunnel

A tunnel through the network of one or more Service Providers (SPs). In this document, the P-tunnels we speak of are instantiated as bidirectional multicast distribution trees.

- C-S

Multicast Source. A multicast source address, in the address space of a customer network.

- C-G

Multicast Group. A multicast group address (destination address) in the address space of a customer network. When used without qualification, "C-G" may refer to either a unidirectional group address or a bidirectional group address.

- C-G-BIDIR

A bidirectional multicast group address (i.e., a group address whose IP multicast distribution tree is built by BIDIR-PIM).

- C-multicast flow or C-flow

A customer multicast flow. A C-flow travels through VPN customer sites on a multicast distribution tree set up by the customer. These trees may be unidirectional or bidirectional, depending upon the multicast routing protocol used by the customer. A C-flow travels between VPN customer sites by traveling through P-tunnels.

A C-flow from a particular customer source is identified by the ordered pair (source address, group address), where each address is in the customer's address space. The identifier of such a C-flow is usually written as (C-S,C-G).

If a customer uses the "Any Source Multicast" (ASM) model, the some or all of the customer's C-flows may be traveling along the same "shared tree". In this case, we will speak of a "(C-\*,C-G)" flow to refer to a set of C-flows that travel along the same shared tree in the customer sites.

- C-BIDIR flow or bidirectional C-flow

A C-flow that, in the VPN customer sites, travels along a bidirectional multicast distribution tree. The term "C-BIDIR flow" indicates that the customer's bidirectional tree has been set up by BIDIR-PIM.

- RP

A "Rendezvous Point", as defined in [PIM].

- C-RP

A Rendezvous Point whose address is in the customer's address space.

- RPA

A "Rendezvous Point Address", as defined in [BIDIR-PIM].

- C-RPA

An RPA in the customer's address space.

- P-RPA

An RPA in the Service Provider's address space

- Selective P-tunnel

A P-tunnel that is joined only by Provider Edge (PE) routers that need to receive one or more of the C-flows that are traveling through that P-tunnel.

- Inclusive P-tunnel

A P-tunnel that is joined by all PE routers that attach to sites of a given MVPN.

- Intra-AS I-PMSI A-D route

Intra Autonomous System Inclusive Provider Multicast Service Interface Auto-Discovery route. Carried in BGP Update messages, these routes can be used to advertise the use of Inclusive P-tunnels. See [MVPN-BGP] section 4.1.

- S-PMSI A-D route

Selective Provider Multicast Service Interface Auto-Discovery route. Carried in BGP Update messages, these routes are used to advertise the fact that a particular C-flow or a particular set of C-flows is bound to (i.e., is traveling through) a particular P-tunnel. See [MVPN-BGP] section 4.3.

- (C-S,C-G) S-PMSI A-D route

An S-PMSI A-D route whose NLRI ("Network Layer Reachability Information") contains C-S in its "Multicast Source" field and C-G in its "Multicast Group" field.

- (C-\*,C-G) S-PMSI A-D route

An S-PMSI A-D route whose NLRI contains the wildcard (C-\*) in its "Multicast Source" field and C-G in its "Multicast Group" field. See [MVPN-WILDCARDS].

- (C-\*,C-G-BIDIR) S-PMSI A-D route

An S-PMSI A-D route whose NLRI contains the wildcard (C-\*) in its "Multicast Source" field and C-G-BIDIR in its "Multicast Group" field. See [MVPN-WILDCARDS].

- (C-\*,C-\*) S-PMSI A-D route

An S-PMSI A-D route whose NLRI contains the wildcard C-\* in its "Multicast Source" field and the wildcard C-\* in its "Multicast Group" field. See [MVPN-WILDCARDS].

- (C-\*,C-\*) S-PMSI A-D route

An S-PMSI A-D route whose NLRI contains the wildcard C-\* in its "Multicast Source" field and the wildcard C-\* in its "Multicast Group" field. See [MVPN-WILDCARDS].

- (C-\*,C-\*-BIDIR) S-PMSI A-D route

An S-PMSI A-D route whose NLRI contains the wildcard C-\* in its "Multicast Source" field and the wildcard "C-\*-BIDIR" in its "Multicast Group" field. See section 2 of this document.

- (C-S,C-\*) S-PMSI A-D route

An S-PMSI A-D route whose NLRI contains C-S in its "Multicast Source" field and the wildcard C-\* in its "Multicast Group" field. See [MVPN-WILDCARDS].

- Wildcard S-PMSI A-D route

A (C-\*,C-G) S-PMSI A-D route, or a (C-\*,C-\*) S-PMSI A-D route, or a (C-S,C-\*) S-PMSI A-D route, or a (C-\*,C-\*-BIDIR) S-PMSI A-D route.

- PTA

PMSI Tunnel attribute, a BGP attribute that identifies a P-tunnel. See [MVPN-BGP] section 8.

The terminology used for categorizing S-PMSI A-D routes will also be used for categorizing the S-PMSIs advertised by those routes. E.g., the S-PMSI advertised by a (C-\*,C-G) S-PMSI A-D route will be known as a "(C-\*,C-G) S-PMSI".

Familiarity with multicast concepts and terminology [PIM] is also



presupposed.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document, when and only when appearing in all caps, are to be interpreted as described in [RFC2119].

## 1.2. Overview

The base documents for MVPN ([MVPN], [MVPN-BGP]) define a "PMSI Tunnel attribute" (PTA). This is a BGP Path Attribute that may be attached to the BGP "I-PMSI A-D routes" and "S-PMSI A-D routes" that are defined in those documents. The base documents define the way in which the identifier of a bidirectional P-tunnel is to be encoded in the PTA. However, those documents do not contain the full set of specifications governing the use bidirectional P-tunnels; rather, those documents declare the full set of specifications for using bidirectional P-tunnels to be outside their scope. Similarly, the use of bidirectional P-tunnels advertised in wildcard S-PMSI A-D routes is declared by [MVPN-WILDCARDS] to be "out of scope."

This document provides the specifications governing the use of bidirectional P-tunnels to provide MVPN support. This includes the procedures for assigning C-flows to specific bidirectional P-tunnels, for advertising the fact that a particular C-flow has been assigned to a particular bidirectional P-tunnel, and for determining the bidirectional P-tunnel on which a given C-flow may be expected.

The C-flows carried on bidirectional P-tunnels may themselves be either unidirectional or bidirectional. Procedures are provided for both cases.

This document does not specify any new data encapsulations for bidirectional P-tunnels. Section 12 ("Encapsulations") of [MVPN] applies unchanged.

With regard to the procedures for using bidirectional P-tunnels to instantiate PMSIs, if there is any conflict between the procedures specified in this document and the procedures of [MVPN], [MVPN-BGP], or [MVPN-WILDCARDS], the procedures of this document take precedence.

The use of bidirectional P-tunnels to support extranets [MVPN-XNET] is outside the scope of this document. The use of bidirectional P-tunnels as "segmented P-tunnels" (see [MVPN] section 8 and various sections of [MVPN-BGP]) is also outside the scope of this document.

### 1.2.1. Bidirectional P-tunnel Technologies

This document supports two different technologies for creating and maintaining bidirectional P-tunnels:

- Multipoint-to-multipoint Label Switched Paths (MP2MP LSPs) that are created through the use of the Label Distribution Protocol (LDP) Multipoint-to-Multipoint extensions [mLDP].
- Multicast distribution trees that are created through the use of BIDIR-PIM [BIDIR-PIM].

An implementation may be considered compliant with this document if it provides either one of these tunneling technologies. Other bidirectional tunnel technologies are outside the scope of this document.

### 1.2.2. Reasons for Using Bidirectional P-tunnels

Bidirectional P-tunnels can be used to instantiate I-PMSIs and/or S-PMSIs.

An SP may decide to use bidirectional P-tunnels to instantiate certain I-PMSIs and/or S-PMSIs in order to provide its customers with C-BIDIR support, using the "Partitioned Set of PEs" technique discussed in [MVPN] section 11.2 and [RFC6517] section 3.6. This technique can be used whether the C-BIDIR flows are being carried on an I-PMSI or an S-PMSI.

Even if an SP does not need to provide C-BIDIR support, it may still decide to use bidirectional P-tunnels, in order to save state in the network's transit nodes. For example, if an MVPN has *n* PEs attached to sites with multicast sources, and there is an I-PMSI for that MVPN, instantiating the I-PMSI with unidirectional P-tunnels (i.e., with P2MP multicast distribution trees) requires *n* multicast distribution trees, each one rooted at a different PE. If the I-PMSI is instantiated by a bidirectional P-tunnel, a single multicast distribution tree can be used.

An SP may decide to use bidirectional P-tunnels for either or both of these reasons. Note that even if the reason for using bidirectional P-tunnels is to provide C-BIDIR support, the same P-tunnels can also be used to carry unidirectional C-flows, if that is the choice of the SP.

These two reasons for using bidirectional P-tunnels may appear to be somewhat in conflict with each other, since (as will be seen in

subsequent sections), the use of bidirectional P-tunnels for C-BIDIR support may require multiple bidirectional P-tunnels per VPN. Each such P-tunnel is associated with a particular "distinguished PE", and can only carry those C-BIDIR flows whose C-RPAs are reachable through its distinguished PE. However, on platforms that support MPLS upstream-assigned labels [RFC5331], "PE Distinguisher Labels" can be used to aggregate multiple bidirectional P-tunnels onto a single "outer" bidirectional P-tunnel, thereby allowing one to provide C-BIDIR support with minimal state at the transmit nodes.

Since there are two fundamentally different reasons for using bidirectional P-tunnels, and since many deployed router platforms do not support upstream-assigned labels at the current time, this document specifies several different methods of using bidirectional P-tunnels to instantiate PMSIs. We refer to these as "PMSI Instantiation Methods". The method or methods deployed by any particular SP will depend upon that SP's goals and engineering tradeoffs, and upon the set of platforms deployed by that SP.

The rules for using bidirectional P-tunnels in I-PMSI or S-PMSI A-D routes are not exactly the same as the rules for using unidirectional P-tunnels, and the rules are also different for the different PMSI instantiation methods. Subsequent sections of this document specify the rules in detail.

### 1.2.3. Knowledge of Group-to-RP and/or Group-to-RPA Mappings

If a VPN customer is making use of a particular "Any Source Multicast" (ASM) group address, the PEs of that VPN generally need to know the group-to-RP mappings that are used within the VPN. If a VPN customer is making use of BIDIR-PIM group addresses, the PEs need to know the group-to-RPA mappings that are used within the VPN. Commonly, the PEs obtain this knowledge either through provisioning or by participating in a dynamic "group-to-RP(A) mapping discovery protocol" that runs within the VPN. However, the way in which this knowledge is obtained is outside the scope of this document.

The PEs also need to be able to forward traffic towards the C-RPs and/or C-RPAs, and to determine whether the next hop "interface" of the route to a particular C-RP(A) is a VRF interface or a PMSI. This is done by applying the procedures of [MVPN] section 5.1.

#### 1.2.4. PMSI Instantiation Methods

This document specifies three methods for using bidirectional P-tunnels to instantiate PMSIs: the Flat Partitioned Method, the Hierarchical Partitioned Method, and the Unpartitioned Method.

##### - Partitioned Methods

In the Partitioned Methods, a particular PMSI is instantiated by a set of bidirectional P-tunnels. These P-tunnels may be aggregated (as "inner" P-tunnels) into a single "outer" bidirectional P-tunnel ("Hierarchical Partitioning"), or they may be unaggregated ("Flat Partitioning"). Any PE that joins one of these P-tunnels can transmit a packet on it, and the packet will be received by all the other PEs that have joined the P-tunnel. For each such P-tunnel (each "inner" P-tunnel, in the case of Hierarchical Partitioning) there is one PE that is its "distinguished PE". When a PE receives a packet from a given P-tunnel, the PE can determine from the packet's encapsulation the P-tunnel it has arrived on, and can thus infer the identity of the distinguished PE associated with the packet. This association plays an important role in the treatment of the packet, as specified later on in this document.

The number of P-tunnels needed (the number of "inner" P-tunnels needed, if Hierarchical Partitioning is used) depends upon a number of factors that are described later in this document.

The Hierarchical Partitioned Method requires the use of upstream-assigned MPLS labels ("PE Distinguisher Labels"), and requires the use of the PE Distinguisher Labels attribute in BGP. The Flat Partitioned Method requires neither of these.

The Partitioned Method (either flat or hierarchical) is a pre-requisite for implementing the "Partitioned Sets of PEs" technique of supporting C-BIDIR, as discussed in [MVPN] section 11.2. The Partitioned Method (either flat or hierarchical) is also a pre-requisite for applying the "Discarding Packets from Wrong PE" technique, discussed in [MVPN] Section 9.1.1, to a PMSI that is instantiated by a bidirectional P-tunnel.

The Flat Partitioned Method is a pre-requisite for implementing the "Partial Mesh of MP2MP P-tunnels" technique for carrying customer bidirectional (C-BIDIR) traffic, as discussed in [MVPN] Section 11.2.3.

The Hierarchical Partitioned Method is a pre-requisite for implementing the "Using PE Distinguisher Labels" technique of

carrying customer bidirectional (C-BIDIR) traffic, as discussed in [MVPN] Section 11.2.2.

Note that a particular deployment may choose to use the Partitioned Method for carrying the C-BIDIR traffic on bidirectional P-tunnels, while carrying other traffic either on unidirectional P-tunnels, or on bidirectional P-tunnels using the Unpartitioned Method. Routers in a given deployment must be provisioned to know which PMSI instantiation method to use for which PMSIs.

There may be ways of implementing the Partitioned Method with PMSIs that are instantiated by unidirectional P-tunnels. (See, e.g., [MVPN-BIDIR-IR].) However, that is outside the scope of the current document.

#### - Unpartitioned Method

In the Unpartitioned Method, a particular PMSI can be instantiated by a single bidirectional P-tunnel. Any PE that joins the tunnel can transmit a packet on it, and the packet will be received by all the other PEs that have joined the tunnel. The receiving PEs can determine the tunnel on which the packet was transmitted, but they cannot determine which PE transmitted the packet, nor can they associate the packet with any particular "distinguished PE".

When the Unpartitioned Method is used, this document does not mandate that only one bidirectional P-tunnel be used to instantiate each PMSI. It allows for the case where more than one P-tunnel is used. In this case, the transmitting PEs will have a choice of which such P-tunnel to use when transmitting, and the receiving PEs must be prepared to receive from any of those P-tunnels. The use of multiple P-tunnels in this case provides additional robustness, but no additional functionality.

I-PMSIs may be instantiated by bidirectional P-tunnels using either the Partitioned (either Flat or Hierarchical) or the Unpartitioned Method. The method used for a given MVPN is determined by provisioning. It SHOULD be possible to provision this on a per-MVPN basis, but all the VRFs of a single MVPN MUST be provisioned to use the same method for the given MVPN's I-PMSI.

If a bidirectional P-tunnel is used to instantiate an S-PMSI (including the case of a (C-\*,C-\*) S-PMSI), either the Partitioned Method (either Flat or Hierarchical) or the Unpartitioned Method may be used. The method used by a given VRF used is determined by provisioning. It SHOULD be possible to provision this on a per-MVPN

basis, but all the VRFs of a single MVPN MUST be provisioned to use the same method for those of their S-PMSIs that are instantiated by bidirectional P-tunnels.

If the Partitioned Method is used, all the VRFs of a single MVPN MUST be provisioned to use the same variant of the Partitioned Method, i.e., either they must all use the Flat Partitioned Method, or they must all use the Hierarchical Partitioned Method.

It is valid to use the Unpartitioned Method to instantiate the I-PMSIs, while using one of the Partitioned Methods to instantiate the S-PMSIs.

It is valid to instantiate some S-PMSIs by unidirectional P-tunnels and others by bidirectional P-tunnels.

The procedures for the use of bidirectional P-tunnels, specified in subsequent sections of this document, depend on both the tunnel technology and on the PMSI instantiation method. Note that this document does not necessarily specify procedures for every possible combination of tunnel technology and PMSI instantiation method.

## 2. The All BIDIR-PIM Wild Card

When an MVPN customer is using BIDIR-PIM, it is useful to be able to advertise an S-PMSI A-D route whose semantics are: "by default, all BIDIR-PIM C-multicast traffic (within a given VPN) that has not been bound to any other P-tunnel is bound to the bidirectional P-tunnel identified by the PTA of this route". This can be especially useful if one is using a bidirectional P-tunnel to carry the C-BIDIR flows, while using unidirectional P-tunnels to carry other C-flows. To do this, it is necessary to have a way to encode a (C-\*,C-\*) wildcard that is restricted to BIDIR-PIM C-groups.

We therefore define a special value of the group wildcard, whose meaning is "all BIDIR-PIM groups". The "BIDIR-PIM groups wildcard" is encoded as a group field whose length is 8 bits and whose value is zero. That is, the "multicast group length" field contains the value 0x08, and the "multicast group" field is a single octet containing the value 0x00. We will use the notation (C-\*,C-\*-BIDIR) to refer to the "all BIDIR-PIM groups" wildcard.

### 3. Using Bidirectional P-Tunnels

A bidirectional P-tunnel may be advertised in the PTA of an Intra-AS I-PMSI A-D route or in the PTA of an S-PMSI A-D route. The advertisement of a bidirectional P-tunnel in the PTA of an Inter-AS I-PMSI A-D route is outside the scope of this document.

#### 3.1. Procedures Specific to the Tunneling Technology

This section discusses the procedures that are specific to a given tunneling technology (BIDIR-PIM or MP2MP mLDP), but that are independent of the method (Unpartitioned, Flat Partitioned, or Hierarchical Partitioned) used to instantiate a PMSI.

##### 3.1.1. BIDIR-PIM P-Tunnels

Each BIDIR-PIM P-Tunnel is identified by a unique P-group address [MVPN, section 3.1]. (The P-group address is called a "P-Multicast Group" in [MVPN-BGP]). Section 5 of [MVPN-BGP] specifies the way to identify a particular BIDIR-PIM P-tunnel in the PTA of an I-PMSI or S-PMSI A-D route.

Ordinary BIDIR-PIM procedures are used to set up the BIDIR-PIM P-tunnels. A BIDIR-PIM P-group address is always associated with a unique "Rendezvous Point Address" (RPA) in the SP's address space. We will refer to this as the "P-RPA". Every PE needing to join a particular BIDIR-PIM P-tunnel must be able to determine the P-RPA that corresponds to the P-tunnel's P-group address. To construct the P-tunnel, PIM Join/Prune messages are sent along the path from the PE to the P-RPA. Any P routers along that path must also be able to determine the P-RPA, so that they too can send PIM Join/Prune messages towards it. The method of mapping a P-group address to an RPA may be static configuration, or some automated means of RPA discovery that is outside the scope of this specification.

If a BIDIR-PIM P-tunnel is used to instantiate an I-PMSI or an S-PMSI, it is RECOMMENDED that the path from each PE in the tunnel to the RPA consist entirely of point-to-point links. On a point-to-point link, there is no ambiguity in determining which router is upstream towards a particular RPA, so the BIDIR-PIM "Designated Forwarder Election" is very quick and simple. Use of a BIDIR-PIM P-tunnel containing multiaccess links is possible, but considerably more complex.

The use of BIDIR-PIM P-tunnels to support the Hierarchical Partitioned Method is outside the scope of this document.

When the PTA of an Intra-AS I-PMSI A-D route or an S-PMSI A-D route identifies a BIDIR-PIM tunnel, the originator of the route SHOULD NOT include a PE Distinguisher Labels attribute. If it does, that attribute MUST be ignored. When we say the attribute is "ignored", we do not mean that its normal BGP processing is not done, but that the attribute has no effect on the data plane. It MUST however be treated by BGP as if it were an unsupported optional transitive attribute. (PE Distinguisher Labels are used for the Hierarchical Partitioning Method, but this document does not provide support for the Hierarchical Partitioning Method with BIDIR-PIM P-tunnels.)

### 3.1.2. MP2MP LSPs

Each MP2MP LSP is identified by a unique "MP2MP FEC (Forwarding Equivalence Class) element" [mLDP]. The FEC element contains the IP address of the "root node", followed by an "opaque value" that identifies the MP2MP LSP uniquely in the context of the root node's IP address. This opaque value may be configured or autogenerated, and within an MVPN, there is no need for different root nodes to use the same opaque value. The mLDP specification supports the use of several different ways of constructing the tunnel identifiers. The current specification does not place any restriction on the type of tunnel identifier that might be used. However, a given implementation might not support every possible type of tunnel identifier.

Section 5 of [MVPN-BGP] specifies the way to identify a particular MP2MP P-tunnel in the PTA of an I-PMSI or S-PMSI A-D route.

Ordinary mLDP procedures for MP2MP LSPs are used to set up the MP2MP LSP.

### 3.2. Procedures Specific to the PMSI Instantiation Method

When either the Flat Partitioned Method or the Hierarchical Partitioned Method is used to implement the "Partitioned Sets of PEs" method of supporting C-BIDIR, as discussed in section 11.2 of [MVPN] and section 3.6 of [RFC6517], a C-BIDIR flow MUST be carried only on an I-PMSI or on a (C-\*,C-G-BIDIR), (C-\*,C-\*-BIDIR), or (C-\*,C-\*) S-PMSI. A PE MUST NOT originate any (C-S,C-G-BIDIR) S-PMSI A-D routes. (Though it may of course originate (C-S,C-G) S-PMSI A-D routes for C-G's that are not C-BIDIR groups.) Packets of a C-BIDIR flow MUST NOT be carried on a (C-S,C-\*) S-PMSI.

Sections 3.2.1 and 3.2.2 specify additional details of the two Partitioned Methods.



### 3.2.1. Flat Partitioning

The procedures of this section and its sub-sections apply when (and only when) the Flat Partitioned Method is used. This method is introduced in [MVPN] Section 11.2.3, where it is called "Partial Mesh of MP2MP P-tunnels". This method can be used with MP2MP LSPs or with BIDIR-PIM P-tunnels.

When a PE originates an I-PMSI or S-PMSI A-D route whose PTA specifies a bidirectional P-tunnel, the PE MUST be the root node of the specified P-tunnel. It follows that two different PEs may not advertise the same bidirectional P-tunnel. Any PE that receives a packet from the P-tunnel can infer the identity of the P-tunnel from the packet's encapsulation. Once the identity of the P-tunnel is known, the root node of the P-tunnel is also known. The root node of the P-tunnel on which the packet arrived is treated as the "distinguished PE" for that packet.

If MP2MP LSPs are used, each P-tunnel MUST have a distinct MP2MP FEC (i.e., distinct combination of "root node" and "opaque value"). The PE advertising the tunnel MUST be the same PE identified in the "root node" field of the MP2MP FEC that is encoded in the PTA.

If BIDIR-PIM P-tunnels are used, each advertised P-tunnel MUST have a distinct P-group address. The PE advertising the tunnel will be considered to be the root node of the tunnel. Note that this creates a unique mapping from P-group address to "root node".

The Flat Partitioned Method does not use upstream-assigned labels in the data plane, and hence does not use the BGP PE Distinguisher Labels attribute. When this method is used, I-PMSI and/or S-PMSI A-D routes SHOULD NOT contain a PE Distinguisher Labels attribute; if such an attribute is present in a received I-PMSI or S-PMSI A-D route, it MUST be ignored. (When we say the attribute is "ignored", we do not mean that its normal BGP processing is not done, but that the attribute has no effect on the data plane. It MUST however be treated by BGP as if it were an unsupported optional transitive attribute.)

When the Flat Partitioned Method is used to instantiate the I-PMSIs of a given MVPN, every PE in that MVPN that originates an Intra-AS I-PMSI A-D route MUST include a PTA that specifies a bidirectional P-tunnel. If the intention is to carry C-BIDIR traffic on the I-PMSI, a PE MUST originate an Intra-AS I-PMSI A-D route if one of its VRF interfaces is the next hop interface on its best path to the C-RPA of any bidirectional C-group of the MVPN.

When the Flat Partitioned Method is used to instantiate a (C-\*,C-\*)

S-PMSI, a (C-\*,C-\*-BIDIR) S-PMSI, or a (C-\*,C-G-BIDIR) S-PMSI, a PE that originates the corresponding S-PMSI A-D route MUST include in that route a PTA specifying a bidirectional P-tunnel. Per the procedures of [MVPN] and [MVPN-BGP], a PE will originate such an S-PMSI A-D route only if one of the PE's VRF interfaces is the next hop interface of the PE's best path to the C-RPA of a C-BIDIR group that is to be carried on the specified S-PMSI.

PMSIs that are instantiated via the Flat Partitioned Method may carry customer bidirectional traffic AND customer unidirectional traffic. The rules of sections 3.2.1.1 and 3.2.1.2 determine when a given customer multicast packet is a "match for transmission" to a given PMSI. However, if the "Partitioned Set of PEs" method of supporting C-BIDIR traffic is being used, the PEs must be provisioned in such a way that packets from a C-BIDIR flow never match any PMSI that is not instantiated by a bidirectional P-tunnel. (For example, if the (C-\*,C-\*) S-PMSI were not instantiated by a bidirectional P-tunnel, one could meet this requirement by carrying all C-BIDIR traffic on a (C-\*,C-\*-BIDIR) S-PMSI.)

When a PE receives a customer multicast data packet from a bidirectional P-tunnel, it associates that packet with a "distinguished PE". The distinguished PE for a given packet is the root node of the tunnel from which the packet is received. The rules of section 3.2.1.1 and 3.2.1.2 ensure that:

- If the received packet is part of a unidirectional C-flow, its "distinguished PE" is the PE that transmitted the packet onto the P-tunnel.
- If the received packet is part of a bidirectional C-flow, its "distinguished PE" is not necessarily the PE that transmitted it, but rather the transmitter's "upstream PE" for the C-RPA of the bidirectional C-group.

The rules of sections 3.2.1.3 and 3.2.1.4 allow the receiving PEs to determine the expected distinguished PE for each C-flow, and ensure that a packet will be discarded if its distinguished PE is not the expected distinguished PE for the C-flow to which the packet belongs. This prevents duplication of data for both bidirectional and unidirectional C-flows.

#### 3.2.1.1. When an S-PMSI is a 'Match for Transmission'

Suppose a given PE, say PE1, needs to transmit multicast data packets of a particular C-flow. [MVPN-WILDCARDS] Section 3.1 gives a four-step algorithm for determining the S-PMSI A-D route, if any,

that "matches" that C-flow for transmission.

If the C-flow is not a BIDIR-PIM C-flow, those rules apply unchanged; the remainder of this section applies only to C-BIDIR flows. If a C-BIDIR flow has group address C-G-BIDIR, the rules applied by PE1 are given below:

- If the C-RPA for C-G-BIDIR is a C-address of PE1, or if PE1's route to the C-RPA is via a VRF interface, then:
  - \* If there is a (C-\*,C-G-BIDIR) S-PMSI A-D route currently originated by PE1, then the C-flow matches that route.
  - \* Otherwise, if there is a (C-\*,C-\*-BIDIR) S-PMSI A-D route currently originated by PE1, then the C-flow matches that route.
  - \* Otherwise, if there is a (C-\*,C-\*) S-PMSI A-D route currently originated by PE1, then the C-flow matches that route.
- If PE1 determines the upstream PE for C-G-BIDIR's C-RPA to be some other PE, say PE2, then:
  - \* If there is an installed (C-\*,C-G-BIDIR) S-PMSI A-D route originated by PE2, then the C-flow matches that route.
  - \* Otherwise, if there is an installed (C-\*,C-\*-BIDIR) S-PMSI A-D route originated by PE2, then the C-flow matches that route.
  - \* Otherwise, if there is an installed (C-\*,C-\*) S-PMSI A-D route originated by PE2, then the C-flow matches that route.

If there is an S-PMSI A-D route that matches a given C-flow, and if PE1 needs to transmit packets of that C-flow or other PEs, then it MUST transmit those packets on the bidirectional P-tunnel identified in the PTA of the matching S-PMSI A-D route.

#### 3.2.1.2. When an I-PMSI is a 'Match for Transmission'

Suppose a given PE, say PE1, needs to transmit packets of a given C-flow (of a given MVPN) to other PEs, but according to the conditions of section 3.2.1.1 and/or [MVPN-WILDCARDS] section 3.1, that C-flow does not match any S-PMSI A-D route. Then the packets of the C-flow need to be transmitted on the MVPN's I-PMSI.

If the C-flow is not a BIDIR-PIM C-flow, the P-tunnel on which the

C-flow MUST be transmitted is the one identified in the PTA of the Intra-AS I-PMSI A-D route originated by PE1 for the given MVPN.

If the C-flow is a BIDIR-PIM C-flow with group address C-G-BIDIR, the rules applied by PE1 are:

- If the C-RPA for C-G-BIDIR is a C-address of PE1, or if PE1's route to the C-RPA is via a VRF interface, then if there is an I-PMSI A-D route currently originated by PE1, then the C-flow MUST be transmitted on the P-tunnel identified in the PTA of that I-PMSI A-D route.
- If PE1 determines the upstream PE for C-G-BIDIR's C-RPA to be some other PE, say PE2, then if there is an installed I-PMSI A-D route originated by PE2, the C-flow MUST be transmitted on the P-tunnel identified in the PTA of that route.

If there is no I-PMSI A-D route meeting the above conditions, the C-flow MUST NOT be transmitted.

#### 3.2.1.3. When an S-PMSI is a 'Match for Reception'

Suppose a given PE, say PE1, needs to receive multicast data packets of a particular C-flow. [MVPN-WILDCARDS] Section 3.2 specifies procedures for determining the S-PMSI A-D route, if any, that "matches" that C-flow for reception. Those rules apply unchanged for C-flows that are not BIDIR-PIM C-flows. The remainder of this section applies only to C-BIDIR flows.

The rules of [MVPN-WILDCARDS] Section 3.2.1 are not applicable to C-BIDIR flows. The rules of [MVPN-WILDCARDS] Section 3.2.2 are replaced by the following rules.

Suppose PE1 needs to receive (C-\*,C-G-BIDIR) traffic. Suppose also that PE1 has determined that PE2 is the "upstream PE" [MVPN] for the C-RPA of C-G-BIDIR. Then:

- If PE1 has an installed (C-\*,C-G-BIDIR) S-PMSI A-D route originated by PE2, then (C-\*,C-G-BIDIR) matches this route.
- Otherwise, if PE1 has an installed (C-\*,C-\*-BIDIR) route originated by PE2, then (C-\*,C-G-BIDIR) matches this route.
- Otherwise, if PE1 has an installed (C-\*,C-\*) S-PMSI A-D route originated by PE2, then (C-\*,C-G-BIDIR) matches this route.

If there is an S-PMSI A-D route matching (C-\*,C-G-BIDIR), according

to these rules, the root node of that P-tunnel is considered to be the "distinguished PE" for that (C-\*,C-G-BIDIR) flow. If a (C-\*,C-G-BIDIR) packet is received on a P-tunnel whose root node is not the distinguished PE for the C-flow, the packet MUST be discarded.

#### 3.2.1.4. When an I-PMSI is a 'Match for Reception

Suppose a given PE, say PE1, needs to receive packets of a given C-flow (of a given MVPN) from another PE, but according to the conditions of Section 3.2.1.3 and/or [MVPN-WILDCARDS] section 3.2, that C-flow does not match any S-PMSI A-D route. Then the packets of the C-flow need to be received on the MVPN's I-PMSI.

If the C-flow is not a BIDIR-PIM C-flow, the rules for determining the P-tunnel on which packets of the C-flow are expected are given in [MVPN]. The remainder of this section applies only to C-BIDIR flows.

Suppose that PE1 needs to receive (C-\*,C-G-BIDIR) traffic from other PEs. Suppose also that PE1 has determined that PE2 is the "upstream PE" [MVPN] for the C-RPA of C-G-BIDIR. Then PE1 considers PE2 to be the "distinguished PE" for (C-\*,C-G-BIDIR). If PE1 has an installed Intra-AS I-PMSI A-D route originated by PE2, PE1 will expect to receive packets of the C-flow from the tunnel specifies in that route's PTA. (If all VRFs of the MVPN have been properly provisioned to use the Flat Partitioned Method for the I-PMSI, the PTA will specify a bidirectional P-tunnel.)

If a (C-\*,C-G-BIDIR) packet is received on a P-tunnel other than the expected one, packet MUST be discarded.

#### 3.2.2. Hierarchical Partitioning

The procedures of this section and its sub-sections apply when (and only when) the Hierarchical Partitioned Method is used. This method is introduced in [MVPN] Section 11.2.2. This document only provides procedures for using this method when using MP2MP LSPs as the P-tunnels.

The Hierarchical Partitioned Method provides the same functionality as the Flat Partitioned Method, but requires a smaller amount of state to be maintained in the core of the network. However, it requires the use of upstream-assigned MPLS labels ("PE Distinguisher Labels"), which are not necessarily supported by all hardware platforms. The upstream-assigned labels are used to provide an LSP hierarchy, in which an "outer" MP2MP LSP carries multiple "inner"

MP2MP LSPs. Transit routers along the path between PE routers then only need to maintain state for the outer MP2MP LSP.

When this method is used to instantiate a particular PMSI, the bidirectional P-tunnel advertised in the PTA of the corresponding I-PMSI or S-PMSI A-D route is the "outer" P-tunnel. When a packet is received from a P-tunnel, the PE that receives it can infer the identity of the outer P-tunnel from the MPLS label that has risen to the top of the packet's label stack. However, the packet's "distinguished PE" is not necessarily the root node of the the outer P-tunnel. Rather, the identity of the packet's distinguished PE is inferred from the PE Distinguisher Label further down in the label stack. (See [MVPN] Section 12.3.) The PE Distinguisher Label may be thought of as identifying an "inner" MP2MP LSP whose root is the PE corresponding to that label.

In the context of a given MVPN, if it is desired to use the Hierarchical Partitioned Method to instantiate an I-PMSI, a (C-\*,C-\*) S-PMSI, or a (C-\*,C-\*-BIDIR) S-PMSI, the corresponding A-D routes MUST be originated by some of the PEs that attach to that MVPN. The PEs are REQUIRED to originate these routes are those that satisfy one of the following conditions:

- There is a C-BIDIR group for which the best path from the PE to the C-RPA of that C-group is via a VRF interface, or
- The PE might have to transmit unidirectional customer multicast traffic on the PMSI identified in the route (of course this condition does not apply to (C-\*,C-\*-BIDIR) or to (C-\*,C-G-BIDIR) S-PMSIs).
- The PE is the root node of the MP2MP LSP that is used to instantiate the PMSI.

When the Hierarchical Partitioned method is used to instantiate a (C-\*,C-G-BIDIR) S-PMSI, the corresponding (C-\*,C-G-BIDIR) S-PMSI route MUST NOT be originated by a given PE unless either (a) that PE's best path to the C-RPA for C-G-BIDIR is via a VRF interface, or (b) the C-RPA is a C-address of the PE. Further, that PE MUST be the root node of the MP2MP LSP identified in the PTA of the S-PMSI A-D route.

If any VRF of a given MVPN uses this method to instantiate an S-PMSI with a bidirectional P-tunnel, all VRFs of that MVPN must use this method.

Suppose that for a given MVPN, the Hierarchical Partitioned Method is used to instantiate the I-PMSI. In general, more than one of the PEs

in the MVPN will originate an Intra-AS I-PMSI A-D route for that MVPN. This document allows the PTAs of those routes to all specify the same MP2MP LSP as the "outer tunnel". However, it does not require that those PTAs all specify the same MP2MP LSP as the outer tunnel. By having all the PEs specify the same outer tunnel for the I-PMSI, one can minimize the amount of state in the transit nodes. By allowing them to specify different outer tunnels, one uses more state, but may increase the robustness of the system.

The considerations of the previous paragraph apply as well when the Hierarchical Partitioned Method is used to instantiate an S-PMSI.

#### 3.2.2.1. Advertisement of PE Distinguisher Labels

A PE Distinguisher Label is an upstream-assigned MPLS label [RFC5331] that can be used, in the context of a MP2MP LSP, to denote a particular PE that either has joined or may in the future join that LSP.

In order to use upstream-assigned MPLS labels in the context of an "outer" MP2MP LSP, there must be a convention that identifies a particular router as the router that is responsible for allocating the labels and for advertising the labels to the PEs that may join the MP2MP LSP. This document REQUIRES that the PE Distinguisher Labels used in the context of a given MP2MP LSP be allocated and advertised by the router that is the root node of the LSP.

This convention accords with the rules of section 7 of [RFC5331]. Note that according to section 7 of [RFC5331], upstream-assigned labels are unique in the context of the IP address of the root node; if two MP2MP LSPs have the same root node IP address, the upstream-assigned labels used within the two LSPs come from the same label space.

A PE Distinguisher Labels attribute SHOULD NOT be attached to an I-PMSI or S-PMSI A-D route unless that route also contains a PTA that specifies an MP2MP LSP. (While PE Distinguisher Labels could in theory also be used if the PTA specifies a BIDIR-PIM P-tunnel, such use is outside the scope of this document.)

The PE Distinguisher Labels attribute specifies a set of <MPLS label, IP address> bindings. Within a given PE Distinguisher Labels attribute, each such IP address MUST appear at most once, and each MPLS label MUST appear only once; otherwise the attribute is considered to be malformed.

When a PE Distinguisher Labels attribute is included in a given

I-PMSI or S-PMSI A-D route, it MUST assign a label to the IP address of each of the following PEs:

- The root node of the MP2MP LSP identified in the PTA of the route,
- Any PE that is possibly the ingress PE for a C-RPA of any C-BIDIR group.
- Any PE that may need to transmit non-C-BIDIR traffic on the MP2MP LSP identified in the PTA of the route.

One simple way to meet these requirements is to assign a PE Distinguisher label to every PE that has originated an Intra-AS I-PMSI A-D route.

#### 3.2.2.2. When an S-PMSI is a 'Match for Transmission'

Suppose a given PE, say PE1, needs to transmit multicast data packets of a particular C-flow. [MVPN-WILDCARDS] Section 3.1 gives a four-step algorithm for determining the S-PMSI A-D route, if any, that "matches" that C-flow for transmission.

If the C-flow is not a BIDIR-PIM C-flow, these rules apply unchanged. If there is a matching S-PMSI A-D route, the P-tunnel on which the C-flow MUST be transmitted is the one identified in the PTA of the matching route. Each packet of the C-flow MUST carry the PE Distinguisher Label assigned by the root node of that P-tunnel to the IP address of PE1. See section 12.3 of [MVPN] for encapsulation details.

The remainder of this section applies only to C-BIDIR flows. If a C-BIDIR flow has group address C-G-BIDIR, the rules applied by PE1 are the same as the rules given in section 3.2.1.1.

If there is a matching S-PMSI A-D route, PE1 MUST transmit the C-flow on the P-tunnel identified in its PTA. In constructing the packet's MPLS label stack, it must use the PE Distinguisher Label that was assigned by the P-tunnel's root node to the IP address of "PE2", not the label assigned to the IP address of "PE1". (Section 3.2.1.1 specifies the difference between PE1 and PE2.) See section 12.3 of [MVPN] for encapsulation details. Note that the root of the P-tunnel might be a PE other than PE1 or PE2.



### 3.2.2.3. When an I-PMSI is a 'Match for Transmission'

Suppose a given PE, say PE1, needs to transmit packets of a given C-flow (of a given MVPN) to other PEs, but according to the conditions of section 3.2.3.1 and/or [MVPN-WILDCARDS] section 3.1, that C-flow does not match any S-PMSI A-D route. Then the packets of the C-flow need to be transmitted on the MVPN's I-PMSI.

If the C-flow is not a BIDIR-PIM C-flow, the P-tunnel on which the C-flow MUST be transmitted is the one identified in the PTA of the Intra-AS I-PMSI A-D route originated by PE1 for the given MVPN. Each packet of the C-flow MUST carry the PE Distinguisher Label assigned by the root node of that P-tunnel to the IP address of PE1.

If the C-flow is a BIDIR-PIM C-flow with group address C-G-BIDIR, the rules as applied by PE1 are the same as those given in section 3.2.1.2.

Note that if a matching I-PMSI A-D route is found, the PTA of that route will have a non-zero MPLS label. This label must be pushed on each packet of the C-flow before that packet is transmitted through the P-tunnel identified in the PTA.

If, for a packet of a particular C-flow, there is no S-PMSI A-D route or I-PMSI A-D route that is a match for transmission, the packet MUST NOT be transmitted.

### 3.2.2.4. When an S-PMSI is a 'Match for Reception'

Suppose a given PE, say PE1, needs to receive multicast data packets of a particular C-flow. [MVPN-WILDCARDS] Section 3.2 specifies procedures for determining the S-PMSI A-D route, if any, that "matches" that C-flow for reception. Those rules require that the matching S-PMSI A-D route has been originated by the upstream PE for the C-flow. The rules are modified in this section, as follows.

Consider a particular C-flow. Suppose either:

- the C-flow is unidirectional, and PE1 determines that its upstream PE is PE2, or
- the C-flow is bidirectional, and PE1 determines that the upstream PE for its C-RPA is PE2.

Then the C-flow may match an installed S-PMSI A-D route that was not originated by PE2, as long as:

1. the PTA of that A-D route identifies an MP2MP LSP, and
2. there is an installed S-PMSI A-D route originated the root node of that LSP, or PE1 itself the root node of the LSP and there is a currently originated S-PMSI A-D route from PE1 whose PTA identifies that LSP, and
3. the latter S-PMSI A-D route (the one identified in 2 just above) contains a PE Distinguisher Labels attribute that assigned an MPLS label to the IP address of PE2.

However, a bidirectional C-flow never matches an S-PMSI A-D route whose NLRI contains (C-S,C-G).

If a multicast data packet is received over a matching P-tunnel, but does not carry the value of the PE Distinguisher Label that has been assigned to the upstream PE for its C-flow, then the packet MUST be discarded.

#### 3.2.2.5. When an I-PMSI is a 'Match for Reception'

If a PE needs to receive packets of a given C-flow (of a given MVPN) from another PE, and if, according to the conditions of section 3.2.3.3, that C-flow does not match any S-PMSI A-D route, then the packets of the C-flow need to be received on the MVPN's I-PMSI. The P-tunnel on which the packets are expected to arrive is determined by the Intra-AS I-PMSI A-D route originated by the "distinguished PE" for the given C-flow. The PTA of that route specifies the "outer P-tunnel", and thus determines the top label that packets of that C-flow will be carrying when received. A PE that needs to receive packets of a given C-flow must determine the expected value of the second label for packets of that C-flow. This will be the value of a PE Distinguisher Label, taken from the PE Distinguisher Labels attribute of the Intra-AS I-PMSI A-D route of the root node of that outer tunnel. The expected value of the second label on received packets (corresponding to the "inner tunnel") of a given C-flow is determined according to the following rules.

First, the "distinguished PE" for the C-flow is determined:

- If the C-flow is not a BIDIR-PIM C-flow, the "distinguished PE" for the C-flow is its "upstream PE", as determined by the rules of [MVPN].

- If the C-flow is a BIDIR-PIM C-flow, the "distinguished PE" for the C-flow is its "upstream PE" of the C-flow's C-RPA, as determined by the rules of [MVPN].

The expected value of the second label is the value that the root PE of the outer tunnel has assigned, in the PE Distinguisher Labels attribute of its Intra-AS I-PMSI A-D route, to the IP address of the "distinguished PE".

Packets addresses to C-G that arrive on other than the expected inner and outer P-tunnels (i.e., that arrive with unexpected values of the top two labels) MUST be discarded.

### 3.2.3. Unpartitioned

When a particular MVPN uses the Unpartitioned Method of instantiating an I-PMSI with a bidirectional P-tunnel, it MUST be the case that at least one VRF of that MVPN originates an Intra-AS I-PMSI A-D route that includes a PTA specifying a bidirectional P-tunnel. The conditions under which an Intra-AS I-PMSI A-D route must be originated from a given VRF are as specified in [MVPN-BGP]. This document allows all but one of such routes to omit the PTA. However, each such route MAY contain a PTA. If the PTA is present, it MUST specify a bidirectional P-tunnel. As specified in [MVPN] and [MVPN-BGP], every PE that imports such an Intra-AS I-PMSI A-D route into one of its VRFs MUST, if the route has a PTA, join the P-tunnel specified in the route's PTA.

Packets received on any of these P-tunnels are treated as having been received over the I-PMSI. The disposition of a received packet MUST NOT depend upon the particular P-tunnel over which it has been received.

When a PE needs to transmit a packet on such an I-PMSI, then if that PE advertised a P-tunnel in the PTA of an Intra-AS I-PMSI A-D route that it originated, the PE SHOULD transmit the on that P-tunnel. However, any PE that transmits a packet on the I-PMSI MAY transmit it on any of the P-tunnels advertised in any of the currently installed Intra-AS I-PMSI A-D routes for its VPN.

This allows a single bidirectional P-tunnel to be used to instantiate the I-PMSI, but also allows the use of multiple bidirectional P-tunnels. There may be a robustness advantage in having multiple P-tunnels available for use, but the number of P-tunnels used does not impact the functionality in any way. If there are, e.g., two P-tunnels available, these procedures allow each P-tunnel to be advertised by a single PE, but they also allow each P-tunnel to be

advertised by multiple PEs. Note that the PE advertising a given P-tunnel does not have to be the root node of the tunnel. The root node might not even be a PE router, and might not originate any BGP routes at all.

In the Unpartitioned Method, packets received on the I-PMSI cannot be associated with a distinguished PE, so duplicate detection using the techniques of [MVPN] section 9.1.1 is not possible; the techniques of [MVPN] 9.1.2 or 9.1.3 would have to be used instead. Support for C-BIDIR using the "Partitioned set of PEs" technique ([MVPN] section 11.2 and [RFC6517] section 3.6) is not possible when the Unpartitioned Method is used. If it is desired to use that technique to support C-BIDIR, but also to use the Unpartitioned Method to instantiate the I-PMSI, then all the C-BIDIR traffic would have to be carried on an S-PMSI, where the S-PMSI is instantiated using one of the Partitioned Methods.

When a PE, say PE1, needs to transmit multicast data packets of a particular C-flow to other PEs, and PE1 does not have an S-PMSI that is a "match for transmission for that C-flow (see section 3.2.3.1), PE1 transmits the packets on one of the P-tunnel(s) that instantiates the I-PMSI. When a PE, say PE1, needs to receive multicast data packets of a particular C-flow from another PE, and PE1 does not have an S-PMSI that is a "match for reception for that C-flow (see section 3.2.3.2), PE1 expects to receive the packets on any of the P-tunnel(s) that instantiates the I-PMSI.

When a particular MVPN uses the Unpartitioned Method to instantiate a (C-\*,C-\*) S-PMSI or a (C-\*,C-\*-BIDIR) S-PMSI using a bidirectional P-tunnel, the same conditions apply as when an I-PMSI is instantiated via the Unpartitioned Method. The only difference is that a PE need not join a P-tunnel that instantiates the S-PMSI unless that PE needs to receive multicast packets on the S-PMSI.

When a particular MVPN uses bidirectional P-tunnels to instantiate other S-PMSIs, different S-PMSI A-D routes that do not contain (C-\*,C-\*) or (C-\*,C-\*-BIDIR), originated by the same or by different PEs, MAY have PTAs that identify the same bidirectional tunnel, and they MAY have PTAs that do not identify the same bidirectional tunnel.

While the Unpartitioned Method MAY be used to instantiate an S-PMSI to which one or more C-BIDIR flows are bound, it must be noted that the "Partitioned Set of PEs" method discussed in [MVPN] section 11.2 and [RFC6517] section 3.6 cannot be supported using the Unpartitioned Method. C-BIDIR support would have to be provided by the procedures of [MVPN] section 11.1.

### 3.2.3.1. When an S-PMSI is a 'Match for Transmission'

Suppose a PE needs to transmit multicast data packets of a particular customer C-flow. [MVPN-WILDCARDS] Section 3.1 gives a four-step algorithm for determining the S-PMSI A-D route, if any, that "matches" that C-flow for transmission. When referring to that section, please recall that BIDIR-PIM groups are also "Any Source Multicast" (ASM) groups.

When bidirectional P-tunnels are used in the Unpartitioned Method, the same algorithm applies, with one modification, when the PTA of an S-PMSI A-D route identifies a bidirectional P-tunnel. One additional step is added to the algorithm. This new step occurs before the fourth step of the algorithm, and is as follows:

- Otherwise, if there is a (C-\*,C-\*-BIDIR) S-PMSI A-D route currently originated by PE1, and if C-G is a BIDIR group, the C-flow matches that route.

When the Unpartitioned Method is used, the PE SHOULD transmit the C-flow on the P-tunnel advertised in the in the matching S-PMSI A-D route, but it MAY transmit the C-flow on any P-tunnel that is advertised in the PTA of any installed S-PMSI A-D route that contains the same (C-S,C-G) as the matching S-PMSI A-D route.

### 3.2.3.2. When an S-PMSI is a 'Match for Reception'

Suppose a PE needs to receive multicast data packets of a particular customer C-flow. [MVPN-WILDCARDS] Section 3.2 specifies the procedures for determining the S-PMSI A-D route, if any, that advertised the P-tunnel on which the PE should expect to receive that C-flow.

When bidirectional P-tunnels are used in the Unpartitioned Method, the same procedures apply, with one modification.

The last paragraph of Section 3.2.2 of [MVPN-WILDCARDS] begins:

"If (C-\*,C-G) does not match a (C-\*,C-G) S-PMSI A-D route from PE2, but PE1 has an installed (C-\*,C-\*) S-PMSI A-D route from PE2, then (C-\*,C-G) matches the (C-\*,C-\*) route if one of the following conditions holds:"

This is changed to:

"If (C-\*,C-G) does not match a (C-\*,C-G) S-PMSI A-D route from PE2, but C-G is a BIDIR group and PE1 has an installed (C-\*,C-\*-BIDIR) S-PMSI A-D route, then (C-\*,C-G) matches that route. Otherwise, if PE1 has an installed (C-\*,C-\*) S-PMSI A-D route from PE2, then (C-\*,C-G) matches the (C-\*,C-\*) route if one of the following conditions holds:"

When the Unpartitioned Method is used, the PE MUST join the P-tunnel that is advertised in the matching S-PMSI A-D route, and MUST also join the P-tunnels that are advertised in other installed S-PMSI A-D routes that contain the same (C-S,C-G) as the matching S-PMSI A-D route.

#### 3.2.4. Minimal Feature Set for Compliance

A PE that does not provide C-BIDIR support using the "partitioned set of PEs" method may be deemed compliant to this specification if it supports the Unpartitioned Method, using either MP2MP LSPs or BIDIR-PIM multicast distribute trees as P-tunnels.

A PE that does provide C-BIDIR support using the "partitioned set of PEs" method, MUST, at a minimum, be able to provide C-BIDIR support using the "Partial Mesh of MP2MP P-tunnels" variant of this method (see section 11.2 of [MVPN]). An implementation will be deemed complaint to this minimum requirement if it can carry all of a VPN's C-BIDIR traffic on a (C-\*,C-\*-BIDIR) S-PMSI that is instantiated by a bidirectional P-tunnel, using the flat partitioned method.

#### 4. IANA Considerations

This document has no actions for IANA.

#### 5. Security Considerations

There are no additional security considerations beyond those of [MVPN] and [MVPN-BGP], or any that may apply to the particular protocol used to set up the bidirectional tunnels ([BIDIR-PIM], [mLDP]).

## 6. Acknowledgments

The authors wish to thank Karthik Subramanian, Rajesh Sharma, and Apoorva Karan for their input. We also thank Yakov Rekhter for his valuable critique.

Special thanks go to Jeffrey Zhang for his careful review, probing questions, and useful suggestions.

## 7. Authors' Addresses

Arjen Boers  
E-mail: arjen@boers.com

Yiqun Cai  
Microsoft  
1065 La Avenida  
Mountain View, CA 94043  
E-mail: yiqunc@microsoft.com

Eric C. Rosen  
Cisco Systems, Inc.  
1414 Massachusetts Avenue  
Boxborough, MA, 01719  
E-mail: erosen@cisco.com

IJsbrand Wijnands  
Cisco Systems, Inc.  
De kleetlaan 6a Diegem 1831  
Belgium  
E-mail: ice@cisco.com

## 8. Normative References

[BIDIR-PIM] "Bidirectional Protocol Independent Multicast", Handley, Kouvelas, Speakman, Vicisano, RFC 5015, October 2007

[L3VPN], "BGP/MPLS IP Virtual Private Networks", Rosen, Rekhter (editors), RFC 4364, February 2006

[mLDP] "Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths", Wijnands, Minei, Kompella, Thomas, RFC 6388, November 2011

[MVPN] "Multicast in MPLS/BGP IP VPNs", Rosen, Aggarwal, et. al., RFC 6513, February 2012

[MVPN-BGP] "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", Aggarwal, Rosen, Morin, Rekhter, RFC 6514, February 2012

[MVPN-WILDCARDS] "Wild Cards in Multicast VPN Auto-Discovery Routes", Rosen, Rekhter, Hendrickx, Qiu, RFC 6625, May 2012

[PIM] "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", Fenner, Handley, Holbrook, Kouvelas, RFC 4601, August 2006

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels.", Bradner, March 1997

## 9. Informative References

[RFC5331] "MPLS Upstream Label Assignment and Context-Specific Label Space", Aggarwal, Rekhter, Rosen, RFC 5331, August 2008

[RFC6517] "Mandatory Features in a Layer 3 Multicast BGP/MPLS VPN Solution", Morin, Niven-Jenkins, Kamite, Zhang, Leymann, Bitar, RFC 6517, February 2012

[MVPN-BIDIR-IR] "Simulating 'Partial Mesh of MP2MP P-Tunnels' with Ingress Replication", Zhang, Rekhter, Dolganow, draft-ietf-l3vpn-mvpn-bidir-ingress-replication-00.txt, February 2014

[MVPN-XNET] "Extranet Multicast in BGP/IP MPLS VPNs", Rekhter, Rosen (editors), draft-ietf-l3vpn-mvpn-extranet-04.txt, March 2014



INTERNET-DRAFT  
Intended Status: Proposed Standard  
Expires: 2014-04-12

Saud Asif  
AT&T  
Andy Green  
BT  
Sameer Gulrajani  
Cisco  
Pradeep Jain  
Alcatel-Lucent  
Jeffrey Zhang  
Juniper  
2013-10-12

MPLS/BGP Layer 3 VPN Multicast  
Management Information Base

draft-ietf-l3vpn-mvpn-mib-04

Abstract

This memo defines an portion of the Management Information Base (MIB) for use with network management protocols in the Internet community.

In particular, it describes managed objects to configure and/or monitor multicast in MPLS/BGP-based Layer-3 VPN (MVPN) on an MVPN router.

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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

The list of current Internet-Drafts can be accessed at  
<http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at

<http://www.ietf.org/shadow.html>

#### Copyright and License Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

#### Table of Contents

0	Draft history	3
1	Introduction	3
1.1	Terminology	3
2	MVPN MIB	4
2.1	Summary of MIB Module	4
2.2	MIB Module Definitions	5
3	Security Considerations	30
4	IANA Considerations	30
5	Acknowledgement	30
6	References	30
6.1	Normative References	30
6.2	Informative References	31
	Authors' Addresses	31

## 0 Draft history

This draft is a first pass at a MIB document for [MVPN]. As such, it should be considered as a early work.

Some aspects of BGP-MVPN (see definition below in "Introduction"), such as exranet, may be specified in future revisions.

[note to author/reviewers: conformance groups to be added ]

[this section should be removed as soon as its stops being relevant]

## 1 Introduction

Multicast in MPLS/BGP L3 VPNs is specified in {[MVPN], [BGP-MVPN]}. These specifications support either PIM or BGP as the protocol for exchanging VPN multicast (referred to as C-multicast states, where 'C-' stands for 'VPN Customer-') among PEs. In the rest of this document we'll use the term "PIM-MVPN" to refer to {[MVPN], [BGP-MVPN]} with PIM being used for exchanging C-multicast states, and "BGP-MVPN" to refer to {[MVPN], [BGP-MVPN]} with BGP is used for exchanging C-multicast states.

This document defines a standard MIB for MVPN-specific objects that are generic to both PIM-MVPN and BGP-MVPN.

This document borrowed some text from Cisco PIM-MVPN MIB [CISCO-MIB]. For PIM-MVPN this document attempts to provide coverage comparable to [CISCO-MIB], but in a generic way that applies to both PIM-MVPN and BGP-MVPN.

Comments should be made directly to the Layer-3 VPN (L3VPN) WG at [l3vpn@ietf.org](mailto:l3vpn@ietf.org).

### 1.1 Terminology

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].

This document adopts the definitions, acronyms and mechanisms described in [MVPN] and other documents that [MVPN] refers to. Familiarity with Multicast, MPLS, L3VPN, MVPN concepts and/or mechanisms is assumed.

Interchangeably, the term MVRF and MVPN are used to refer to a partiular Multicast VPN instantiation on a particular PE device.

## 2 MVPN MIB

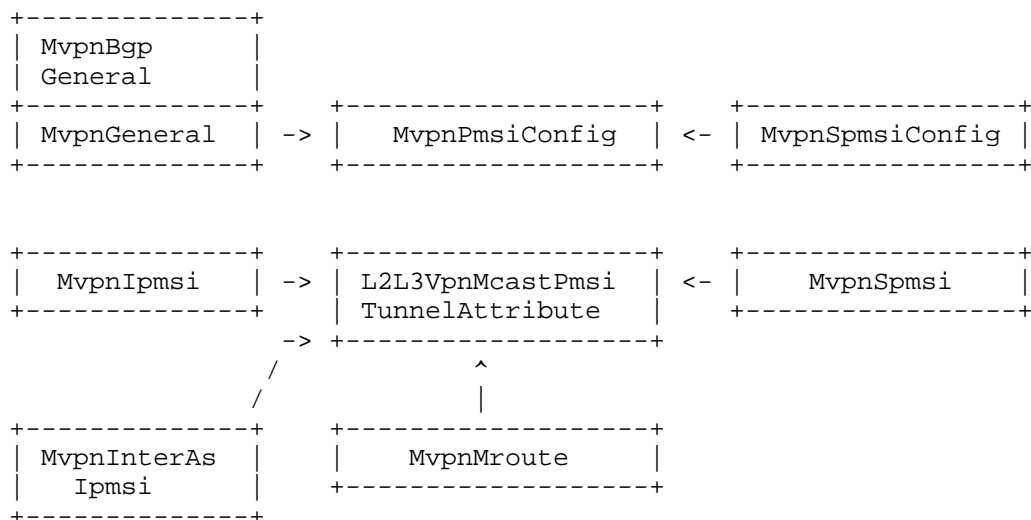
This MIB enables configuring and/or monitoring of MVPNs on PE devices: the whole multicast VPN machinery and the per-MVRFs information, including the configuration, status and operational details, such as different PMSIs and the provider tunnels implementing them.

### 2.1 Summary of MIB Module

The configuration and states specific to an MVPN include the following:

- C-multicast routing exchange protocol (PIM or BGP)
- I-PMSI, S-PMSI and corresponding provider tunnels
- Mapping of c-multicast states to PMSI/tunnels

To represent them, the following tables are defined.



#### - mvpnGeneralTable/Entry

An entry in this table is created for every MVRF in the device, for general configuration/states of the MVRF, including I-PMSI configuration.

Existence of the corresponding VRF in [L3VPN-MIB] is necessary for

a row to exist in this table.

- mvpnBgpGeneralTable/Entry

This table augments mvpnGeneralTable and is for BGP-MVPN specific information.

- mvpnSpmsiConfigTable/Entry

This table contains objects for S-PMSI configurations in an MVRF.

- mvpnPmsiConfigTable/Entry

Both I-PMSI configuration (in mvpnGeneralEntry) and S-PMSI configuration (in mvpnSpmsiConfigEntry) refer to entries in this table.

- mvpnIpmsiTable/Entry

This table contains all advertised or received intra-as I-PMSIs. With PIM-MVPN, it is applicable only when BGP-Based Autodiscovery of MVPN Membership is used.

- mvpnInterAsIpmsiTable/Entry

This table contains all advertised or received inter-as I-PMSIs. With PIM-MVPN, it is applicable only when BGP-Based Autodiscovery of MVPN Membership is used.

- mvpnSpmsiTable/Entry

This table contains all advertised or received S-PMSIs.

- l2l3VpnMcastPmsiTunnelAttributeTable/Entry

This table is defined separately in l2l3VpnMcastMIB [L2L3MVPN-MIB], which is common for both VPLS Multicast and MVPN. It contains sent/received PMSI attribute entries referred to by mvpnIpmsiEntry, mvpnSpmsiEntry, mvpnInterAsIpmsiEntry, and other MIB objects (e.g., VPLS Multicast ones).

- mvpnMrouteTable/Entry

This table augments ipMcastMIB.ipMcast.ipMcastRouteTable, for some MVPN specific information.

## 2.2 MIB Module Definitions

```
MCAST-VPN-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
    experimental, Unsigned32
        FROM SNMPv2-SMI

    MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
        FROM SNMPv2-CONF

    TruthValue, RowPointer, RowStatus, TimeStamp, TimeInterval
        FROM SNMPv2-TC

    SnmpAdminString
        FROM SNMP-FRAMEWORK-MIB

    InetAddress, InetAddressType
        FROM INET-ADDRESS-MIB

    MplsLabel
        FROM MPLS-TC-STD-MIB

    mplsL3VpnVrfName, MplsL3VpnRouteDistinguisher
        FROM MPLS-L3VPN-STD-MIB

    ipMcastRouteEntry
        FROM IPMCAST-MIB

    L2L3VpnMcastProviderTunnelType
        FROM L2L3-VPN-MCAST-MIB;

mvpnMIB MODULE-IDENTITY
    LAST-UPDATED "201301071200Z" -- 07 January 2013 12:00:00 GMT
    ORGANIZATION "IETF Layer-3 Virtual Private
        Networks Working Group."
    CONTACT-INFO
        " Jeffrey (Zhaohui) Zhang
          zzhang@juniper.net

          Comments and discussion to l3vpn@ietf.org"

    DESCRIPTION
        "This MIB contains managed object definitions for
        multicast in BGP/MPLS IP VPNs defined by [MVPN].
        Copyright (C) The Internet Society (2013)."
```

-- Revision history.

```
REVISION "201301071200Z" -- 07 January 2013 12:00:00 GMT
```

```
DESCRIPTION
    "Initial version of the draft."
    ::= { experimental 99 } -- number to be assigned

-- Top level components of this MIB.
mvpnNotifications OBJECT IDENTIFIER ::= { mvpnMIB 0 }

-- tables, scalars
mvpnObjects          OBJECT IDENTIFIER ::= { mvpnMIB 1 }

-- conformance information
mvpnConformance     OBJECT IDENTIFIER ::= { mvpnMIB 2 }

-- mvpn Objects

mvpnScalars          OBJECT IDENTIFIER ::= { mvpnObjects 1 }
mvpnGeneral           OBJECT IDENTIFIER ::= { mvpnObjects 2 }
mvpnConfig            OBJECT IDENTIFIER ::= { mvpnObjects 3 }
mvpnStates            OBJECT IDENTIFIER ::= { mvpnObjects 4 }

-- Scalar Objects

mvpnMvrfNumber OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The number of MVRFs for IPv4 or IPv6 or mLDP C-Multicast
         that are present in this device."
    ::= { mvpnScalars 1 }

mvpnMvrfNumberV4 OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The number of MVRFs for IPv4 C-Multicast that are present
         in this device."
    ::= { mvpnScalars 2 }

mvpnMvrfNumberV6 OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The number of MVRFs for IPv6 C-Multicast that are present
         in this device."
    ::= { mvpnScalars 3 }
```

```
mvpnMvrfNumberPimV4 OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The number of PIM-MVPN MVRFs for IPv4 C-Multicast that are present
         in this device."
    ::= { mvpnScalars 4 }

mvpnMvrfNumberPimV6 OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The number of PIM-MVPN MVRFs for IPv6 C-Multicast that are present
         in this device."
    ::= { mvpnScalars 5 }

mvpnMvrfNumberBgpV4 OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The number of BGP-MVPN MVRFs for IPv4 C-Multicast that are present
         in this device."
    ::= { mvpnScalars 6 }

mvpnMvrfNumberBgpV6 OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The number of BGP-MVPN MVRFs for IPv6 C-Multicast that are present
         in this device."
    ::= { mvpnScalars 7 }

mvpnMvrfNumberMldp OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The number of BGP-MVPN MVRFs for mLDP C-Multicast that are present
         in this device."
    ::= { mvpnScalars 8 }

mvpnNotificationEnable OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS   read-write
```



```

STATUS          current
DESCRIPTION
    "If this object is TRUE, then the generation of all
    notifications defined in this MIB is enabled."
DEFVAL { false }
::= { mvpnScalars 9 }

-- General MVRF Information Table

mvpnGeneralTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF MvpnGeneralEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This table specifies the general information about the MVRFs
        present in this device."
    ::= { mvpnGeneral 1 }

mvpnGeneralEntry OBJECT-TYPE
    SYNTAX          MvpnGeneralEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "An entry in this table is created for every MVRF in the
        device."
    INDEX           { mplsL3VpnVrfName }
    ::= { mvpnGeneralTable 1 }

MvpnGeneralEntry ::= SEQUENCE {
    mvpnGenOperStatusChange      INTEGER,
    mvpnGenOperChangeTime       TimeStamp,
    mvpnGenCmcastRouteProtocolV4 INTEGER,
    mvpnGenCmcastRouteProtocolV6 INTEGER,
    mvpnGenIpmsiConfigV4        RowPointer,
    mvpnGenIpmsiConfigV6        RowPointer,
    mvpnGenInterAsPmsiConfigV4  RowPointer,
    mvpnGenInterAsPmsiConfigV6  RowPointer,
    mvpnGenRowStatus             RowStatus
}

mvpnGenOperStatusChange OBJECT-TYPE
    SYNTAX          INTEGER { createdMvrf(1),
                             deletedMvrf(2),
                             modifiedMvrfIpmsiConfig(3),
                             modifiedMvrfSpmsiConfig(4)
                           }
    MAX-ACCESS      read-only
    STATUS          current

```

## DESCRIPTION

"This object describes the last operational change that happened for the given MVRF.

```
createdMvrf - indicates that the MVRF was created in the
device.
```

deletedMvrf - indicates that the MVRF was deleted from the device. A row in this table will never have mvpnGenOperStatusChange equal to deletedMvrf(2), because in that case the row itself will be deleted from the table. This value for mvpnGenOperStatusChange is defined mainly for use in mvpnMvrfChange notification.

modifiedMvrfIpmsiConfig - indicates that the I-PMSI for the MVRF was configured, deleted or changed.

modifiedMvrfSpmsiConfig - indicates that the S-PMSI for the MVRF was configured, deleted or changed."

```
DEFVAL { createdMvrf }
::= { mvpnGeneralEntry 1 }
```

## mvpnGenOperChangeTime OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The time at which the last operational change for the MVRF in question took place. The last operational change is specified by `mvprnGenOperStatusChange`."

$$::= \{ \text{mvpnGeneralEntry } 2 \}$$

## mvpnGenCmcastRouteProtocolV4 OBJECT-TYPE

```
SYNTAX      INTEGER { pim (1),
                    bgp (2)
                    }
```

MAX-ACCESS read-write

STATUS current

## DESCRIPTION

"Protocol used to signal IPv4 C-multicast states across the provider core.

```
pim(1): PIM (PIM-MVPN).
```

```
bgp(2): BGP (BGP-MVPN)."
```

$$::= \{ \text{mvpnGeneralEntry } 3 \}$$

## mvpnGenCmcastRouteProtocolV6 OBJECT-TYPE

```
SYNTAX      INTEGER { pim (1),
                      bgp  (2)
```

```

    }
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        "Protocol used to signal IPv6 C-multicast states across the
        provider core.
        pim(1): PIM (PIM-MVPN).
        bgp(2): BGP (BGP-MVPN).
    ::= { mvpnGeneralEntry 4 }

mvpnGenIpmsiConfigV4 OBJECT-TYPE
    SYNTAX          RowPointer
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        "This points to a row in mvpnPmsiConfigTable,
        for I-PMSI configuration for IPv4."
    ::= { mvpnGeneralEntry 5 }

mvpnGenIpmsiConfigV6 OBJECT-TYPE
    SYNTAX          RowPointer
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        "This points to a row in mvpnPmsiConfigTable,
        for I-PMSI configuration for IPv6."
    ::= { mvpnGeneralEntry 6 }

mvpnGenInterAsPmsiConfigV4 OBJECT-TYPE
    SYNTAX          RowPointer
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        "This points to a row in mvpnPmsiConfigTable,
        for inter-as I-PMSI configuration for IPv4, in case of segmented
        inter-as provider tunnels."
    ::= { mvpnGeneralEntry 7 }

mvpnGenInterAsPmsiConfigV6 OBJECT-TYPE
    SYNTAX          RowPointer
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        "This points to a row in mvpnPmsiConfigTable,
        for inter-as I-PMSI configuration for IPv6, in case of segmented
        inter-as provider tunnels."
    ::= { mvpnGeneralEntry 8 }
```

```

mvpnGenRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "This is used to create or delete a row in this table."
    ::= { mvpnGeneralEntry 9 }

-- General BGP-MVPN table

mvpnBgpGeneralTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF MvpnBgpGeneralEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "This table augments the mvpnGeneralTable and is for BGP-MVPN
        specific information."
    ::= { mvpnGeneral 2 }

mvpnBgpGeneralEntry OBJECT-TYPE
    SYNTAX      MvpnBgpGeneralEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "The mvpnBgpGeneralEntry matches and augments an mvpnGeneralEntry
        for a BGP-MVPN instance, with BGP-MVPN specific informatoin."
    AUGMENTS    { mvpnGeneralEntry }
    ::= { mvpnBgpGeneralTable 1 }

MvpnBgpGeneralEntry ::= SEQUENCE {
    mvpnBgpGenMode          INTEGER,
    mvpnBgpGenUmhSelection  INTEGER,
    mvpnBgpGenSiteType      INTEGER,
    mvpnBgpGenCmcastImportRt MplsL3VpnRouteDistinguisher,
    mvpnBgpGenSrcAs         Unsigned32,
    mvpnBgpGenSptnlLimit    Unsigned32
}

mvpnBgpGenMode OBJECT-TYPE
    SYNTAX      INTEGER {
                    rpt-spt (1),
                    spt-only (2)
                }
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
        "For two different BGP-MVPN modes:
        rpt-spt(1): intersite-site shared tree mode

```

```

        spt-only(2): inter-site source-only tree mode."
 ::= { mvpnBgpGeneralEntry 1}

mvpnBgpGenUmhSelection OBJECT-TYPE
    SYNTAX          INTEGER {
                        highest-pe-address      (1),
                        c-root-group-hashing    (2),
                        ucast-umh-route         (3)
                      }
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        "The UMH selection method for this mvpn, as specified in section
        5.1.3 of [MVPN]:
        highest-pe-address (1): PE with the highest address
        c-root-group-hashing (2): hashing based on (c-root, c-group)
        ucast-umh-route (3): per ucast route towards c-root"

 ::= { mvpnBgpGeneralEntry 2}

mvpnBgpGenSiteType OBJECT-TYPE
    SYNTAX          INTEGER {
                        sender-receiver (1),
                        receiver-only   (2),
                        sender-only     (3)
                      }
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        "Whether this site is a receiver-only site or not.
        sender-receiver (1): both sender and receiver site.
        receiver-only   (2): receiver-only site.
        sender-only     (3): sender-only site."
 ::= { mvpnBgpGeneralEntry 3}

mvpnBgpGenCmcastImportRt OBJECT-TYPE
    SYNTAX          MplsL3VpnRouteDistinguisher
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        "The C-multicast Import RT that this device adds to
        unicast vpn routes that it advertises for this mvpn."
 ::= { mvpnBgpGeneralEntry 4}

mvpnBgpGenSrcAs OBJECT-TYPE
    SYNTAX          Unsigned32
    MAX-ACCESS      read-only
    STATUS          current

```

## DESCRIPTION

"The Source AS number in Source AS Extended Community that this device adds to the unicast vpn routes that it advertises for this mvpn."

```
::= { mvpnBgpGeneralEntry 5}
```

```
mvpnBgpGenSptnlLimit OBJECT-TYPE
```

```
SYNTAX          Unsigned32
```

```
MAX-ACCESS      read-write
```

```
STATUS          current
```

## DESCRIPTION

"The max number of selective provider tunnels this device allows for this mvpn."

```
::= { mvpnBgpGeneralEntry 6}
```

```
-- PMSI Configuration Table
```

```
mvpnPmsiConfigTable OBJECT-TYPE
```

```
SYNTAX          SEQUENCE OF MvpnPmsiConfigEntry
```

```
MAX-ACCESS      not-accessible
```

```
STATUS          current
```

## DESCRIPTION

"This table specifies the configured PMSIs."

```
::= { mvpnConfig 1 }
```

```
mvpnPmsiConfigEntry OBJECT-TYPE
```

```
SYNTAX          MvpnPmsiConfigEntry
```

```
MAX-ACCESS      not-accessible
```

```
STATUS          current
```

## DESCRIPTION

"An entry in this table is created for each PMSI configured on this router. It can be referred to by either I-PMSI configuration (in mvpnGeneralEntry) or S-PMSI configuration (in mvpnSpmsiConfigEntry)"

```
INDEX          { mvpnPmsiConfigTunnelType,
                  mvpnPmsiConfigTunnelAuxInfo,
                  mvpnPmsiConfigTunnelPimGroupAddressType,
                  mvpnPmsiConfigTunnelPimGroupAddress,
                  mvpnPmsiConfigTunnelOrTemplateName }
```

```
::= { mvpnPmsiConfigTable 1 }
```

```
MvpnPmsiConfigEntry ::= SEQUENCE {
```

mvpnPmsiConfigTunnelType	L2L3VpnMcastProviderTunnelType,
mvpnPmsiConfigTunnelAuxInfo	Unsigned32,
mvpnPmsiConfigTunnelPimGroupAddressType	InetAddressType,
mvpnPmsiConfigTunnelPimGroupAddress	InetAddress,
mvpnPmsiConfigTunnelOrTemplateName	SnmpAdminString,
mvpnPmsiConfigEncapsType	INTEGER,
mvpnPmsiConfigRowStatus	RowStatus

```
}

mvpnPmsiConfigTunnelType OBJECT-TYPE
    SYNTAX      L2L3VpnMcastProviderTunnelType
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "Type of tunnel used to instantiate the PMSI."
    ::= { mvpnPmsiConfigEntry 1 }

mvpnPmsiConfigTunnelAuxInfo OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "Additional tunnel information depending on the type.
         pim:          In case of S-PMSI, number of groups starting at
                        mvpnPmsiConfigTunnelPimGroupAddress.
                        This allows a range of PIM provider tunnel
                        group addresses to be specified in S-PMSI case.
                        In I-PMSI case, it must be 1.
         rsvp-p2mp:    1 for statically specified rsvp-p2mp tunnel
                        2 for dynamically created rsvp-p2mp tunnel
         ingress-replication:
                        1 for using any existing p2p/mp2p lsp
                        2 for dynamically creating new p2p lsp"
    ::= { mvpnPmsiConfigEntry 2 }

mvpnPmsiConfigTunnelPimGroupAddressType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "In case of PIM provider tunnel, the type of tunnel address."
    ::= { mvpnPmsiConfigEntry 3 }

mvpnPmsiConfigTunnelPimGroupAddress OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "In case of PIM provider tunnel, the provider tunnel address."
    ::= { mvpnPmsiConfigEntry 4 }

mvpnPmsiConfigTunnelOrTemplateName OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS   not-accessible
    STATUS      current
```

## DESCRIPTION

"The tunnel name or template name used to create tunnels.  
Depending on mvpnPmsiConfigTunnelType and  
mvpnPmsiConfigTunnelAuxInfo:

dynamically created rsvp-p2mp tunnel:	template name
statically specified rsvp-p2mp tunnel:	tunnel name
ingress-replication using	
dynamically created lsps:	template name
other:	null

```
::= { mvpnPmsiConfigEntry 5 }
```

mvpnPmsiConfigEncapsType OBJECT-TYPE

```
SYNTAX      INTEGER { greIp (1),
                      ipIp  (2),
                      mpls  (3)
                    }
```

MAX-ACCESS read-write

STATUS current

## DESCRIPTION

"The encapsulation type to be used, in case of PIM tunnel or  
ingress-replication."

```
::= { mvpnPmsiConfigEntry 6 }
```

mvpnPmsiConfigRowStatus OBJECT-TYPE

```
SYNTAX      RowStatus
```

MAX-ACCESS read-create

STATUS current

## DESCRIPTION

"Used to create/modify/delete a row in this table."

```
::= { mvpnPmsiConfigEntry 7 }
```

-- S-PMSI configuration table

mvpnSpmsiConfigTable OBJECT-TYPE

```
SYNTAX      SEQUENCE OF MvpnSpmsiConfigEntry
```

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

"This table specifies S-PMSI configuration."

```
::= { mvpnConfig 2 }
```

mvpnSpmsiConfigEntry OBJECT-TYPE

```
SYNTAX      MvpnSpmsiConfigEntry
```

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

"An entry is created for each S-PMSI configuration."



```

INDEX      {  mplsL3VpnVrfName,
               mvpnSpmsiConfigCmcastAddressType,
               mvpnSpmsiConfigCmcastGroupAddress,
               mvpnSpmsiConfigCmcastGroupPrefixLen,
               mvpnSpmsiConfigCmcastSourceAddress,
               mvpnSpmsiConfigCmcastSourcePrefixLen }
 ::= { mvpnSpmsiConfigTable 1 }

MvpnSpmsiConfigEntry ::= SEQUENCE {
    mvpnSpmsiConfigCmcastAddressType      InetAddressType,
    mvpnSpmsiConfigCmcastGroupAddress      InetAddress,
    mvpnSpmsiConfigCmcastGroupPrefixLen    Unsigned32,
    mvpnSpmsiConfigCmcastSourceAddress      InetAddress,
    mvpnSpmsiConfigCmcastSourcePrefixLen    Unsigned32,
    mvpnSpmsiConfigThreshold                Unsigned32,
    mvpnSpmsiConfigPmsiPointer               RowPointer,
    mvpnSpmsiConfigRowStatus                 RowStatus
}

mvpnSpmsiConfigCmcastAddressType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "Type of C-multicast address"
    ::= { mvpnSpmsiConfigEntry 1 }

mvpnSpmsiConfigCmcastGroupAddress OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "C-multicast group address"
    ::= { mvpnSpmsiConfigEntry 2 }

mvpnSpmsiConfigCmcastGroupPrefixLen OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "C-multicast group address prefix length.
         A group 0 (or ::0) with prefix length 32 (or 128)
         indicates wildcard group, while a group 0 (or ::0)
         with prefix length 0 indicates any group."
    ::= { mvpnSpmsiConfigEntry 3 }

mvpnSpmsiConfigCmcastSourceAddress OBJECT-TYPE
    SYNTAX      InetAddress

```

```
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION
    "C-multicast source address"
 ::= { mvpnSpmsiConfigEntry 4 }

mvpnSpmsiConfigCmcastSourcePrefixLen OBJECT-TYPE
SYNTAX          Unsigned32
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION
    "C-multicast source address prefix length.
     A source 0 (or ::0) with prefix length 32 (or 128)
     indicates a wildcard source, while a source 0 (or ::0)
     with prefix length 0 indicates any source."
 ::= { mvpnSpmsiConfigEntry 5 }

mvpnSpmsiConfigThreshold OBJECT-TYPE
SYNTAX          Unsigned32 (0..4294967295)
UNITS           "kilobits per second"
MAX-ACCESS      read-write
STATUS          current
DESCRIPTION
    "The bandwidth threshold value which when exceeded for a
     multicast routing entry in the given MVRFB, triggers usage
     of S-PMSI."
 ::= { mvpnSpmsiConfigEntry 6 }

mvpnSpmsiConfigPmsiPointer OBJECT-TYPE
SYNTAX          RowPointer
MAX-ACCESS      read-write
STATUS          current
DESCRIPTION
    "This points to a row in mvpnPmsiConfigTable,
     to specify tunnel attributes."
 ::= { mvpnSpmsiConfigEntry 7 }

mvpnSpmsiConfigRowStatus OBJECT-TYPE
SYNTAX          RowStatus
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION
    "Used to create/modify/delete a row in this table."
 ::= { mvpnSpmsiConfigEntry 8 }

-- Table of intra-as I-PMSIs advertised/received

mvpnIpmsiTable OBJECT-TYPE
```

```
SYNTAX          SEQUENCE OF MvpnIpmsiEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION
    "This table is for all advertised/received I-PMSI
    advertisements."
 ::= { mvpnStates 1 }

mvpnIpmsiEntry OBJECT-TYPE
SYNTAX          MvpnIpmsiEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION
    "An entry in this table corresponds to an I-PMSI
    advertisement that is advertised/received on this router.
    This represents all the sender PEs in the MVPN,
    with the provider tunnel they use to send traffic."
INDEX { mplsL3VpnVrfName,
        mvpnIpmsiAfi,
        mvpnIpmsiRD,
        mvpnIpmsiOrigAddrType,
        mvpnIpmsiOrigAddress }
 ::= { mvpnIpmsiTable 1 }

MvpnIpmsiEntry ::= SEQUENCE {
    mvpnIpmsiAfi      Unsigned32,
    mvpnIpmsiRD       MplsL3VpnRouteDistinguisher,
    mvpnIpmsiOrigAddrType InetAddressType,
    mvpnIpmsiOrigAddress InetAddress,
    mvpnIpmsiUpTime   TimeInterval,
    mvpnIpmsiAttribute RowPointer
}

mvpnIpmsiAfi OBJECT-TYPE
SYNTAX          Unsigned32 (1|2)
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION
    "The address family this I-PMSI is for.
    1 - IPv4
    2 - IPv6"
 ::= { mvpnIpmsiEntry 1 }

mvpnIpmsiRD OBJECT-TYPE
SYNTAX          MplsL3VpnRouteDistinguisher
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION
```

```
        "The Route Distinguisher in this I-PMSI."
 ::= { mvpnIpmsiEntry 2 }

mvpnIpmsiOrigAddrType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "The Internet address type of mvpnIpmsiOrigAddress."
 ::= { mvpnIpmsiEntry 3 }

mvpnIpmsiOrigAddress OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "The BGP address of the device that originated the I-PMSI."
 ::= { mvpnIpmsiEntry 4 }

mvpnIpmsiUpTime OBJECT-TYPE
    SYNTAX      TimeInterval
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The time since this I-PMSI
         was first advertised/received by the device."
 ::= { mvpnIpmsiEntry 5 }

mvpnIpmsiAttribute OBJECT-TYPE
    SYNTAX      RowPointer
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "Points to a row in the l2L3VpnMcastPmsiTunnelAttributeTable."
 ::= { mvpnIpmsiEntry 6 }

-- Table of inter-as I-PMSIs advertised/received

mvpnInterAsIpmsiTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF MvpnInterAsIpmsiEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "This table is for all advertised/received inter-as I-PMSI
         advertisements."
 ::= { mvpnStates 2 }

mvpnInterAsIpmsiEntry OBJECT-TYPE
```

```
SYNTAX          MvpnInterAsIpmsiEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION
    "An entry in this table corresponds to an inter-as I-PMSI
    advertisement that is advertised/received on this router.
    This represents all the ASes in the MVPN,
    with the provider tunnel used to send traffic to."
INDEX { mplsL3VpnVrfName,
        mvpnInterAsIpmsiAfi,
        mvpnInterAsIpmsiRD,
        mvpnInterAsIpmsiSrcAs }
 ::= { mvpnInterAsIpmsiTable 1 }

MvpnInterAsIpmsiEntry ::= SEQUENCE {
    mvpnInterAsIpmsiAfi      Unsigned32,
    mvpnInterAsIpmsiRD      MplsL3VpnRouteDistinguisher,
    mvpnInterAsIpmsiSrcAs    Unsigned32,
    mvpnInterAsIpmsiAttribute RowPointer
}

mvpnInterAsIpmsiAfi OBJECT-TYPE
    SYNTAX      Unsigned32 (1|2)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The address family this I-PMSI is for.
        1 - IPv4
        2 - IPv6"
    ::= { mvpnInterAsIpmsiEntry 1 }

mvpnInterAsIpmsiRD OBJECT-TYPE
    SYNTAX      MplsL3VpnRouteDistinguisher
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The Route Distinguisher in this inter-as I-PMSI."
    ::= { mvpnInterAsIpmsiEntry 2 }

mvpnInterAsIpmsiSrcAs OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The source-as in this inter-as I-PMSI."
    ::= { mvpnInterAsIpmsiEntry 3 }

mvpnInterAsIpmsiAttribute OBJECT-TYPE
```

```

SYNTAX          RowPointer
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "Points to a row in the l2L3VpnMcastPmsiTunnelAttributeTable."
 ::= { mvpnInterAsIpmsiEntry 4 }

-- Table of S-PMSIs advertised/received

mvpnSpmsiTable OBJECT-TYPE
SYNTAX          SEQUENCE OF MvpnSpmsiEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION
    "This table has information about the S-PMSIs sent/received
     by a device."
 ::= { mvpnStates 3 }

mvpnSpmsiEntry OBJECT-TYPE
SYNTAX          MvpnSpmsiEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION
    "An entry in this table is created or updated for every S-PMSI
     advertised/received in a particular MVRF."
INDEX { mplsL3VpnVrfName,
        mvpnSpmsiCmcastAddrType,
        mvpnSpmsiCmcastGroup,
        mvpnSpmsiCmcastGroupPrefixLen,
        mvpnSpmsiCmcastSource,
        mvpnSpmsiCmcastSourcePrefixLen,
        mvpnSpmsiOrigAddrType,
        mvpnSpmsiOrigAddress}
 ::= { mvpnSpmsiTable 1 }

MvpnSpmsiEntry ::= SEQUENCE {
    mvpnSpmsiCmcastAddrType      InetAddressType,
    mvpnSpmsiCmcastGroup         InetAddress,
    mvpnSpmsiCmcastGroupPrefixLen Unsigned32,
    mvpnSpmsiCmcastSource        InetAddress,
    mvpnSpmsiCmcastSourcePrefixLen Unsigned32,
    mvpnSpmsiOrigAddrType        InetAddressType,
    mvpnSpmsiOrigAddress         InetAddress,
    mvpnSpmsiTunnelAttribute     RowPointer,
    mvpnSpmsiUpTime              TimeInterval,
    mvpnSpmsiExpTime             TimeInterval,
    mvpnSpmsiRefCnt              Unsigned32
}

```

mvpnSpmsiCmcastAddrType OBJECT-TYPE  
SYNTAX InetAddressType  
MAX-ACCESS not-accessible  
STATUS current  
DESCRIPTION  
    "The Internet address type of mvpnSpmsiCmcastGroup/Source."  
 ::= { mvpnSpmsiEntry 1 }

mvpnSpmsiCmcastGroup OBJECT-TYPE  
SYNTAX InetAddress  
MAX-ACCESS not-accessible  
STATUS current  
DESCRIPTION  
    "S-PMSI C-multicast group address.  
    If it is 0 (or ::0), this is a wildcard group,  
    and mvpnSpmsiCmcastGroupPrefixLen must be 32 (or 128)."  
 ::= { mvpnSpmsiEntry 2 }

mvpnSpmsiCmcastGroupPrefixLen OBJECT-TYPE  
SYNTAX Unsigned32  
MAX-ACCESS not-accessible  
STATUS current  
DESCRIPTION  
    "S-PMSI C-multicast group address prefix length."  
 ::= { mvpnSpmsiEntry 3 }

mvpnSpmsiCmcastSource OBJECT-TYPE  
SYNTAX InetAddress  
MAX-ACCESS not-accessible  
STATUS current  
DESCRIPTION  
    "S-PMSI C-multicast source address  
    If it is 0 (or ::0), this is a wildcard source,  
    and mvpnSpmsiCmcastSourcePrefixLen must be 32 (or 128)."  
 ::= { mvpnSpmsiEntry 4 }

mvpnSpmsiCmcastSourcePrefixLen OBJECT-TYPE  
SYNTAX Unsigned32  
MAX-ACCESS not-accessible  
STATUS current  
DESCRIPTION  
    "S-PMSI C-multicast source address prefix length."  
 ::= { mvpnSpmsiEntry 5 }

mvpnSpmsiOrigAddrType OBJECT-TYPE  
SYNTAX InetAddressType  
MAX-ACCESS not-accessible  
STATUS current

## DESCRIPTION

"The Internet address type of mvpnSpmsiOrigAddress."

::= { mvpnSpmsiEntry 6 }

## mvpnSpmsiOrigAddress OBJECT-TYPE

SYNTAX InetAddress

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

"The BGP address of the device that originated the S-PMSI."

::= { mvpnSpmsiEntry 7 }

## mvpnSpmsiTunnelAttribute OBJECT-TYPE

SYNTAX RowPointer

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"A row pointer to the l2L3VpnMcastPmsiTunnelAttributeTable"

::= { mvpnSpmsiEntry 8 }

## mvpnSpmsiUpTime OBJECT-TYPE

SYNTAX TimeInterval

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The time since this S-PMSI  
was first advertised/received by the device."

::= { mvpnSpmsiEntry 9 }

## mvpnSpmsiExpTime OBJECT-TYPE

SYNTAX TimeInterval

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"For UDP-based S-PMSI signaling for PIM-MVPN,  
the amount of time remaining before this  
received S-PMSI Join Message expires,  
or the next S-PMSI Join Message refresh is to be  
advertised again from the device.  
Otherwise, it is 0."

::= { mvpnSpmsiEntry 10 }

## mvpnSpmsiRefCnt OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The number of c-multicast routes that are mapped to



```

        this S-PMSI."
 ::= { mvpnSpmsiEntry 11 }

-- Table of multicast routes in an MVPN

mvpnMrouteTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF MvpnMrouteEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This table augments ipMcastRouteTable, to provide some MVPN
        specific information."
    ::= { mvpnStates 4 }

mvpnMrouteEntry OBJECT-TYPE
    SYNTAX          MvpnMrouteEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "The mvpnMrouteEntry matches and augments an ipMcastRouteEntry,
        with MVPN specific information, such as PMSI used."
    AUGMENTS        { ipMcastRouteEntry }
    ::= { mvpnMrouteTable 1 }

MvpnMrouteEntry ::= SEQUENCE {
    mvpnMroutePmsiPointer          RowPointer,
    mvpnMrouteNumberOfLocalReplication  Unsigned32,
    mvpnMrouteNumberOfRemoteReplication Unsigned32,
    mvpnMrouteDataRate             Unsigned32
}

mvpnMroutePmsiPointer OBJECT-TYPE
    SYNTAX          RowPointer
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The I-PMSI or S-PMSI this C-multicast route is using.
        This is important because an implementation may not have an
        interface corresponding to a provider tunnel,
        that can be used in ipMcastRouteNextHopEntry."
    ::= { mvpnMrouteEntry 1 }

mvpnMrouteNumberOfLocalReplication OBJECT-TYPE
    SYNTAX          Unsigned32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "Number of replications to local receivers."

```

```
::= { mvpnMrouteEntry 2 }

mvpnMrouteNumberOfRemoteReplication OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "Number of (local) replications to remote receivers."
    ::= { mvpnMrouteEntry 3 }

mvpnMrouteDataRate OBJECT-TYPE
    SYNTAX      Unsigned32 (0..4294967295)
    UNITS       "kilobits per second"
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The data rate for traffic following this route."
    ::= { mvpnMrouteEntry 4 }

-- MVPN Notifications

mvpnMvrfChange NOTIFICATION-TYPE
    OBJECTS      {
        mvpnGenOperStatusChange
    }
    STATUS      current
    DESCRIPTION
        "A mvpnMvrfChange notification signifies a change about
        a MVRF in the device. The change event can be creation of
        the MVRF, deletion of the MVRF or an update on the I-PMSI
        or S-PMSI configuration of the MVRF. The change event
        is indicated by mvpnGenOperStatusChange embedded in
        the notification. The user can then query
        mvpnGeneralTable, and/or mvpnSpmsiConfigTable to
        get the details of the change as necessary."

        Note: Since the creation of a MVRF is often followed by
        configuration of I-PMSI and/or S-PMSIs for the MVRF,
        more than one (three at most) notifications for a MVRF may
        be generated serially, and it is really not necessary to
        generate all three of them. An agent may choose to generate a
        notification for the last event only, that is for S-PMSI
        configuration.

        Similarly, deletion of I-PMSI and S-PMSI configuration on a
        MVRF happens before a MVRF is deleted and it is recommended
        that the agent send the notification for MVRF deletion
        event only."
```

```
::= { mvpnNotifications 2 }

-- MVPN MIB Conformance Information

mvpnGroups      OBJECT IDENTIFIER ::= { mvpnConformance 1 }
mvpnCompliances OBJECT IDENTIFIER ::= { mvpnConformance 2 }

-- Compliance Statements

mvpnCompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
        "The compliance statement "
    MODULE -- this module
    MANDATORY-GROUPS {
        mvpnScalarGroup,
        mvpnGeneralGroup,
        mvpnSpmsiConfigGroup,
        mvpnSpmsiGroup,
        mvpnMrouteGroup
    }

    GROUP mvpnIpmsiGroup
    DESCRIPTION
        "This group is mandatory for systems that support
        BGP signaling for I-PMSI."

    GROUP mvpnInterAsIpmsiGroup
    DESCRIPTION
        "This group is mandatory for systems that support
        Inter-AS Segmented I-PMSI."

    GROUP mvpnBgpGeneralGroup
    DESCRIPTION
        "This group is mandatory for systems that support
        BGP-MVPN."

::= { mvpnCompliances 1 }

-- units of conformance

mvpnScalarGroup OBJECT-GROUP
    OBJECTS {
        mvpnMvrfNumber,
        mvpnMvrfNumberV4,
        mvpnMvrfNumberV6,
        mvpnMvrfNumberPimV4,
```

```
        mvpnMvrfNumberPimV6,
        mvpnMvrfNumberBgpV4,
        mvpnMvrfNumberBgpV6,
        mvpnMvrfNumberMldp,
        mvpnNotificationEnable
    }
STATUS          current
DESCRIPTION
    "These objects are used to monitor/manage
    global MVPN parameters."
::= { mvpnGroups 1 }

mvpnGeneralGroup    OBJECT-GROUP
OBJECTS {
    mvpnGenOperStatusChange,
    mvpnGenOperChangeTime,
    mvpnGenCmcastRouteProtocolV4,
    mvpnGenCmcastRouteProtocolV6,
    mvpnGenIpmsiConfigV4,
    mvpnGenIpmsiConfigV6,
    mvpnGenInterAsPmsiConfigV4,
    mvpnGenInterAsPmsiConfigV6,
    mvpnGenRowStatus
}
STATUS          current
DESCRIPTION
    "These objects are used to monitor/manage
    per-VRF MVPN parameters."
::= { mvpnGroups 2 }

mvpnPmsiConfigGroup    OBJECT-GROUP
OBJECTS {
    mvpnPmsiConfigEncapsType,
    mvpnPmsiConfigRowStatus
}
STATUS          current
DESCRIPTION
    "These objects are used to monitor/manage
    PMSI tunnel configurations."
::= { mvpnGroups 3 }

mvpnSpmsiConfigGroup    OBJECT-GROUP
OBJECTS {
    mvpnSpmsiConfigThreshold,
    mvpnSpmsiConfigPmsiPointer,
    mvpnSpmsiConfigRowStatus
}
STATUS          current
```

```
DESCRIPTION
    "These objects are used to monitor/manage
    S-PMSI configurations."
 ::= { mvpnGroups 4 }

mvpnIpmsiGroup      OBJECT-GROUP
OBJECTS {
    mvpnIpmsiUpTime,
    mvpnIpmsiAttribute
}
STATUS              current
DESCRIPTION
    "These objects are used to monitor/manage
    Intra-AS I-PMSI attributes."
 ::= { mvpnGroups 5 }

mvpnInterAsIpmsiGroup  OBJECT-GROUP
OBJECTS {
    mvpnInterAsIpmsiAttribute
}
STATUS              current
DESCRIPTION
    "These objects are used to monitor/manage
    Inter-AS I-PMSI attributes."
 ::= { mvpnGroups 6 }

mvpnSpmsiGroup      OBJECT-GROUP
OBJECTS {
    mvpnSpmsiTunnelAttribute,
    mvpnSpmsiUpTime,
    mvpnSpmsiExpTime,
    mvpnSpmsiRefCnt
}
STATUS              current
DESCRIPTION
    "These objects are used to monitor/manage
    S-PMSI attributes."
 ::= { mvpnGroups 7 }

mvpnMrouteGroup      OBJECT-GROUP
OBJECTS {
    mvpnMrouteNumberOfLocalReplication,
    mvpnMrouteNumberOfRemoteReplication,
    mvpnMrouteDataRate
}
STATUS              current
DESCRIPTION
    "These objects are used to monitor/manage
```

```
        VPN multicast forwarding states."
 ::= { mvpnGroups 8 }

mvpnBgpGeneralGroup OBJECT-GROUP
OBJECTS {
    mvpnBgpGenMode,
    mvpnBgpGenUmhSelection,
    mvpnBgpGenSiteType,
    mvpnBgpGenCmcastImportRt,
    mvpnBgpGenSrcAs,
    mvpnBgpGenSptnlLimit
}
STATUS current
DESCRIPTION
    "These objects are used to monitor/manage BGP-MVPN "
 ::= { mvpnGroups 9 }

mvpnOptionalGroup OBJECT-GROUP
OBJECTS {
    mvpnMroutePmsiPointer
}
STATUS current
DESCRIPTION
    "Support of these object is not required."
 ::= { mvpnGroups 10}
```

END

### 3 Security Considerations

<Security considerations text>

### 4 IANA Considerations

<IANA considerations text>

### 5 Acknowledgement

Some of the text has been taken almost verbatim from [CISCO-MIB].

We would like to thank Yakov Rekhter, Jeffrey Haas, Huajin Jeng, Durga Prasad Velamuri for their helpful comments.

### 6 References

#### 6.1 Normative References

- [KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4382] Nadeau, T., Ed., and H. van der Linde, Ed., "MPLS/BGP Layer 3 Virtual Private Network (VPN) Management Information Base", RFC 4382, February 2006.
- [MROUTE-MIB]McWalter, D., Thaler, D., and A. Kessler, "IP Multicast MIB", RFC 5132, December 2007.
- [MVPN] Eric C. Rosen, Rahul Aggarwal, et. al., Multicast in MPLS/BGP IP VPNs, RFC 6513.
- [BGP-MVPN] R. Aggarwal, E. Rosen, T. Morin, Y. Rekhter, BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs, RFC 6514.
- [L2L3MVPN-MIB] Zhang, J., L2L3 VPN Multicast MIB, draft-zzhang-l2l3-vpn-mcast-mib, Work In Progress.

## 6.2 Informative References

- [CISCO-MIB] Susheela Vaidya, Thomas D. Nadeau, Harmen Van der Linde, Multicast in BGP/MPLS IP VPNs Management Information Base, draft-svaidya-mcast-vpn-mib-02.txt, Work In Progress, April 2005.

## Authors' Addresses

Saud Asif  
AT&T  
C5-3D30  
200 South Laurel Avenue  
Middletown, NJ 07748  
USA  
Email: sasif@att.com

Andy Green  
BT Design 21CN Converged Core IP & Data  
01473 629360  
Adastral Park, Martlesham Heath, Ipswich IP5 3RE  
UK  
Email: andy.da.green@bt.com

Sameer Gulrajani  
Cisco Systems  
Tasman Drive  
San Jose, CA 95134

USA

Email: sameerg@cisco.com

Pradeep G. Jain  
Alcatel-Lucent Inc  
701 E Middlefield road  
Mountain view, CA 94043  
USA  
Email: pradeep.jain@alcatel-lucent.com

Jeffrey (Zhaohui) Zhang  
Juniper Networks, Inc.  
10 Technology Park Drive  
Westford, MA 01886  
USA  
Email: zzhang@juniper.net



Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 31, 2013

P. Marques  
Contrail Systems  
L. Fang  
Cisco Systems  
P. Pan  
Infinera Corp  
A. Shukla  
Juniper Networks  
M. Napierala  
AT&T Labs  
N. Bitar  
Verizon  
August 2012

BGP-signaled end-system IP/VPNs.  
draft-marques-l3vpn-end-system-07

#### Abstract

This document describes a solution in which the control plane protocol specified in BGP/MPLS IP VPNs [RFC4364] is used to provide a Virtual Network service to end-systems. These end-systems may be used to provide network services or may directly host end-to-end applications.

#### Status of this Memo

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

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

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

This Internet-Draft will expire on January 31, 2013.

#### Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the

document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
1.1. Terminology . . . . .	2
2. Requirements . . . . .	3
3. Applicability of BGP IP VPNs . . . . .	4
4. Virtual network end-points . . . . .	6
5. VPN Forwarder . . . . .	8
6. XMPP signaling protocol . . . . .	10
7. End-System Route Server behavior . . . . .	14
8. Operational Model . . . . .	14
9. Security Considerations . . . . .	17
10. Acknowledgements . . . . .	18
11. References . . . . .	18
11.1. Normative References . . . . .	18
11.2. Informational References . . . . .	18
Authors' Addresses . . . . .	19

## 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 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.

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 decouples the control plane and forwarding functionality of the PE device 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 Route Server** 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. This functionality may be co-located with the virtual interface or implemented by an external device.

## 2. Requirements

Network Virtualization is used in both service provider as well as enterprise networks to support multi-tenancy, network-based access control. It may also be used to facilitate end-system 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 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 end-systems that are allowed to exchange traffic freely, while traffic between virtual networks is subject to access controls. This scenario can be found in both 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.

End-systems 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.

End-system 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 recur to home agents or large L2 broadcast domains. Alternative techniques that are used in both Service Provider as well as enterprise networks.

IP Mobility requires the ability to "move" a device without disrupting its TCP (or UDP) transport sessions. These sessions often deploy second or sub-second keepalives to detect application failure. Experience with failure restoration in Service Provider networks shows that fast-failure restoration often requires the pre-provisioning of a restoration path.

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").

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 virtual interface (e.g., implemented in the hypervisor switch or host OS network drivers). It may also be external to the end-system residing in a data-center switch or specialized appliance.

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 directly exchange traffic with 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 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, then each Route Server is now responsible to generate 50 client updates. This example illustrates the linear scaling properties of BGP: doubling the number of Route Servers (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 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 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 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 available to software appliances 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" VPNs 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 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.

#### 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. Virtual network interfaces can be associated with a restricted list of applications via OS-dependent mechanisms, 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.

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.

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

A virtual network interface is connected to a VPN Forwarder. This VPN Forwarder MAY be co-located in the end-system or external.

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:

point-to-point interface model.

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 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 virtual interface moves between different machines.

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 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 Virtual Router Identifier (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 mac-address assigned to 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 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 client virtual interface may move between locations where redundancy may not be present.

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 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 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:

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

When the 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 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 Route Server 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.



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 6-byte value that uniquely identifies it in its administrative domain. This specification recommends the use of the MAC address of one of the physical ethernet interfaces.

Each VPN shall be identified by a 128 octet 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 End-System Route Servers. The control software relays to the End-System Route Server(s) VPN membership messages it receives from the end-system. VPN routing information received from the Route Server(s) SHOULD NOT be propagated to the end-system.

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

Subscription request from co-located VPN Forwarder to Route Server:

```
<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 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 a collection which is implicitly created by the End-System Route Server. Creation of a virtual interface may precede any IP address becoming active on the interface, as it is the case with VM instantiation.

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 VPN Forwarder to End-System Route Server:

```
<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 interface version # -->
          <label>10000</label> <!-- 20 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 End-System Route Server 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' received from the VPN Forwarder as the Network Layer Reachability Information (NLRI).

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

It optionally associates the route with an extended community TDB containing a version number of the virtual-interface.

Update notification from Route Server to VPN Forwarder:

```
<message to='system-id@domain.org' from='network-control.domain.org'>
  <event xmlns='http://jabber.org/protocol/pubsub#event'>
    <items node='vpn-customer-name'>
      <item id='ae890ac52d0df67ed7cfd51b644e901'>
        <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 interface version # -->
          <label>10000</label> <!-- 20 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 Route Server to the VPN Forwarder does not contain the system-id of the destination end-system. The "from" attribute in the 'message' element contains a "jid" associated with the Route Servers in the domain. 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 Forwarder.

When routes are withdrawn, the End-System Route Server generates both a "collection disassociate" request as well as a node "delete" request.

In situations where an automated system is controlling the instantiation of virtual interfaces it may be possible to have that system assign a non-decreasing version number for each instantiation of that particular interface. In that case, this number, carried in the 'version' field may be used to help gateways select the most recent instantiation of an interface during the interval of time where multiple routes are present.

## 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 end-system 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 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 already have locally originated route for the route target the 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.

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

End-System Route Servers SHOULD automatically assign a BGP route distinguisher per VPN routing table.

## 8. Operational Model

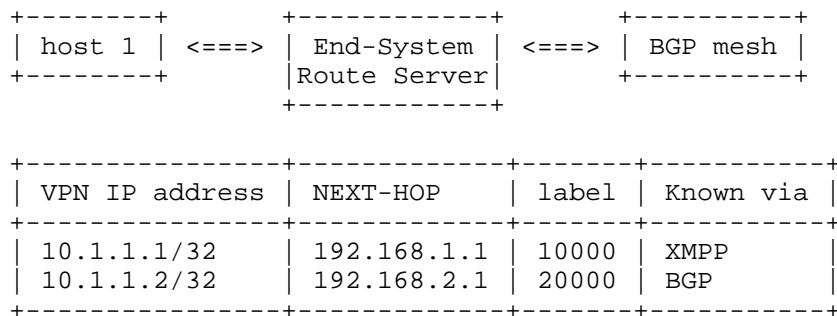
In the simplest case, a VPN is a collection of systems that are allowed to exchange traffic with each other and only with each other. Since all the forwarding tables in this VPN have the same routing entries they are often referred to as symmetrical VPNs.

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

Each of these hosts 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 1", the local XMPP client generates a XMPP subscription message to its respective Route Server. This message 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.

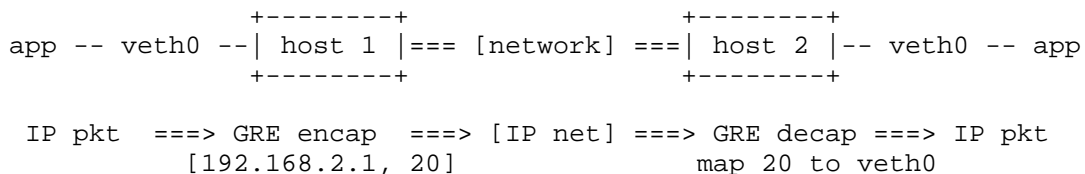
Once the interface is operational, "host 1" will publish any IP addresses that are configured on the respective virtual interface. This will in turn cause the End-System Route Server to advertise these (directly or indirectly) to any other BGP speaker on the network which is connected to an attachment point of that VPN.



VPN Routing table on Route Server

The figure above represents the contents of the VRF routing table on RS1 after the IPv4 address 10.1.1.1 has been added to the virtual interface on host 1. It assumes that there is another attachment 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 End-System Route Servers 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.



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 an application that uses the virtual interface 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.

In the case the virtual interface on host is associated with a guest OS, this guest OS has had its address resolution queries answered with the Virtual Router MAC address. As a result, that is the address it uses as the destination MAC address in packets it originates. This MAC address is not present on the GRE encapsulated packet.

End-System Route Servers 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 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 through a middle-box for inspection. Assuming 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. Accompanied with a change in the End-System Route Server 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 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. Security Considerations

The signaling protocol defines the access control policies for each virtual interface and any guest application associated with it. It is important to secure the end-system access to End-System Route Servers and the BGP infrastructure itself.

The XMPP session between end-systems and the Route Servers 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 Route Server.

BGP sessions MUST 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.

## 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-sdnf-flow-spec]

Marques, P., Fang, L., Pan, P., Shukla, A. and M.  
Napierala, "Traffic classification in end-system IP  
VPNs.", Internet-Draft draft-marques-sdnp-flow-spec-01,  
April 2012.

[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  
Contrail Systems  
2350 Mission College Blvd.  
Santa Clara, CA 95054

Email: roque@contrailsystems.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](mailto:nabil.bitar@verizon.com)

L3VPN Working Group  
Internet Draft  
Intended Status: Proposed Standard  
Updates: 6514  
Expires: February 20, 2013

IJsbrand Wijnands  
Eric C. Rosen  
Cisco Systems, Inc.  
  
Uwe Joorde  
Deutsche Telekom

August 20, 2012

## Encoding mLDP FECs in the NLRI of BGP MCAST-VPN Routes

draft-rosen-l3vpn-mvpn-mlbp-nlri-01.txt

### Abstract

Many service providers offer "BGP/MPLS IP VPN" service to their customers. Existing IETF standards specify the procedures and protocols that a service provider uses in order to offer this service to customers who have IP unicast and IP multicast traffic in their VPNs. It is also desirable to be able to support customers who have MPLS multicast traffic in their VPNs. This document specifies the procedures and protocol extensions that are needed to support customers who use the Multicast Extensions to Label Distribution Protocol (mLDP) as the control protocol for their MPLS multicast traffic. Existing standards do provide some support for customers who use mLDP, but only under a restrictive set of circumstances. This document generalizes the existing support to include all cases where the customer uses mLDP, without any restrictions.

### Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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

The list of current Internet-Drafts can be accessed at  
<http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at  
<http://www.ietf.org/shadow.html>.

#### Copyright and License Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

#### Table of Contents

1	Introduction .....	3
2	Why This Document is Needed .....	4
3	Encoding an mLDP FEC in the MCAST-VPN NLRI .....	5
4	Wildcards .....	7
5	IANA Considerations .....	8
6	Security Considerations .....	9
7	Acknowledgments .....	9
8	Authors' Addresses .....	9
9	Normative References .....	10
10	Informative References .....	10

## 1. Introduction

Many service providers (SPs) offer "BGP/MPLS IP VPN" service to their customers. When a customer has IP multicast traffic in its VPN, the service provider needs to signal the customer multicast states across the backbone. A customer with IP multicast traffic is typically using PIM ("Protocol Independent Multicast") [PIM] and/or IGMP ("Internet Group Management Protocol") [IGMP] as the multicast control protocol in its VPN. The IP multicast states of these protocols are commonly denoted as "(S,G)" and/or "(\*,G)" states, where "S" is a multicast source address and "G" is a multicast group address. [MVPN-BGP] specifies the way an SP may use BGP to signal a customer's IP multicast states across the SP backbone. This is done by using "Multiprotocol BGP" Updates whose "Subsequent Address Family" value is "MCAST-VPN" (5). The NLRI ("Network Layer Reachability Information") field of these Updates includes a customer Multicast Source field and a customer Multicast Group field, thus enabling the customer's (S,G) or (\*,G) states to be encoded in the NLRI.

It is also desirable for the BGP/MPLS IP VPN service to be able to support customers who are using MPLS multicast, either instead of, or in addition to, IP multicast. This document specifies the procedures and protocol extensions needed to support customers who use mLDP ("Multicast Extensions to Label Distribution Protocol") [mLDP] to create and maintain Point-to-Multipoint (P2MP) and/or Multipoint-to-Multipoint (MP2MP) Label Switched Paths (LSPs). While mLDP is not the only protocol that can be used to create and maintain multipoint LSPs, consideration of other MPLS multicast control protocols is outside the scope of this document.

When a customer is using mLDP in its VPN, the customer multicast states associated with mLDP are denoted by an mLDP "FEC Element" ("Forwarding Equivalence Class element", see [mLDP]), instead of by an (S,G) or (\*,G). Thus it is necessary to have a way to encode a customer's mLDP FEC Elements in the NLRI field of the BGP MCAST-VPN routes.

While [MVPN-BGP] does specify a way of encoding an mLDP FEC Element in the MCAST-VPN NLRI field, the encoding specified therein makes a variety of restrictive assumptions about the customer's use of mLDP. (These assumptions are described in section 2 of this document.) The purpose of this document is to update [MVPN-BGP] so that customers using mLDP in their VPNs can be supported even when those assumptions do not hold.

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 [RFC2119].

## 2. Why This Document is Needed

An mLDP FEC Element consists of a FEC Type, a Root Node, and an Opaque Value. mLDP uses several FEC types, and in particular, uses the FEC type to distinguish between P2MP LSPs and MP2MP LSPs.

Section 11.1.2 of [MVPN-BGP] ("Originating routes: mLDP as the C-multicast control protocol") states:

Whenever a PE receives from one of its CEs a P2MP Label Map <X, Y, L> over interface I, where X is the Root Node Address, Y is the Opaque Value, and L is an MPLS label ... the PE constructs a Source Tree Join C-multicast route whose MCAST-VPN NLRI contains X as the Multicast Source field, and Y as the Multicast Group field.

In other words, the Root Node of the mLDP FEC Element appears in the Multicast Source Field, and the Opaque Value of the mLDP FEC Element appears in the Multicast Group field.

This method of encoding an mLDP FEC in an MCAST-VPN NLRI can only be used if all of the following conditions hold:

1. A customer using mLDP is not also using PIM/IGMP.

The encoding in [MVPN-BGP] does not specify any way in which one can determine, upon receiving a BGP Update, whether the Multicast Group field contains an IP address or whether it contains an mLDP FEC Element Opaque Value. Therefore it may not uniquely identify a customer multicast state if the customer is using both PIM/IGMP and mLDP in its VPN.

2. A customer using mLDP is using only the mLDP P2MP FEC Element, and is not using the mLDP MP2MP FEC Element.

The encoding in [MVPN-BGP] does not specify any way to encode the type of the mLDP FEC Element; it just assumes it to be a P2MP FEC Element.

3. A customer using mLDP is using only an mLDP Opaque Value type for which the Opaque Value is exactly 32 bits or 128 bits long.

The use of Multicast Group fields that have other lengths is declared by [MVPN-BGP] to be "out of scope" of that document (see, e.g., section 4.3 of that document).



This condition holds if the customer uses only the mLDP "Generic LSP Identifier" Opaque Value type (defined in [mLDP]). However, mLDP supports many other Opaque Value types whose length is not restricted to be 32 or 128 bits.

The purpose of this document is to update [MVPN-BGP] so that customers using mLDP can be supported, even when these conditions do not hold.

In addition, neither [MVPN-BGP] nor [MVPN-WILDCARDS] addresses the use of "wild cards" when the MCAST-VPN NLRI encodes an mLDP FEC. This document specifies a way to encode mLDP FEC Element wild cards in the NLRI of the relevant BGP MCAST-VPN routes.

### 3. Encoding an mLDP FEC in the MCAST-VPN NLRI

This section specifies the way to encode an mLDP FEC element in the NLRI of the following three MCAST-VPN route types defined in [MVPN-BGP]:

- C-multicast Source Tree Join,
- S-PMSI A-D route, and
- Leaf A-D route.

The other four MCAST-VPN route types defined in [MVPN-BGP] do not ever need to carry mLDP FEC Elements. The C-multicast Shared Tree Join route and the Source Active A-D route are used to communicate state about unidirectional shared trees; since mLDP does not have unidirectional shared trees, these routes are not used to signal mLDP states. The Intra-AS I-PMSI A-D route and the Inter-AS I-PMSI A-D route do not identify specific customer multicast states, and hence do not carry any information that is specific to the customer's multicast control protocol.

Per [MVPN-BGP], the first octet of the NLRI of an MCAST-VPN route is a "route type". Only values 1-7 are defined. The high order 5 bits of that octet are thus always zero.

This document updates [MVPN-BGP] by specifying a use for the high order 2 bits of the "route type" octet. The following two values are defined:

- If the two high order bits are both zero, the NLRI is as specified in [MVPN-BGP] and/or [MVPN-WILDCARDS].
- If the two high order bits have the value 01, the NLRI encoding is modified as follows: the "Multicast Source Length", "Multicast Source", "Multicast Group" length, and "Multicast Group" fields are omitted, and in their place is a single mLDP FEC Element, as defined in [mLDP]. See section 2.2 of [mLDP] for a diagram of the mLDP FEC element.

The other two possible values (11 and 10) for the two high order bits may be used at a later time to identify other multicast control protocols.

As a result, the NLRI of an S-PMSI A-D route with an mLDP FEC in its NLRI will consist of a Route Distinguisher, followed by the mLDP FEC, followed by the "Originating Router's IP Address Field".

The NLRI of a C-multicast Source Tree Join route with an mLDP FEC in its NLRI will consist of a Route Distinguisher, followed by the Source AS, followed by the mLDP FEC.

In a Leaf A-D route that has been derived from an S-PMSI A-D route, the "route key" field remains the NLRI of the S-PMSI A-D route from which it was derived.

In a Leaf A-D route that has not been derived from an S-PMSI A-D route, the "route key" field is as specified in [SEGMENTED-MVPN], except that the "Multicast Source Length", "Multicast Source", "Multicast Group" length, and "Multicast Group" fields are omitted, and in their place is a single mLDP FEC Element. Thus the route key field consists of a Route Distinguisher, an mLDP FEC element, and the IP address of the Ingress PE router.

An mLDP FEC element contains an "address family" field from IANA's "Address Family Numbers" registry. This identifies the address family of the "root node address" field of the FEC element. When an mLDP FEC element is encoded into the NLRI of an a BGP update whose SAFI is MCAST-VPN, the address family of the root node (as indicated in the FEC element) MUST "correspond to" the address family that is identified in the AFI field of that BGP update. These two "address family" fields are considered to "correspond" under the following conditions:

- they contain identical values, or

- the BGP update's AFI field identifies IPv4 as the address family, and the mLDP FEC element identifies "Multi-Topology IPv4" as the address family of the root node, or
- the BGP update's AFI field identifies IPv6 as the address family, and the mLDP FEC element identifies "Multi-Topology IPv6" as the address family of the root node.

For more information about the "multi-topology" address families, see [LDP-MT] and [mLDP-MT].

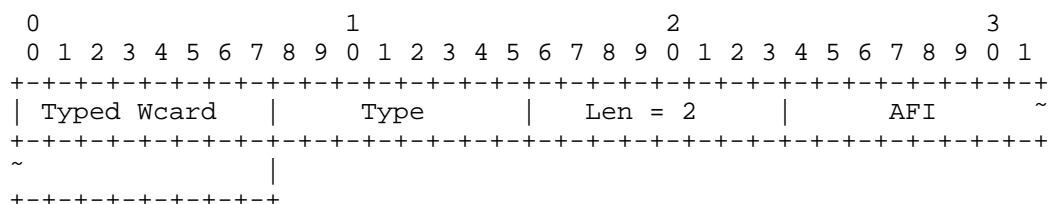
#### 4. Wildcards

[MVPN-WILDCARDS] specifies encodings and procedures that allow "wildcards" to be specified in the NLRI of S-PMSI A-D routes. A set of rules are given that specify when a customer multicast flow "matches" a given S-PMSI A-D route whose NLRI contains wildcards. However, the use of these wildcards is defined only for the case where the customer is using PIM as its multicast control protocol. In this section, we define the wildcard encodings for the case where the customer is using mLDP as its multicast control protocol.

Customer mLDP Multipoint LSPs do NOT ever match S-PMSI A-D routes containing the wildcards specified in [MVPN-WILDCARDS].

To specify a wildcard that can be matched by a customer mLDP Multipoint LSP, one encodes an mLDP "typed wildcard FEC" [LDP-WC] into the NLRI of the S-PMSI A-D route.

The mLDP typed wildcard FEC is specified in section 9 of [mLDP], which includes the following diagram:



The field "Typed Wcard" contains the value in the IANA LDP Registry "Forwarding Equivalence Class (FEC) Type Name Space" that is assigned to "Typed Wildcard FEC Element" (i.e., 5).

The AFI field contains an address family identifier, from IANA's

"Address Family Numbers" registry.

The "Type" field MUST either be set to zero, or contain one of the following values from the IANA LDP Registry "Forwarding Equivalence Class (FEC) Type Name Space":

- P2MP FEC (6)
- MP2MP-Up FEC (7)
- MP2MP-Down FEC (8)

If the type field is set to "P2MP-FEC", the wildcard FEC element means "any P2MP FEC whose root node address is of the specified address family".

If the type field is set to "MP2MP-Up" or "MP2MP-Down", the wildcard FEC element means "any MP2MP FEC" whose root node address is of the specified address family. When generating this wildcard FEC, the value "MP2MP-Down" SHOULD be used.

If the type field is set to 0, the wildcard FEC element means "any P2MP or MP2MP" FEC whose root node address is of the specified address family.

A future revision of this document will discuss use cases, and provide a more detailed set of procedures for using these wildcards.

## 5. IANA Considerations

[MVPN-BGP] does not create a registry for the allocation of new MCAST-VPN Route Type values. In retrospect, it seems that it should have done so. IANA should create a registry called "MCAST-VPN Route Types", referencing this document and [MVPN-BGP]. The allocation policy should be "Standards Action with Early Allocation", and the assignable values are in the range 0-0xFF. The following values should be assigned:

- 0x00: Reserved
- 0x01: Intra-AS I-PMSI A-D route (reference: [MVPN-BGP])
- 0x02: Inter-AS I-PMSI A-D route (reference: [MVPN-BGP])

- 0x03: S-PMSI A-D route for PIM as the C-multicast control protocol (reference: [MVPN-BGP])
- 0x43: S-PMSI A-D route for mLDP as the C-multicast control protocol (reference: this document)
- 0x04: Leaf A-D route for PIM as the C-multicast control protocol (reference: [MVPN-BGP])
- 0x44: Leaf A-D route for mLDP as the C-multicast control protocol (reference: this document)
- 0x05: Source Active A-D route for PIM as the C-multicast control protocol (reference: [MVPN-BGP])
- 0x06: Shared Tree Join route for PIM as the C-multicast control protocol (reference: [MVPN-BGP])
- 0x07: Source Tree Join route for PIM as the C-multicast control protocol (reference: [MVPN-BGP])
- 0x47: Source Tree Join route for mLDP as the C-multicast control protocol (reference: this document)

## 6. Security Considerations

No new security issues.

## 7. Acknowledgments

The authors wish to thank Pradosh Mohapatra and Saquib Najam for their ideas and comments. We also thank Yakov Rekhter for his comments.

## 8. Authors' Addresses

IJsbrand Wijnands  
Cisco Systems, Inc.  
De kleetlaan 6a Diegem 1831  
Belgium  
E-mail: ice@cisco.com

Eric C. Rosen  
Cisco Systems, Inc.  
1414 Massachusetts Avenue  
Boxborough, MA, 01719  
E-mail: erosen@cisco.com

Uwe Joerde  
Deutsche Telekom  
Hammer Str. 216-226  
D-48153 Muenster, Germany  
E-mail: Uwe.Joerde@telekom.de

## 9. Normative References

[LDP-WC] Asati, R., Minei, I., and B. Thomas, "Label Distribution Protocol (LDP) 'Typed Wildcard' Forward Equivalence Class (FEC)", RFC 5918, August 2010.

[mLDP] "Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths", Wijnands, Minei, Kompella, Thomas, RFC 6388, November 2011

[MVPN-BGP] "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", Aggarwal, Rosen, Morin, Rekhter, RFC 6514, February 2012

[MVPN-WILDCARDS], "Wildcards in Multicast VPN Auto-Discovery Routes", Rosen, Rekhter, Hendrickx, Qiu, RFC 6625, may 2012

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels.", Bradner, March 1997

## 10. Informative References

[IGMP] "Internet Group Management Protocol, Version 3", Cain, Deering, Kouvelas, Fenner, Thyagarajan, RFC 3376, October 2002

[LDP-MT] "LDP Extension for Multi-Topology Support", Zhao, et. al., draft-ietf-mppls-ldp-multi-topology-04.txt, July 2012

[mLDP-MT] "mLDP Extensions for Multi Topology Routing", Wijnands, Raza, draft-iwijnand-mppls-mldp-multi-topology-02.txt, July 2012

[PIM] "Protocol Independent Multicast - Sparse Mode (PIM-SM)", Fenner, Handley, Holbrook, Kouvelas, August 2006, RFC 4601

[SEGMENTED-MVPN] "Inter-Area P2MP Segmented LSPs", Rekhter, Aggarwal,  
Morin, Grosclaude, Leymann, Saad, draft-ietf-mpls-seamless-  
mcast-05.txt, August 2012

INTERNET-DRAFT  
Intended Status: Informational  
Expires: September 6, 2012

Paul Unbehagen  
Roger Lapuh  
Avaya  
Sue Hares  
Peter Ashwood-Smith  
Hauwei

March 5, 2012

IP/IPVPN services with IEEE 802.1aq SPB networks  
draft-unbehagen-spb-ip-ipvpn-00.txt

## Abstract

This document describes a compact method of using a IEEE 802.1aq Shortest Path Backbone Bridging (SPB) network to natively enable and carry IP and IPVPN services on native Ethernet links. Further this documents the extensions to SPB's control protocol, IS-IS, required to allow it to be a single mechanism for providing all these services types. On its own SPB provides virtual Ethernet networks; utilizing IS-IS to create loop free Ethernet topologies that forward Ethernet traffic using a standard Ethernet header. This document shows how the same SPB network can also be leveraged to provide IP based services.

## Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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

The list of current Internet-Drafts can be accessed at  
<http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at  
<http://www.ietf.org/shadow.html>



## Copyright and License Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1	Introduction . . . . .	3
1.1	Terminology . . . . .	3
2.	SPB control of BMAC forwarding . . . . .	4
3.	IP forwarding with SPB . . . . .	5
3.1.	IP Unicast . . . . .	5
4.	IPVPN services with SPB . . . . .	6
4.1.	Route Propagation Techniques . . . . .	6
4.2.	Ethernet underlay modes - 802.1aq and/or 802.1Qbp . . . . .	8
4.3.	I-TAG Encapsulation . . . . .	9
5.	Interworking with MPLS based Networks . . . . .	10
6.	OAM . . . . .	12
7.	Security Considerations . . . . .	12
8.	IANA Considerations . . . . .	12
9.	References . . . . .	12
9.1	Normative References . . . . .	12
9.2	Informative References . . . . .	12
10.	Acknowledgments . . . . .	13
	Authors' Addresses . . . . .	13

## 1 Introduction

The IEEE has defined a method for L2 virtualization called Shortest Path Bridging (SPB), which is leveraging IS-IS as a new Ethernet control plane to control the BMAC layer of the 802.1ah PBB encapsulation, replacing Ethernet's flooding and learning as the backbone forwarding protocol. In addition to layer 2 (bridging), the 24 bits of the Service Instance defined in the 802.1ah frame format can also be leveraged for any network connectivity service including layer 3 Unicast and Multicast for IPv4 and IPv6. This document outlines the proposed extensions to ISIS-SPB to enable this functionality. The benefits of leveraging one protocol (ISIS-SPB) to provide any type of connectivity service on top of Ethernet are significant.

SPB, through the use of ISIS to exchange the connectivity service topology, provides a powerful end-point-only provisioning model. IP/SPB leverages this and extends this to Layer 3, thus not only L2 VPNs can be formed by attaching Virtual LANs to service IDs (ISIDs), but also L3 VPNs can be formed by binding Virtual Route Forwarders (VRFs) to ISIDs at the service attachment points.

Due to the fact that all connectivity services described above are using the same SPB forwarding plane defined in IEEE802.1aq/.1Qbp, network availability is defined by the convergence timing of the single ISIS-SPB control plane.

### 1.1 Terminology

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].

[IEEE802.1aq] defines a technology for providing a link state protocol for the control of a common Ethernet switching layer.

[IEEE802.1ah] Provider Backbone Bridging is a set of architecture and protocols for transporting of a customer network traffic over a provider's network

- BCB - Backbone Core Bridge
- BDA - Backbone Destination Address
- BEB - Backbone Edge Bridge
- BMAC - Backbone MAC Address

BSA - Backbone Source Address

BVID - Backbone VLAN ID

ESP - Ethernet Switched Path

ISID - Service Identifier

ISIS - Intermediate System to Intermediate System Routing Protocol

ISIS-SPB - ISIS extensions for SPB

MDT - Multicast Distribution Tree

SPF - Shortest Path Forwarding

SPB - Shortest Path Bridging

TLV - Type Length Value

VLAN - Virtual LAN

## 2. SPB control of BMAC forwarding

SPB uses the terms Backbone Core Bridge (BCB) and Backbone Edge Bridge (BEB) to describe the functions of network nodes in the network. These terms describe features that are similar in function to the PE and P nodes in an MPLS network.

SPB enables a loop free construction of Ethernet switched paths between every SPB enabled node by reusing some existing components of IS-IS and by adding a small set of new IS-IS TLVs. SPB nodes use a standard IS-IS mechanism of operation for neighbor discovery and database distribution. SPB utilizes an IS-IS based on IS-IS Ethernet link level peering protocol so it does not depend on link level IP addressing. Also, due to the fact that the links are forwarding on the source and destination information in the Ethernet header, there is no requirement to verify that each node can do IP routing. Multicast Forwarding entries (BMAC FDB or FIB) on each node are constructed based on a combination of a nodal unique identifier and a administratively controlled service identifier providing multicast trees that can be adjusted to match a desired service granularity.

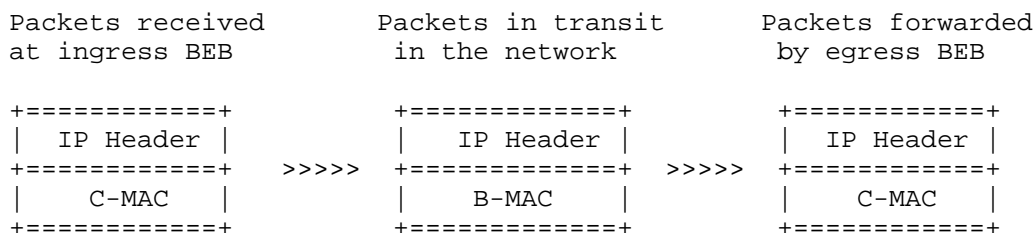
Each node uses standard IS-IS methods of sharing link state PDU's. Those PDUs contain the System-ID of each node, the attached neighbors and information such as EVPN ISIDs for SPB. A unicast SPF process runs on each switch to construct the unicast connectivity of each

switch to every other switch based on these nodal MAC's (BMACs) derived from the System-ID. Each node that is on the shortest path between any other given nodes will install corresponding FDB entries only on their associated ports. This has the added benefit that Ethernet FDB entries exist only on nodes that are the source, root, or tandem point for a give SPF ESP between any given set of nodes. Any node that does not need to participate in the tandem calculations may use the IS-IS overload bit to exclude tandem paths and behave as only the root or source for any given service traffic.

### 3. IP forwarding with SPB

IP unicast and multicast can leverage this base BMAC switching layer by mapping IP to the Ethernet service points. For unicast forwarding the standard mechanisms of IS-IS IP route propagation can be used to associate remote IP networks to the far end nodal Ethernet address.

The encapsulation of IP unicast packets would use the standard method to include an Ethertype of 0x800 behind the BMAC header.



#### 3.1. IP Unicast

The native unicast entries SPB constructs, provide a simple way of enabling end to end switching of IP packets along a deterministic Ethernet Switched Path (ESP). Knowledge of the location of IP subnet's is achieved by utilizing the existing functionality of IS-IS TLVs for IPv4 and IPv6. The control plane operation becomes one of simply creating FDB entries that map IP routes to their points of presence. In other words the FDB entry for an IP route simply gives the last hop BMAC of the BEB which advertised that route. No shortest path computations are required since that has already been accomplished by the SPB layer. For IP subnet awareness the existing IS-IS TLVs are used to propagate routes. The encapsulation is the standard IP in Ethernet encapsulation.

To enable this behavior, this document specifies an efficient learning and forwarding operation where an edge BEB provides a

standard IP interface to its attached IP devices. The BEB performs a route lookup on the destination IP addresses which will resolve to a remote BMAC to forward towards on the SPB portion of the network, and then encapsulates the IP packet in a Ethernet header using the unicast BMAC operation with a standard IPv4 or IPv6 Ethertype. In essence this is IP over Ethernet where the Ethernet is a BMAC based Ethernet. One simple way to think of this is that existing ISIS for IP implementations forward IP packets to the next hop router, while this mechanism forwards an IP packet directly to the last hop BEB within the SPB domain thereby eliminating IP forwarding operations on tandem devices.

#### 4. IPVPN services with SPB

Using the TLV extensions described below it is possible to extend SPB to not only provide EVPN and the above described IP connectivity, but also provide virtualized solutions for IP services such as IPVPNs. The next sections will outline the route propagation techniques for two different deployment models.

Two common modes of carrying information are: BGP or ISIS [ISIS]. The IS-IS method will be described in this version of the draft. The control plane encapsulation of VRF routes passed in BGP is similar to the method used here with ISIS.

##### 4.1. Route Propagation Techniques

The ability to share routes within a network can be provided within IS-IS. To accomplish this flexibility the use of a new IPVPN TLV and an optional sub-TLV is proposed.

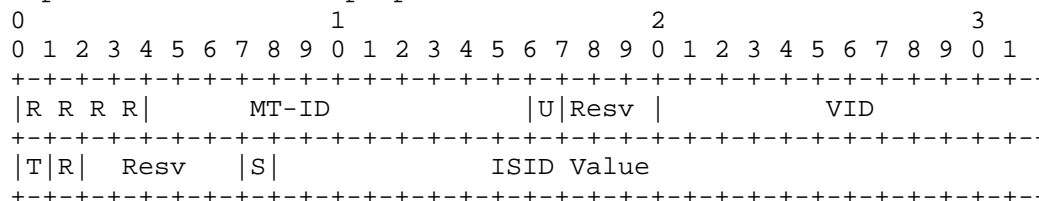


Figure 2 IPVPN TLV

The TLV MAY appear any number of times (including none) within a Link State PDU.

The up/down bit used for notification if this TLV has been leaked down into a L1.

The T bit and R bit are used to signal to the other nodes whether to construct Multicast trees to and from this announcing node depending on the combination of bits set to 1. If neither the T nor R bit is

set, then the IPVPN service is unicast only. The next 4 bits are reserved and the S bit is used to signify that the VPN-Route Sub-TLVs are present.

Sub-TLVs may be set to carry specific routes for each VPN within IS-IS. This allows for routes from multiple VPNs to be carried under their respective VPN ISID IDs and allow for overlapping IP subnet's.

IP/SPB VPN ISIDs are comparable to RFC4364 Route distinguishers and route targets to separate VPN traffic in the control plane. VPN routes are exchanged through IS-IS and can be distinguished by the individual ISID they are associated with. In order to form different types of IP VPNs on BEB's, routes can be exported into ISID's or imported from ISID's.

This document also defines the following new sub-TLV types that need to be reflected in the IS-IS sub-TLV registry for the SPB ipvpn TLV:

Sub-Type	Description
-----	-----
1	IPv4 Prefix information
2	IPv6 Prefix information
3	Reserved
4	Reserved

The encapsulation is as follows:

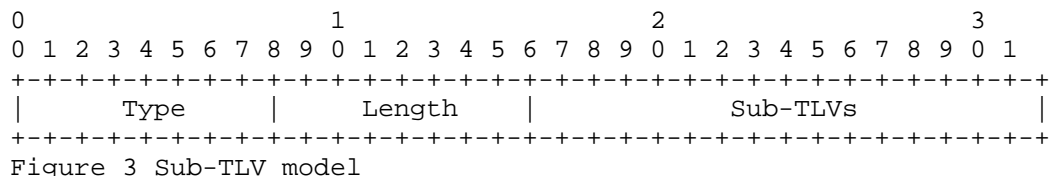


Figure 3 Sub-TLV model

The sub-TLVs are designed to mimic the top level TLVs for IP reachability within the VPN. Allowing for a given VRF to support multiple IP service models within a single VPN-ID.

Sub-TLV 1 is similar as the Extended IPv4 TLV 135 and MAY appear multiple times in the same TLV with a data structure consisting of:

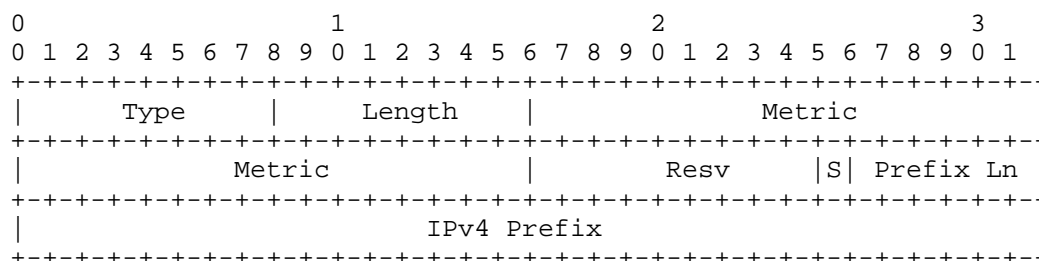


Figure 4 IPv4 sub-TLV

Sub-TLV 2 is similar to the IPv6 TLV 236 and MAY appear multiple times in the same TLV with a data structure consisting of:

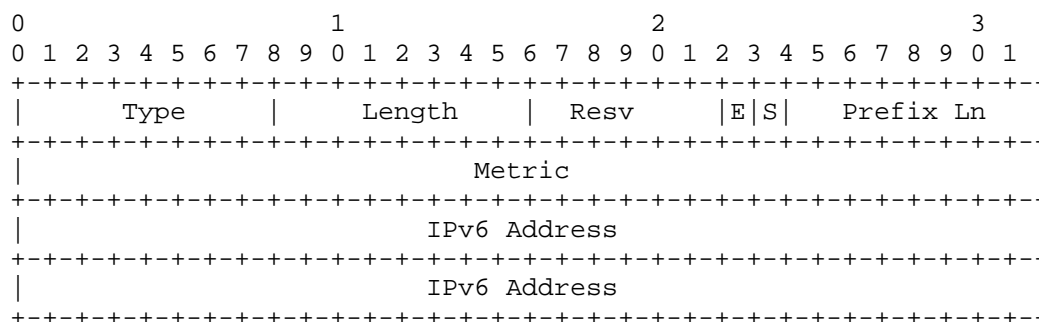


Figure 5 IPv6 sub-TLV

#### 4.2. Ethernet underlay modes - 802.1aq and/or 802.1Qbp

Shortest Path Bridging supports two major modes for L2VPNs via 802.1aq [SPB] and 802.1Qbp [ECMP] the behaviors of which may be selected on a per ISID basis for carriage of L3VPNs as follows.

The 802.1aq [SPB] L2 underlay on which IPVPN packets flow, supports deterministic and symmetric multiple equal cost routes with hashing to the different routes at the head end via normal IP n-tuple or other micro flow order preserving hash mechanisms. To obtain this behavior the IPVPN TLV VID field MUST contain a BVID value which has been associated with one of the 802.1aq ECT-ALGORITHMS by the SPB underlay in the SPB Base VLAN-Identifiers sub-TLV. This will cause the ISID information which follows to have 802.1aq semantics - i.e. (S,G) multicast and symmetric/congruent routing. The normal 802.1aq ECT-ALGORITHM values 00-80-C2-01 thru 00-80-C2-10 and their associated VIDs are advertised normally in the IIH and LSPs for 802.1aq and the head end IPVPN behavior is free to hash over any or all of those VIDs.

The 802.1Qbp [ECMP] extensions to [SPB] supports a flow-tag with a FlowId and TTL resulting in a per hop hashed choice of next hop by L2 forwarding. To obtain this behavior the IPVPN TLV VID field MUST contain a BVID value which has been associated with one of the 802.1Qbp ECT-ALGORITHMS by the SPB underlay in the SPB Base VLAN-Identifiers sub-TLV. This will cause the ISID information which follows to have 802.1Qbp semantics - i.e. (\*,G) and/or (S,G) multicast and non deterministic non symmetric routing. The normal 802.1Qbp ECT-ALGORITHM value 00-80-C2-11 and its associated VIDs are advertised normally in the IIH and LSPs for 802.1Qbp and the head end IPVPN behavior is free to hash over all ECMP L2 next hops for this VID and to insert a flow-tag with appropriate TTL value and FlowId based on the L3 hash result. The TTL and FlowId will then be used to enable tandem .1Qbp ECMP behavior without knowledge of or inspection of the L3VPN headers.

#### 4.3. I-TAG Encapsulation

For the forwarding of IPVPN traffic the use of the standard 802.1ah I-TAG would require the encapsulation of a nulled out CMAC address, since the traffic is IP and routed at each BEB there is no need for a CMAC. Therefore the more optimal encapsulated method used for IPVPN traffic within the SPB network is to use the ISID portion of the I-TAG without a CMAC header, called the short I-TAG. This allows the network to carry forward the benefits of the global label/VPN ID purpose of the ISID without enforcing unnecessary header overhead.

The encapsulation of IP packets being forwarded for a IPVPN would use a short I-Tag (ISID with no CMAC) behind the BMAC header. While the short I-TAG is the recommended mode for carrying IPVPN traffic this standard permits the inclusion of a zeroed out CMAC. In this case the sender MUST zero out the CMAC on transmission and MUST ignore a non-zero CMAC on receipt.

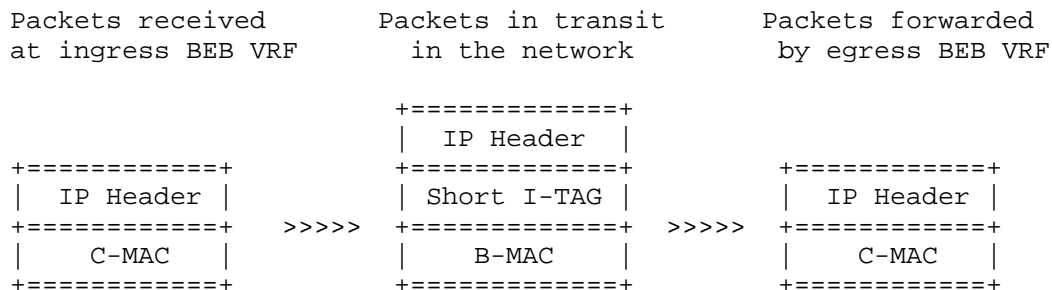


Figure 7 Short I-TAG encapsulation of IPVPN Packets



Where the ISID encapsulation is as follows. Directly between the VLAN and IP header.

```

                                .1ah I-TAG TCI
+-----+-----+-----+-----+-----+-----+-----+-----+
| P/DE  | R1  | R2  |                                     I-SID  |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Figure 9 Short I-Tag

P/DE 3 Bits Priority, 1 bit Drop Eligible

R1 Res1

R2 Res2

## 5. Interworking with MPLS based Networks

Any IP/SPB and/or IPVPN SPB network may exist on its own or may be part of a larger network. For example it may be attached to a standard IP or IP/MPLS network. This section will define a use case for using an SPB base network as a method of extending IPVPNs from a IP/MPLS core network. A similar model exists for the ELAN service that may span both a SPB and VPLS portions of the network as defined in [PBBVPLS]. The scale and size of the SPB portion of the network and network devices can then be utilized more efficiently as a single protocol will drain less resources and thereby allow the IPVPN VRFs extend closer to the end customer. Another benefit is that packets that are destined within the SPB network can be forwarded directly in the SPB portion of the domain without needing to be processed by the MPLS PE.

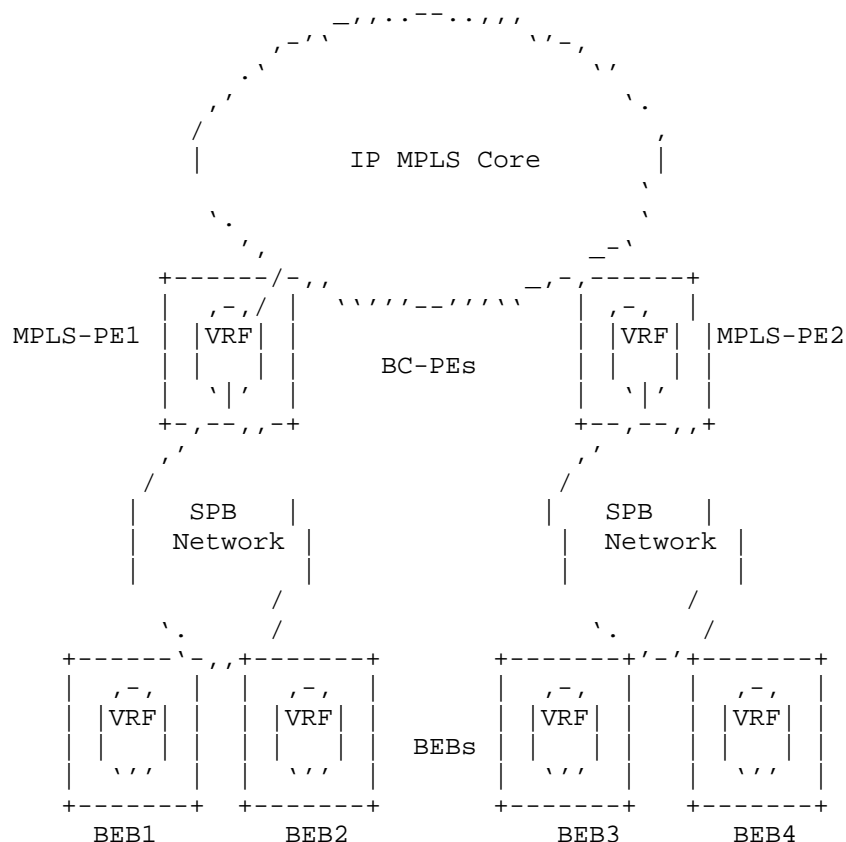


Figure 12 MPLS Interworking

A vrf does Service Address Encapsulation on a VPN basis with some entries of the VRF having MPLS labels and some have IPVPN SPB entries based on the direction from which the route was learned. The VRF also gets information from different topologies. Routes learned via BGP have FIB entries pointing to the appropriate next hop and Label set. While routes learned from the IPVPN ISID SPB domain have FDB entries for the appropriate BMAC address and ISID.

Route propagation to and from each domain happens naturally via the protocol interaction within a given VRF. Routes learned from the SPB domain are automatically announced into the BGP domain or may have policies applied and routes learned from other PE's from BGP are automatically propagated towards the CE's by injecting them into the SPB domain as a sub-TLV under the VRF's IPVPN ISID TLV.

## 6. OAM

Various techniques exist within the Ethernet standards space for scalable management of Ethernet based networks. For example OAM techniques defined in the IEEE 802.1ag and the ITU Y.1731 can be used for the IP based services similarly as they are defined for the EVPNs

## 7. Security Considerations

The extensions defined in this document do not incur any additional security considerations. Any IS-IS SPB network may utilize the security enhancements already defined within the IS-IS working group.

## 8. IANA Considerations

IP/SPB requires that IANA/ISO allocate a new number from ISIS-TLV Codepoints for the IPVPN TLV.

IP/SPB also requires that IANA/ISO allocate a new registry for sub TLV values within the above TLV. The first four values being:

IPV4	2	IPV6	3	reserved	4	reserved	1
------	---	------	---	----------	---	----------	---

## 9. References

### 9.1 Normative References

- [KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [MTISIS] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)", RFC 5120, February 2008.
- [IPVPN] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.
- [802.1AQ] "IEEE P802.1aq/D4.5 Draft Standard for Local and Metropolitan Area Networks -- Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks, Amendment 8: Shortest Path Bridging", IEEE 802.1aq D4.5, Feb 6, 2012
- [802.1AQ] Ashwood-Smith, P, Fedyk, D., "IS-IS Extensions Supporting IEEE 802.1aq Shortest Path Bridging" RFC 6329, XXX, 2012.

### 9.2 Informative References

- [IEEE802.1ah] "IEEE Standard for Local and Metropolitan Networks, Virtual Bridged Local Area Networks, Amendment 7: Provider Backbone Bridges" IEEE Std 802.1ah - 2008 amendment to IEEE 802.Q - 2005.
- [ISIS] ISO/IEC 10589:2002, "Intermediate system to Intermediate system routing information exchange protocol," ISO/IEC10589:2002.
- [PBBVPLS] Extensions to VPLS PE model for Provider Backbone Bridging, IETF, Internet Draft, draft-ietf-l2vpn-pbb-vpls-pe-model-00.txt, Work in Progress, May 12 2009

## 10. Acknowledgments

The authors would like to thank Don Fedyk for input into the contents of this document. And also Dave Allan, Nigel Bragg, Gautam Khara, Srikanth Keesara and Harish Sankaran for their detailed review of larger work that is behind this memo.

This document was prepared using nroff.

## Authors' Addresses

Paul Unbehagen Jr  
Avaya  
1300 W. 120th Avenue  
Westminster, CO 80234 USA  
Email: unbehagen@avaya.com

Roger Lapuh  
Avaya  
Flughofstrasse 54  
8152 Glatthbrugg  
Switzerland  
Email: rlapuh@avaya.com

Peter Ashwood-Smith  
Huawei Technologies Canada LTD.  
303 Terry Fox drive, Suite 400  
Kanata, Ontario , K2K 2J1, Canada  
Email: peter.ashwoodsmith@huawei.com

Susan Hares  
Huawei  
Email: shares@ndzh.com  
1-734-604-0332



Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: April 20, 2013

IJ. Wijnands, Ed.  
Cisco Systems  
P. Hitchen  
BT  
N. Leymann  
Deutsche Telekom  
W. Henderickx  
Alcatel-Lucent  
A. Gulko  
Thomson Reuters  
October 17, 2012

Multipoint Label Distribution Protocol  
In-Band Signaling in a VRF Context  
draft-wijnands-l3vpn-mlbp-vrf-in-band-signaling-01

Abstract

Sometimes an IP multicast distribution tree (MDT) traverses both MPLS-enabled and non-MPLS-enabled regions of a network. Typically the MDT begins and ends in non-MPLS regions, but travels through an MPLS region. In such cases, it can be useful to begin building the MDT as a pure IP MDT, then convert it to an MPLS Multipoint LSP (Label Switched Path) when it enters an MPLS-enabled region, and then convert it back to a pure IP MDT when it enters a non-MPLS-enabled region. Other documents specify the procedures for building such a hybrid MDT, using Protocol Independent Multicast (PIM) in the non-MPLS region of the network, and using Multipoint Extensions to Label Distribution Protocol (mLDP) in the MPLS region. This document extends those procedures to handle the case where the link connecting the two regions is a "Virtual Routing and Forwarding Table" (VRF) link, as defined in the "BGP IP/MPLS VPN" specifications. However, this document is primarily aimed at particular use cases where VRFs are used to support multicast applications other than Multicast VPN.

Status of this Memo

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

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 20, 2013.

#### Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	4
1.1. Conventions used in this document . . . . .	5
1.2. Terminology . . . . .	5
2. VRF In-band signaling for MP LSPs . . . . .	6
3. Encoding the Opaque Value of an LDP MP FEC . . . . .	7
3.1. Transit VPNv4 Source TLV . . . . .	7
3.2. Transit VPNv6 Source TLV . . . . .	8
3.3. Transit VPNv4 bidir TLV . . . . .	9
3.4. Transit VPNv6 bidir TLV . . . . .	10
4. Security Considerations . . . . .	11
5. IANA considerations . . . . .	11
6. Acknowledgments . . . . .	11
7. References . . . . .	11
7.1. Normative References . . . . .	11
7.2. Informative References . . . . .	12
Authors' Addresses . . . . .	12



## 1. Introduction

Sometimes an IP multicast distribution tree (MDT) traverses both MPLS-enabled and non-MPLS-enabled regions of a network. Typically the MDT begins and ends in non-MPLS regions, but travels through an MPLS region. In such cases, it can be useful to begin building the MDT as a pure IP MDT, then convert it to an MPLS Multipoint LSP (Label Switched Path) when it enters an MPLS-enabled region, and then convert it back to a pure IP MDT when it enters a non-MPLS-enabled region. Other documents specify the procedures for building such a hybrid MDT, using Protocol Independent Multicast (PIM) in the non-MPLS region of the network, and using Multipoint Extensions to Label Distribution Protocol (mLDP) in the MPLS region. This document extends those procedures to handle the case where the link connecting the two regions is a "Virtual Routing and Forwarding Table" (VRF) link, as defined in the "BGP IP/MPLS VPN" specifications. However, this document is primarily aimed at particular use cases where VRFs are used to support multicast applications other than Multicast VPN.

In PIM, a tree is identified by a source address (or in the case of bidirectional trees [RFC5015], a rendezvous point address or "RPA") and a group address. The tree is built from the leaves up, by sending PIM control messages in the direction of the source address or the RPA. In mLDP, a tree is identified by a root address and an "opaque value", and is built by sending mLDP control messages in the direction of the root. The procedures of [I-D.ietf-mpls-mldp-in-band-signaling] explain how to convert a PIM <source address or RPA, group address> pair into an mLDP <root node, opaque value> pair, and how to convert the mLDP <root node, opaque value> pair back into the original PIM <source address or RPA, group address> pair.

However, those procedures assume that the routers doing the PIM/mLDP conversion have routes to the source address or RPA in their global routing tables. Thus the procedures cannot be applied exactly as specified when the interfaces connecting the non-MPLS-enabled region to the MPLS-enabled region are interfaces that belong to a VRF as described in [RFC4364]. This specification extends the procedures of [I-D.ietf-mpls-mldp-in-band-signaling] so that they may be applied in the VRF context.

As in [I-D.ietf-mpls-mldp-in-band-signaling], the scope of this document is limited to the case where the multicast content is distributed in the non-MPLS-enabled regions via PIM-created Source-Specific or Bidirectional trees. Bidirectional trees are always mapped onto Multipoint-to-Multipoint LSPs, and source-specific trees are always mapped onto Point-to-Multipoint LSPs.

Note that the procedures described herein do not support non-bidirectional PIM ASM groups, do not support the use of multicast trees other than mLDP multipoint LSPs in the core, and do not provide the capability to aggregate multiple PIM trees onto a single multipoint LSP. If any of those features are needed, they can be provided by the procedures of [RFC6513] and [RFC6514]. However, there are cases where multicast services are offered through VRF interfaces, and where mLDP is used in the core, but where aggregation is not desired. For example, some service providers offer multicast content to their customers, but have chosen to use VRFs to isolate the various customers and services. This is a simpler scenario than one in which the customers provide their own multicast content, out of the control of the service provider, and can be handled with a simpler solution. Also, when PIM trees are mapped one-to-one to multipoint LSPs, it is helpful for troubleshooting purposes to have the PIM source/group addresses encoded into the mLDP FEC element.

In order to use the mLDP in-band signaling procedures for a particular group address in the context of a particular set of VRFs, those VRFs MUST be configured with a range of multicast group addresses for which mLDP in-band signaling is to be enabled. This configuration is per VRF ("Virtual Routing and Forwarding table", defined in [RFC4364]). For those groups, and those groups only, the procedures of this document are used. For other groups the general purpose Multicast VPN procedures MAY be used, although it is more likely this VRF is dedicated to mLDP in-band signaling procedures and all other groups are discarded. The configuration must be present in all PE routers that attach to sites containing senders or receivers for the given set of group addresses. Note, since the provider most likely owns the multicast content and how it is transported across the network is transparent to the end-user, no co-ordination needs to happen between the end-user and the provider.

### 1.1. Conventions used in this document

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].

### 1.2. Terminology

IP multicast tree : An IP multicast distribution tree identified by an source IP address and/or IP multicast destination address, also referred to as (S,G) and (\*,G).

mLDP : Multicast LDP.

In-band signaling : Using the opaque value of a mLDP FEC element to encode the (S,G) or (\*,G) identifying a particular IP multicast tree.

P2MP LSP: An LSP that has one Ingress LSR and one or more Egress LSRs (see [RFC6388]).

MP2MP LSP: An LSP that connects a set of leaf nodes, acting indifferently as ingress or egress (see [RFC6388]).

MP LSP: A multipoint LSP, either a P2MP or an MP2MP LSP.

Ingress LSR: Source of a P2MP LSP, also referred to as root node.

VRF: Virtual Routing and Forwarding table.

## 2. VRF In-band signaling for MP LSPs

Suppose that a PE router, PE1, receives a PIM Join(S,G) message over one of its VRF interfaces. Following the procedure of section 5.1 of [RFC6513], PE1 determines the "upstream RD", the "upstream PE", and the "upstream multicast hop" (UMH) for the source address S. Please note that sections 5.1.1 and 5.1.2 of [RFC6513] are applicable.

In order to transport the multicast tree via a MP LSP using VRF in-band signaling, an mLDP Label Mapping Message is sent by PE1. This message will contain either a P2MP FEC or an MP2MP FEC (see [RFC6388], depending upon whether the PIM tree being transported is a source-specific tree, or a bidirectional tree, respectively. The FEC contains a "root" and an "opaque value".

If the UMH and the upstream PE have the same IP address (i.e., the Upstream PE is the UMH), then the root of the Multipoint FEC is set to the IP address of the Upstream PE. If, in the context of this VPN, (S,G) refers to a source-specific MDT, then the values of S, G, and the upstream RD are encoded into the opaque value. If, in the context of this VPN, G is a bidirectional group address, then S is replaced with the value of the RPA associated with G. The coding details are specified in Section 3. Conceptually, the Multipoint FEC

can be thought of as an ordered pair: <root=Upstream-PE, opaque\_value=<S or RPA ,G ,Upstream-RD>. The mLDP Label Mapping Message is then sent by PE1 on its LDP session to the "next hop" on its path to the upstream PE. The "next hop" is usually the IGP next hop, but see [I-D.ietf-mpls-targeted-mldp] for cases in which the next hop is not the IGP next hop.

If the UMH and the upstream PE do not have the same IP address, the procedures of section 2 of [RFC6512] should be applied. The root node of the multipoint FEC is set to the UMH. The recursive opaque value is then set as follows: the root node is set to the upstream PE, and the opaque value is set to the multipoint FEC described in the previous paragraph. That is, the multipoint FEC can be thought of as the following recursive ordered pair: <root=UMH, opaque\_value=<root=Upstream-PE, opaque\_value =<S or RPA, G, Upstream-RD>>.

The encoding of the multipoint FEC also specifies the "type" of PIM MDT being spliced onto the multipoint LSP. Four types of MDT are defined: IPv4 source-specific tree, IPv6 source-specific tree, IPv4 bidirectional tree, and IPv6 bidirectional tree.

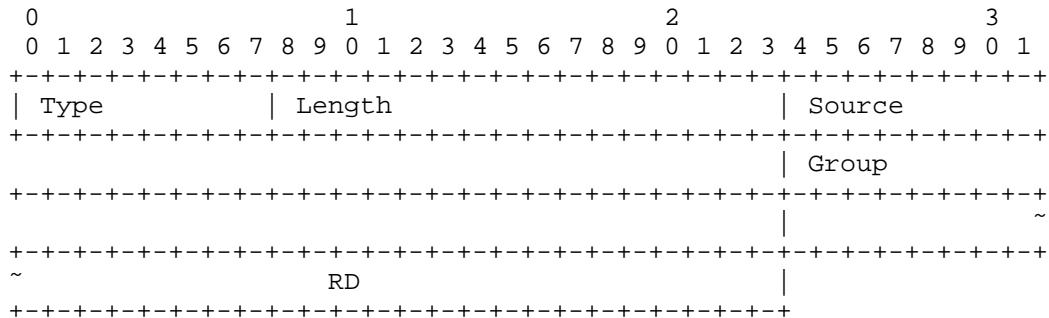
When a PE router receives an mLDP message with a P2MP or MP2MP FEC, where the PE router itself is the root node, and the opaque value is one of the types defined in Section 3, then it uses the RD encoded in the opaque value field to determine the VRF context. (This RD will be associated with one of the PEs VRFs.) Then, in the context of that VRF, the PE follows the procedure specified in section 2 of [I-D.ietf-mpls-mldp-in-band-signaling].

### 3. Encoding the Opaque Value of an LDP MP FEC

This section documents the different transit opaque encodings.

#### 3.1. Transit VPNv4 Source TLV

This opaque value type is used when transporting a source-specific mode multicast tree whose source and group addresses are IPv4 addresses.



Type: (to be assigned by IANA).

Length: 16

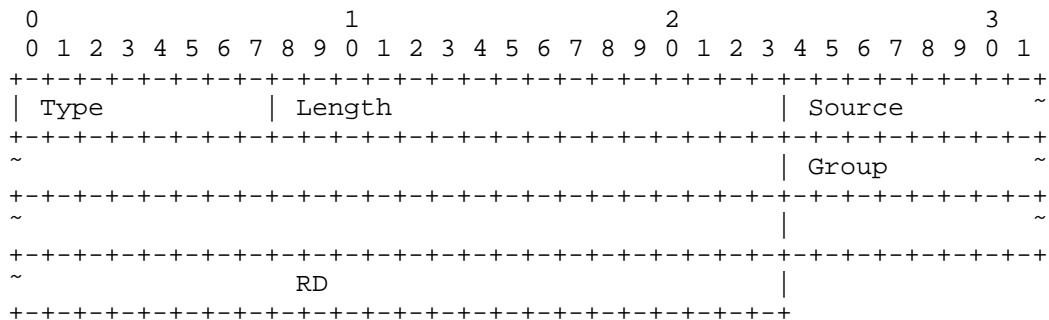
Source: IPv4 multicast source address, 4 octets.

Group: IPv4 multicast group address, 4 octets.

RD: Route Distinguisher, 8 octets.

### 3.2. Transit VPNv6 Source TLV

This opaque value type is used when transporting a source-specific mode multicast tree whose source and group addresses are IPv6 addresses.



Type: (to be assigned by IANA).

Length: 40

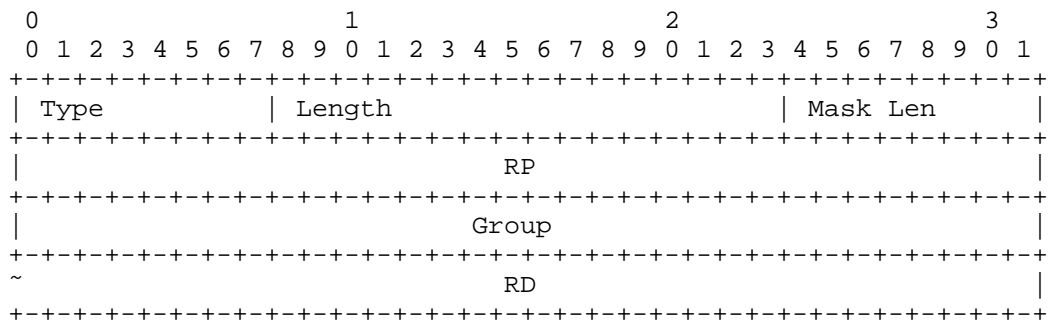
Source: IPv6 multicast source address, 16 octets.

Group: IPv6 multicast group address, 16 octets.

RD: Route Distinguisher, 8 octets.

### 3.3. Transit VPNv4 bidir TLV

This opaque value type is used when transporting a bidirectional multicast tree whose group address is an IPv4 address. The RP address is also an IPv4 address in this case.



Type: (to be assigned by IANA).

Length: 17

Mask Len: The number of contiguous one bits that are left justified and used as a mask, 1 octet.

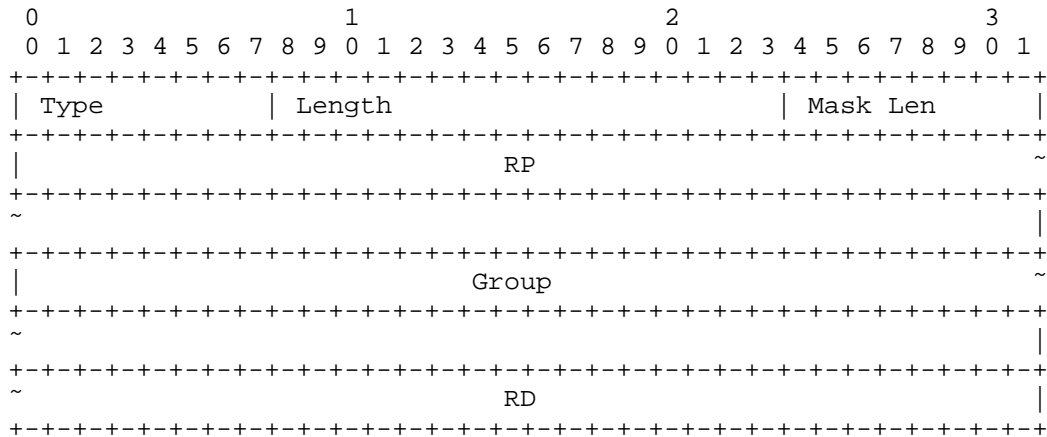
RP: Rendezvous Point (RP) IPv4 address used for encoded Group, 4 octets.

Group: IPv4 multicast group address, 4 octets.

RD: Route Distinguisher, 8 octets.

### 3.4. Transit VPNv6 bidir TLV

This opaque value type is used when transporting a bidirectional multicast tree whose group address is an IPv6 address. The RP address is also an IPv6 address in this case.



Type: (to be assigned by IANA).

Length: 41

Mask Len: The number of contiguous one bits that are left justified and used as a mask, 1 octet.

RP: Rendezvous Point (RP) IPv6 address used for encoded group, 16 octets.

Group: IPv6 multicast group address, 16 octets.

RD: Route Distinguisher, 8 octets.

#### 4. Security Considerations

The same security considerations apply as for the base LDP specification, described in [RFC5036], and the base mLDP specification, described in [RFC6388]

#### 5. IANA considerations

[RFC6388] defines a registry for "The LDP MP Opaque Value Element Basic Type". This document requires the assignment of four new code points in this registry:

Transit VPNv4 Source TLV type

Transit VPNv6 Source TLV type

Transit VPNv4 Bidir TLV type

Transit VPNv6 Bidir TLV type

#### 6. Acknowledgments

Thanks to Eric Rosen, Andy Green and Yakov Rekhter for their comments on the draft.

#### 7. References

##### 7.1. Normative References

- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.
- [RFC5015] Handley, M., Kouvelas, I., Speakman, T., and L. Vicisano, "Bidirectional Protocol Independent Multicast (BIDIR-PIM)", RFC 5015, October 2007.
- [RFC5036] Andersson, L., Minei, I., and B. Thomas, "LDP Specification", RFC 5036, October 2007.



- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC6388] Wijnands, IJ., Minei, I., Kompella, K., and B. Thomas, "Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths", RFC 6388, November 2011.
- [RFC6512] Wijnands, IJ., Rosen, E., Napierala, M., and N. Leymann, "Using Multipoint LDP When the Backbone Has No Route to the Root", RFC 6512, February 2012.
- [I-D.ietf-mpls-mldp-in-band-signaling]  
Wijnands, I., Eckert, T., Leymann, N., and M. Napierala, "Multipoint LDP in-band signaling for Point-to-Multipoint and Multipoint- to-Multipoint Label Switched Paths", draft-ietf-mpls-mldp-in-band-signaling-06 (work in progress), June 2012.

## 7.2. Informative References

- [I-D.ietf-mpls-targeted-mldp]  
Napierala, M. and E. Rosen, "Using LDP Multipoint Extensions on Targeted LDP Sessions", draft-ietf-mpls-targeted-mldp-00 (work in progress), August 2012.
- [RFC6513] Rosen, E. and R. Aggarwal, "Multicast in MPLS/BGP IP VPNs", RFC 6513, February 2012.
- [RFC6514] Aggarwal, R., Rosen, E., Morin, T., and Y. Rekhter, "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", RFC 6514, February 2012.

## Authors' Addresses

IJsbrand Wijnands (editor)  
Cisco Systems  
De kleetlaan 6a  
Diegem 1831  
Belgium

Email: ice@cisco.com

Paul Hitchen  
BT  
BT Adastral Park  
Ipswich IP53RE  
UK

Email: paul.hitchen@bt.com

Nicolai Leymann  
Deutsche Telekom  
Winterfeldtstrasse 21  
Berlin 10781  
Germany

Email: n.leymann@telekom.de

Wim Henderickx  
Alcatel-Lucent  
Copernicuslaan 50  
Antwerp 2018  
Belgium

Email: wim.henderickx@alcatel-lucent.com

Arkadiy Gulko  
Thomson Reuters  
195 Broadway  
New York NY 10007  
USA

Email: arkadiy.gulko@thomsonreuters.com

