Capstone Project Report for Multicast in MPLS based Networks and VPNs

Supervisor: Noonari, Juned

Name: Hamir Riaz

Faculty: Master of Science in Internetworking

Date: April 28, 2015

Abstract

Impetuous development of Internet and IP technologies lead to increasing number of users and burst number of the traffic flows, needed to services demands of these users. Although ISP every few years upgrade their networks with new interfaces: 1G to 10G, 10G to 100G, 400G is near, the exponential growth of requested bandwidth and new services becoming trends for technological answer for implementation of modern means for traffic delivery, specially such as bandwidth hungry video and other over the top applications . Due to special nature of multicast traffic usually it is forwarded in separate way that uncast IP, so we come to concept of isolated traffic infrastructure - MVPN (multicast virtual private network). At ISPs MVPN usually uses overlay model and implemented over IP/MPLS as a service.

The application of MVPN technology is more and more adapted widely these days, the correlated technical standard has been revised for nearly ten years. The purpose of this report is to investigate and compare different MVPN approaches among the most used cases in real networks - Rosen and NG (Next Generation) MVPN. Rosen (long time was in draft state) is PIM protocol based technology with some extensions for tunneling in VPN and NG MVPN, is a technique basing on BGP/MPLS IP VPN supporting Multicast service, that transmit multicast data via private networks by encapsulating private multicast packet and transferring via multicast MPLS tunnels established. Firstly, the report introduces the technical background, then expatiate the basic theory and implementations technical of Multicast in general, then some chapters about MPLS and next focus to MVPN itself. And after it will be the comparison of these techniques of multicast forwarding and trends in this area.

Then the last part is dedicated to real virtual lab environment, considering Rosen scheme.

Keywords: IPv4, IPv6, MPLS, MVPN, BGP, P2MP, PIM, IGMP, Multicast

Table of contents

Introduction

This document describes the different multicast traffic delivery approaches used in service provider networks. To isolate unicast user transit traffic ISPs are using the concept of VPN (Virtual Private Networks).From bird-eye view,the same happens with multicast traffic - and we are coming to the definition of mVPN (multicast VPN), but realization and mechanisms used to forward multicast traffic slightly different. We shortly review the primary definition and understand multicast VPN. Then we dive into the main approaches of mVPN in detail. The objective of this document is to provide detailed information about existing Multicast and MVPN standard implementations. Starting with the older application - draft-Rosen to the most recent, referred to as next-generation BGP-based standards (NG MVPN).Case studies with detailed configurations with lab examples follow. The vendor specific information examples (configurations among others) will use cisco IOS syntax..

As engineers, technical managers, and visionaries in building next-generation IP Multicast infrastructures, we are more interested in standards and areas where there is consensus rather than looking at proprietary implementations, so we will refer the IETF standards, RFC and compare them against each other[27].

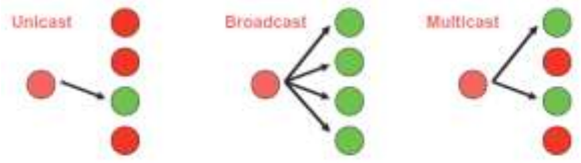
The extent of multicast traffic has been increasing essentially based on the rise of video-based applications. More Layer 3 VPN customers use IP multicast traffic. (Layer 3 VPNs commonly known as 2547 VPNs, that evolved from the original RFC 2547.) Service providers who utilize a base of Layer 3 VPN for unicast service, aim to upsell by incorporating new media-rich solution packages to develop ARPU while achieving operational capability. The meaningful investment in IPTV services and wholesale business models are compelling the need to examine more quantifiable methods to produce multicast services. Likewise, the distribution of multicast-dependent financial services requires quantifiable and substantial MVPN foundations. Additionally, Layer 3 MPLS VPN services, service providers, and cable operators are starting to offer VPLS (virtual private LAN service) to medium and small enterprises, as well as to larger companies. Marketing towards IPv6-enabled services based on new mobile applications is also notable. This multifariousness and extent of services bring forth a challenge to operators to create an infrastructure or foundation that maintains multicast traffic, unicast, Layer 2 VPNs, Layer 3 VPNs, IPv4, and IPv6. The complexity is high for virtual services that require intricate administration and carrier plane operations. Furthermore, the challenge to support emerging multicast applications incrementally and systematically to improve infrastructures of Layer 3 VPN and VPLS without additional operating complexity[26].

Part1.Base Definitions

1.1 IP Multicast

IP Multicast is part of the TCP/IP suite of protocols. There are three modes of communication: Unicast, Broadcast, and Multicast.

Figure 1.1.1 [27]



Routers within the network replicate IP Multicast packets when there are a few subnetworks requiring the same data. IP Unicast provides the root that creates a special IP stream for each receiver. With multicast, only one copy of the packet needs to traverse a link and because of this distributed replication of data it is a robust and quantifiable solution for group communication[30].

Multicast is an efficient model for broadcasting duplicate data to multiple receivers, because of its concord of a Group address which allows a group of receivers to listen to one address. Inside a multicast network, the routers replicate and distribute multicast content to all hosts listening to a particular multicast group. Routers apply Protocol Independent Multicast (PIM) to create distribution trees for packaged software and multicast content to be transmitted; the result is the most efficient delivery of data to multiple receivers.

In applications that use high-bandwidth, like MPEG video, only a few receivers are needed, and IP Multicast is the best solution because streaming video in any other way would use a lot of the accessible network bandwidth. IP Multicast maintains more free resources when transmitting low-bandwidth applications that involve thousands of receivers. Additionally, IP Multicast is the only non-broadcasting alternative for circumstances that require concurrently sending information to multiple receivers. Low-bandwidth applications could replicate data at the source as an alternative to IP Multicast. However, such a solution can decrease application execution, causes temporary inactivity and unsteady pauses that influences users and applications. Situations like these necessitate costly servers to accomplish the replication and distribution of data, which could result in duplicate transmissions that consume too much network bandwidth. IP Multicast is the only viable option for most high-bandwidth applications for the same issues. Today, many applications take advantage of multicast, as shown in Figure 1. Applications that draw benefit from IP Multicast include[27]:

- Corporate communications
- Consumer television and music channel delivery
- Distance learning (e.g., e-learning) and white-boarding solutions
- IP surveillance systems
- Interactive gaming

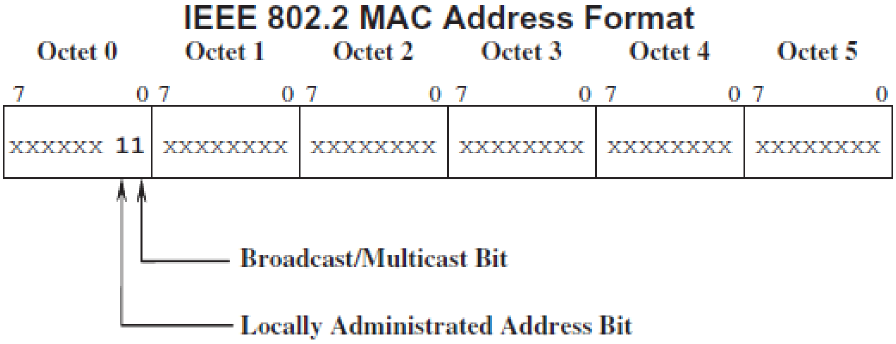
Table1.1.1[27] Summarize some applications for multicast traffic:

	Real Time	**Non-Real Time**
Multimedia	● IPTV ● Live video broadcasting ● Videoconferencing ● Live Internet Audio	● Replication ● Video, Web servers, ● Content delivery
Data-only	● Stock Quotes ● News Feeds ● White-Boarding ● Interactive Gaming	● Information Delivery ● Server to Server ● Server to Desktop ● Database Replication ● Software Distribution

IP Multicast is also supported in[27]

- IPv4 networks
- IPv6 networks
- Multiprotocol Label Switching (MPLS) VPNs
- Mobile and wireless networks

Figure 1.1.2 [27]



IP Multicast capacities are deployed using an array of protocols, and considerations suited to the different network conditions mentioned. Multicast services are also deployed over multiple protocol platforms and domains in the same network. By executing native IP Multicast functionality inside MPLS VPN networks, service providers can deliver bandwidth-intensive streaming services such as online working, video calling, e-learning, and a multitude of other applications more efficiently. Multicast VPN technology reduces the data duplication and performance issues linked with the data transmission related to these applications. Multicast MPLS VPNs also benefits service providers by[27]:

- Minimizing the complexity and configuration time as it is only at the edge routers that the configuration is needed[27]
- Service provider networks are transparent[27]
- Advanced services such as Virtual Multicast Networks are easy to build[27]

- Increasing network scalability[27]

1.2 Multicast addressing

Class D range of IP addresses are used by IP multicast (224.0.0.0 to 239.255.255.255; see Figure 1.5). Within the IP multicast Class D address range, there are a number of addresses reserved by the Internet Assigned Numbers Authority (IANA). These addresses are reserved for well-known multicast protocols and applications, such as routing protocol Hellos, according to IPv4 Multicast Address Space Registry. Examples of these are[27]:

224.0.0.1	All the systems on this particular subnet
224.0.0.2	All the routers on this particular subnet
224.0.0.5	OSPF all routers
224.0.1.1	NTP (Network Time Protocol)
224.0.0.9	RIP-2 (a routing protocol)

Now let's summarize IANA allocations for the Class D range and the recommendations as per RFC 2365:

- The Local Link Scope (224.0.0.0) address was reserved by IANA for use by network protocol. Packets using this range are local in scope and are not forwarded by Multicast routers regardless of the TTL[27].
- The Global Scope (224.0.1.0–238.255.255.255) is reserved for network-wide protocols and commercial Internet multicast applications. These addresses can transit Administrative boundaries and are globally (Internet-wide) unique[27].
- (RFC 2365) Ranges 239.0.0.0/10, 239.64.0.0/10, and 239.128.0.0/10 have not been unassigned and is available for augmentation of the Organizational Local Scope and should be left as such till there is no more space in the 239.192.0.0/14 range to allow for possible future revisions of RFC 2365 for additional scopes larger than organizations (hence the name Greater than Organizational Scope)[27].
- (RFC 2365) The Organizational Local Scope (239.192.0.0/14) is the space from which the Service Provider should allocate groups for the Default and Data MDTs to support multicast VPN. However, they should not be used outside the Service Provider, unless by agreement with interconnected service providers. The multicast groups from these ranges will be used by all PE routers to identify associated MDTs for particular VPNs. The addresses in the Organizational Local Scope can be used across the Service Provider as defined by that service provider[27].
- (RFC 2365) Is the Site-Local Scope (239.255.0.0/16), and the addresses represent applications that are confined to local boundaries. For example, the same 239.255.0.0/16 range could be reused across all precincts in a Service Provider as long as the address was contained within that precinct. RFC 2365 states that the range must not be subdivided for use across sites; therefore all sources in that precinct will be uniquely identified within the range. The groups from this scope must not be advertised across the network into other precincts as each precinct could use the same Site-Local Scope to define local multicast services[27].

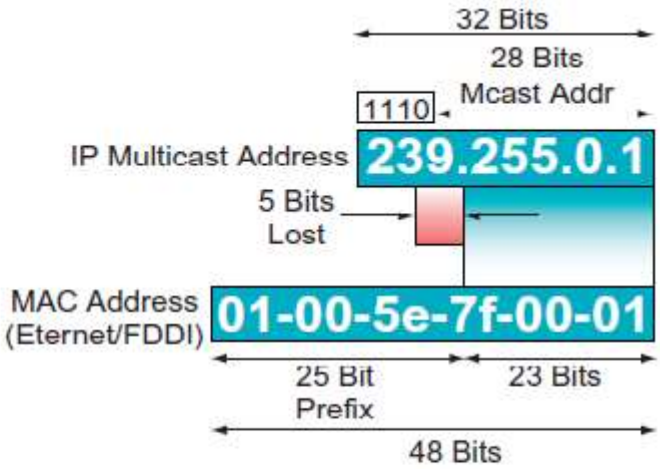
1.3 Layer 2 Multicast Addressing

To support IP multicasting, the address range of 01-00-5E-00-00-00 to 01-00-5E-7F-FF-FF for Ethernet media access control (MAC) addresses was reserved by the IANA. The IEEE 802.2 specification reserves for the transmission of broadcast and/or multicast packets[27].

In an IEEE MAC address Bit 0 of Octet 0 designates if the target address is either a broadcast/multicast address or a unicast address. By setting this Bit, the MAC frame is destined for either a random group of hosts or all the hosts on a particular network (when the MAC target address is the broadcast address, 0xFFFF.FFFF.FFFF). IP Multicasting at Layer 2 makes use of this ability to transmit IP Multicast packets to a group of hosts on an LAN segment. All the IP Multicast frames use MAC layer addresses beginning with the 24-bit prefix of 0x0100.5Exx.But only fifty percent of these MAC addresses are free for use by IP Multicast. This means there are 23 bits of MAC address space for mapping Layer 3 IP Multicast addresses into Layer 2 MAC addresses. Since all Layer 3 IP Multicast addresses have the first 4 of the 32 bits set to 0x1110, this leaves 28 bits of meaningful IP Multicast address information. These 28 bits must map into only 23 bits of the available MAC address. It is not feasible to map all of the 28 bits of the Layer 3 IP Multicast address into the available 23 bits of MAC address space, therefore there are 5 bits of address information lost in the mapping process[27].

This results in a 32:1 address ambiguity when a Layer 3 IP Multicast address is mapped to a Layer 2 IEEE MAC address. It is obvious that this 32:1 address ambiguity has the potential to cause some problems. For example, when a host wishes to permit multicast group 224.1.1.1 they will code the hardware registers in the NIC(network interface card). The aim will be to interpose with the central processing unit once a packet with a target multicast MAC address of 0x0100.5E00.0101 is received. Additionally, the same address is utilized by 31 other IP Multicast groups and if any of these groups are active on the same LAN, the host’s CPU will be interrupted by every frame received from any of these groups. The CPU will have to check the IP of each of the frames received to determine if it is the desired group such as 224.1.1.1, which will negatively impact a host’s CPU power[27]

Figure 1.3.1 [27]



1.4 IGMP protocol

The Internet Group Management Protocol (IGMP) is an industry-standard protocol for managing IPv4 multicast group membership. It is used to dynamically register individual hosts in a multicast group on a particular LAN. Hosts identify group memberships by sending IGMP messages to their local multicast router. Under IGMP, routers listen to IGMP messages and periodically send out queries to discover which groups are active or inactive on a particular subnet. The various IGMP versions are[27]:

- IGMPv1—Provides host mechanisms for joining groups and reporting group membership, as well as a router mechanism for periodically querying for group membership;[27]
- IGMPv2—Provides all of the mechanisms of IGMPv1, as well as a host mechanism for leaving a multicast group, and a router mechanism for sending group-specific membership queries. Today it is the most common used version;[27]
- IGMPv3—Standard for managing multicast group membership, including support for SSM, which allows hosts to join multicast streams on a per-source basis[27].

IGMP reports/joins are the means by which end hosts such as set-top boxes (STBs) request broadcast channels on a particular Multicast address. Either IGMPv2 or IGMPv3 must be supported on IP STBs in an IPTV environment. An IGMPv2 join is a (*, G) join, whereas an IGMPv3 join also includes source information for the multicast group that is being joined [27].

IGMP leave reports are the means by which end hosts such as IP STBs inform the Layer 3 edge device that they are no longer interested in receiving a broadcast channel. When a channel is changed or selected on the STB, it sends an IGMP leave for the channel being watched and sends an IGMP join to the new channel[27].

IGMP snooping fast-leave processing allows IGMP snooping to remove a Layer 2 LAN interface from the forwarding-table entry of a switch without first sending out IGMP group-specific queries. This improves the leave latency, which helps reduce the channel-change time. This feature should be enabled only when there is just one receiver connected behind the Layer 2 interface. This is done because if more than one receiver is connected on this port and if both of them are watching the same channel, then a leave on one STB will cause a leave on the other STB. This feature may therefore be enabled on a residential gateway when there is only one receiver per port[27].

IGMP supports proxy reporting for IGMP messages. In proxy reporting mode, the switch/DSLAM terminates the reports from the STB and forwards only one report for a channel to the upstream router. This feature is also enabled on the residential gateway (RG) in routed mode[27].

IGMPv3 supports explicit tracking of membership information on any port. The explicit-tracking database is used for fast-leave processing for IGMPv3 hosts, proxy reporting, and statistics collection. The main benefit of this feature allows minimal leave latencies when a host leaves a multicast group or channel. A router configured with IGMPv3 and explicit tracking can immediately stop forwarding traffic if the last STB to request to receive a broadcast channel (multicast group) from the router indicates that it no longer wants to receive the channel. The leave latency is thus bound only by the packet transmission latencies in the multi-access network and the processing time in the router. In IGMP Version 2, when a router receives an IGMP leave message from a host, it must first send an IGMP group-specific query to learn if other hosts on the same multi-access network are still requesting to receive traffic. If after a specific time no host replies to the query, the routerwill then stop forwarding the traffic. This query process is required because, in IGMPv2,membership reports are suppressed if the same report has already been sent by another host in the network. Therefore, it is impossible for the router to reliably know how many hosts on a multi-access network are requesting to receive traffic[27].

Static IGMP joins can enable at the network edge to enable the multicast stream to be always avail-able at the respective location. This would help accelerate the channel change time. It is recommended that the Static IGMP configurations are performed only for the most frequently watched channels to ensure optimal bandwidth usage[27].

1.5 PIM (Protocol independent multicast)

The most popular and widely deployed multicast protocol is PIM. Unlike other multicast routing protocols such as Distance Vector Multicast Routing Protocol (DVRMP) or Multicast Open Shortest Path First (MOSPF), PIM does not maintain a separate multicast routing table; instead it relies on the existing Internet Gateway Protocol (IGP) table when performing its Reverse Path Forwarding (RPF) check, one of main multicast concepts.To pass the RPF check, an incoming multicast packet must be received on an interface that the IGP routing table indicates the source (of the multicast packet) is reachable from andIndependent from a multicast routing protocol is where PIM derives its name.[27]

Because PIM does not have to perform multicast routing updates, its overhead is significantly less when compared to other multicast protocols. The main versions of the PIM protocol are in current use, all of which are control plane protocols[27]:

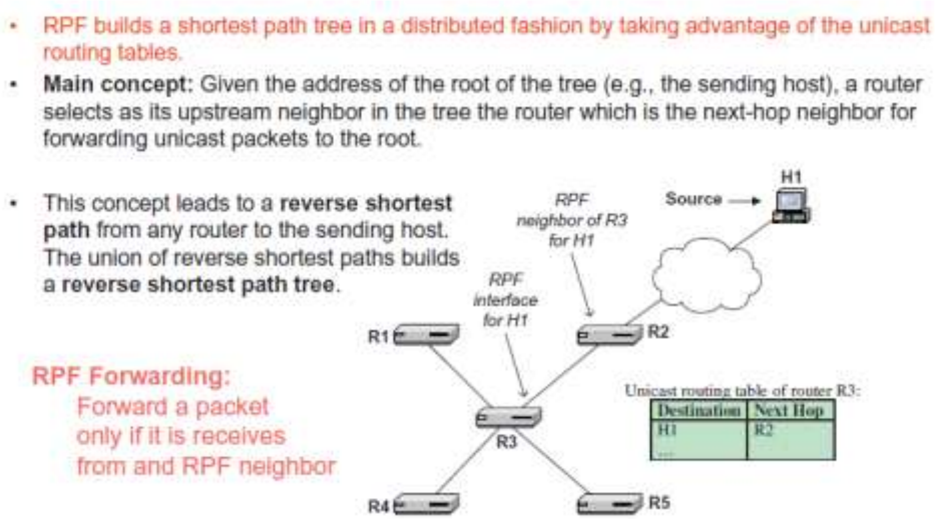
- PIM Dense Mode
- PIM Sparse Mode
- PIM Sparse-Dense Mode
- PIM SSM
- Bidirectional PIM

PIM Dense Mode (PIM-DM) uses a flood and prune mechanism. When a source sends to an IP Multicast group address, each router that receives the packet will create an (S, G) forwarding state entry. The receiving router will initially forward the multicast packet out eligible output interfaces that[27]:

- Pass the RPF check
- Have either IGMP receivers present or PIM neighbors

Note that multicast-enabled interfaces must have the corresponding unicast source routes in the IGP to avoid black holes. In a situation where equal cost paths exist, the unicast route with the highest upstream neighbor IP address is chosen. Also, when there are multiple routers sending on to the same subnet, a PIM assert process is triggered to elect a single designated router (DR) to be the sole forwarder to avoid duplicate frames. When a state is created according to the RPF check, a source tree or shortest path tree (SPT) is developed with the source at the root or first hop router. Multicast packets following the tree take the optimal path through a network and packets are not duplicated over the same subnets. The state created in the routers is referred to as “source comma group” or (S, G), and the routers forwarding interfaces are referred to as an outgoing interface list (OIL). Leaf routers or last hop routers with no receivers then prune back from the tree; however, OILs in the upstream neighbor are maintained. These entries periodically (every 3 minutes) move into a forwarding state and the prune process recurs. PIM-DM is usually not suitable for a large network deployment[27].

*Figure 1.5.1 [27]*The mechanics of RPF check:



- RPF builds a shortest path tree in a distributed fashion by taking advantage of the unicast routing tables.
- **Main concept:** Given the address of the root of the tree (e.g., the sending host), a router selects as its upstream neighbor in the tree the router which is the next-hop neighbor for forwarding unicast packets to the root.
- This concept leads to a **reverse shortest path** from any router to the sending host. The union of reverse shortest paths builds a **reverse shortest path tree**.

RPF Forwarding:
Forward a packet only if it is receives from and RPF neighbor

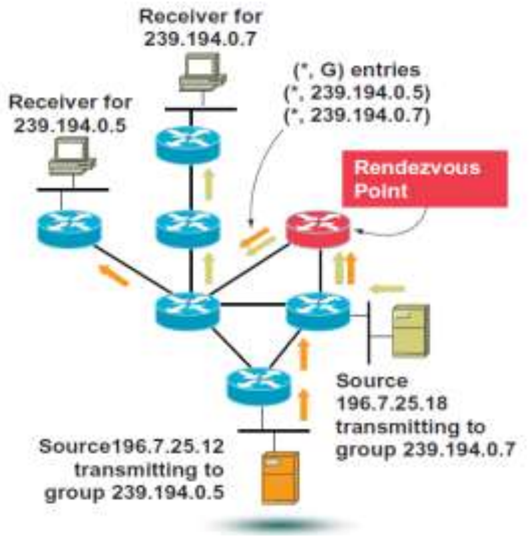
Unicast routing table of router R3:

Destination	Next Hop
H1	R2
...	

PIM Sparse Mode (PIM-SM) uses an explicit join model, where only routers with active receivers will join multicast groups. This has obvious advantages over the flood and prune mechanism deployed in PIM-DM. PIM-SM uses a control point known as Rendezvous Point (RP), which can be viewed as an exchange where receivers and sources can meet. First hop designated routers (the routers with sources attached) register the sources to the RP. When the RP sees the source traffic coming in it will build an SPT back to the source; hence there will be (S, G) state entries between the RP and the source. The last hop designated routers (the routers with the receivers attached) join to the RP hop by hop, creating a shared tree (*, G) with the "*" meaning any source. The RPF check is modified to include the RP for (*, G) entries, whereas the source is used for (S, G) entries. When a source starts transmitting, the initial multicast traffic flows to the RP via an SPT and then down to the receivers for that group via a shared tree (with the RP being the root). This may result in a non-optimal path being created to a receiver depending where the RP is positioned. The operation of a shared tree is shown in Figure 1, and has the following characteristics[27]:

- Root is a common point known as the Rendezvous Point (RP)[27]
- The RP can support many multicast groups[27]
- Receivers join RP to learn of sources[27]
- Sources only transmit to RP[27]
- RP forwards traffic from source to receivers[27]
- Forwarding entries represented as (*, G)[27]
- Less state required at the expense of optimal routing[27]

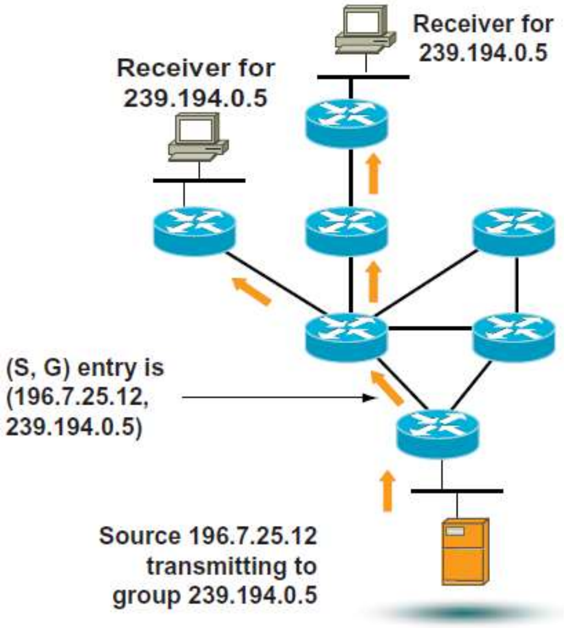
Figure 1.5.2 [27]



In Figure 1.5.2, two receivers are listening for the groups 239.194.0.5 and 239.194.0.7, respectively. The receivers both join the same tree rooted at the source 196.7.25.12. The two groups are associated with different sources: 196.7.25.12 and 196.7.25.18. These sources will transmit their traffic to the RP which will then send the traffic down the shared tree to the receivers. An (*, G) entry in each router along the path represents the distribution tree for each multicast group. To address this problem, a mechanism known as SPT switchover can be used. The last hop router, depending on the traffic rate, sends an (S, G) join toward the source to create an optimal SPT forwarding path, and once established sends RP prunes toward the RP. The decision to create an SPT to the source is dependent upon the SPT-threshold in terms of bandwidth. Most routers have this threshold set to 0; therefore, all traffic received via the RP will be switched to an SPT to the source after the initial flood of packets. This facilitates the creation of a source tree. Source Trees are unidirectional trees rooted at the data source. Traffic flows from the source (at the root) to each receiver (at the leaves) using the shortest possible path. A source tree is also known as a shortest path tree. The operation of a source tree is shown in Figure 1.5.2 and has the following characteristics[27]:

- Simplest form of tree, but receiver requires knowledge of source address [27]
- Traffic travels from source (root) to receivers (leaves) [27]
- Shortest path taken [27]
- Packets replicated at each bifurcation point [27]
- Forwarding entry states represented as (S, G)—(Source IP, Group) [27]
- Provides optimal routing at the expense of more state (S, G) [27]

Figure 1.5.3 [27]



By using PIM-SM, the inefficient flood and prune of PIM-DM is not required. State maintenance is still required in a more controlled manner. To explain this; The RP sends periodic joins toward the source, and last hop routers send periodic joins/prunes toward the RP to maintain group state[27].

PIM Sparse-dense mode. This mode is a combination of both of the previous modes. Depending on whether a group has a matching entry in the Group-to-RP mapping cache, a decision to use sparse or dense mode for a particular multicast group is made. If an entry in the cache exists, then that group in operates in sparse mode on that interface. On the flip side, if there are no analogous entries in the mapping cache, then that group operates in dense mode in that multicast group[27].

PIM SSM (Source-specific multicast).The SSM feature extends IP Multicast where datagram traffic is transmitted to receivers from multicast sources with which the receivers have joined explicitly. When multicast groups are configured for SSM, SSM distribution trees (no shared trees) are created, that is, (S, G) versus (*, G) state. The current IP Multicast infrastructure for the Internet and in many companies and institutions' intranets are based on the PIM-SM protocol and MSDP (Multicast Source Discovery Protocol). These protocols are reliable, extensive, and efficient. Notwithstanding, their ability is bound by the complexity and functionality constraints of the ISM (Internet Standard Multicast) service model. For example, with ISM, the network must maintain knowledge about which hosts in the network are actively sending multicast traffic. With SSM, this information is provided by receivers through the source address(es) relayed to the last hop routers by features such as IGMPv3lite or URL Rendezvous Directory (URD). It is also possible to utilize the SSM-map feature in the network devices, which allows a device to discover the source of a given group automatically, even when the received IGMP message is in the v2 format. SSM is the ideal choice for IPTV deployments. Datagram delivery is based on (S, G) channels in SSM. Traffic for one (S, G) channel consists of datagrams with an IP unicast source address S and the multicast group address G as the IP destination address. Systems will receive this traffic by becoming members of the (S, G) channel. In both SSM and ISM, no signaling is required to become a source. However, in SSM, receivers must subscribe or unsubscribe to (S, G) channels to receive or not receive traffic from specific sources. In other words, receivers can receive traffic only from (S, G) channels that they are subscribed to, whereas in ISM, receivers do not need to know the IP addresses of sources from which they receive their traffic. The proposed standard approach for channel subscription signaling utilizes IGMP INCLUDE mode membership reports, which are only supported in Version 3 of IGMP. The following benefits make SSM the preferred best practice for video-over-IP applications: Internet broadcast services can be provided through SSM without the need for unique IP Multicast addresses. This allows content providers to easily offer their service. IP Multicast address allocation has been a serious problem for Content Providers in the past, and SSM helps provide simpler address management[1][27].

- The large numbers of receivers make broadcast video a common target for attacks. SSM provides greater security and prevents Denial of Service (DoS) attacks due to explicit mapping of sources to Groups (as in the case of SSM with IGMPv3 and SSM-mapping for IGMPv2) [27].
- SSM is relatively easy to implement and maintain (RP configuration not required). Additionally, the control plane for SSM is very simple, and mechanisms such as SPT switchover/threshold are not applicable[27].
- Inter-AS deployments are simpler when using PIM SSM and this assists in forwarding content between multiple independent PIM domains (because there is no need to manage MSDP for SSM between PIM domains)[1][27].

PIM SSM is more modern solution, but as it happens with everything new it must be supported by equipment. Although now most of the modern network equipment has implementation of PIM SSM[27].

A key premise of SSM is that it is the host application program’s responsibility to determine the active source IP address and group multicast address of the desired multicast flow. The host then signals the router via IGMPv3 exactly which specific source (hence the name Source Specific Multicast) and group that it wishes to receive. Since PIM-SSM does not use an RP or a Shared Tree in the 232/8 range, a host will only receive traffic from sources that it has specifically requested. This eliminates interference or DoS attacks from unwanted sources sending to the same multicast group. Furthermore, the lack of Shared Trees in the SSM range means two different sources in the Internet can source traffic to the same group address in the 232/8 range and not have to worry about having a group address conflict. The reason there is no conflict is that the hosts join only the SPT of the desired source. Therefore, one host can receive Stock Quotes from (S1, 232.1.1.1) while another host can be watching live video from (S2, 232.1.1.1), since separate SPTs are being used and no common Shared Tree exists that might accidentally deliver the unwanted source[27].

PIM Bidirectional (RFC 5015) Normally, distribution trees are unidirectional, which means the data flows down the tree toward the receivers and control traffic flows up the tree toward the source[27]. PIM Bi-Dir creates a two-way forwarding tree, which allows the efficient transmission of low bandwidth many-to-many communication (e.g., a financial trading application up and down the tree)[27]. This means that sources and receivers downstream from the RP do not have to transit the RP[27]. But that protocol is rarely used (although RFC 5015 is supported by Cisco and Juniper) and will not be described later in this document. This protocol first was created by Cisco and supported in its network devices.

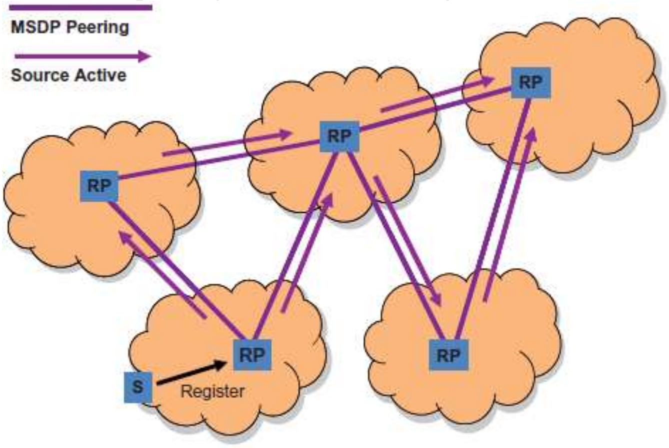
Anycast RP is useful in the application of MSDP (explained in the next section) and can be used within a Service Provider network for scaling purposes. MSDP was incipiently developed for inter-domain multicast applications, and used for Anycast RP, which is an intra-domain feature that provides redundancy and load-sharing capabilities for RPs[27]. Service providers and Enterprise customers mostly use Anycast RP for configuring a PIM-SM network to adhere to fault tolerance specifications within a single multicast domain[27]. The following section includes a brief explanation of MSDP.

1.6MSDP (Multicast Source Discovery Protocol)

Multicast Source Distribution Protocol (MSDP) allows RPs to share information about active sources about the receivers in their local domain. When RPs in remote domains receives data from active sources, they can transmit that data to receivers local to them and facilitates multicast data between domains. A valuable feature of MSDP is that it permits each domain to keep an autonomous RP un-reliant on other domains, but enables RPs to send traffic between domains. PIM-SM is used to transmit data between the multicast domains. The RP in each domain establishes an MSDP peering session using a TCP connection with the RPs in other domains or with border routers leading to the other domains. When the RP learns about a new multicast source within its own domain (through the normal PIM register mechanism), the RP encapsulates the first data packet in a Source-Active (SA) message and sends the SA to all MSDP peers. The SA is forwarded by each receiving peer using a modified RPF check, until the SA reaches every MSDP router in the interconnected networks—theoretically the entire multicast Internet. If the receiving MSDP peer is an RP, and the RP has a (*, G) entry for the group in the SA (there is an interested receiver), the RP creates an (S, G) state for the source and joins to the SPT for the source. The encapsulated data is de-capsulated and forwarded down the shared tree of that RP. When the packet is received by the last hop router of the receiver, the last hop router also may join the SPT to the source. The MSDP speaker periodically sends SAs that include all sources within the own domain of the RP[4][27].

Such logic makes MSDP useful for Inter-AS multicast traffic forwarding implementations.

Figure 1.6.1[27] MSDP - peering between domains for multicast sources exchange



In this section we went through the building blocks or basics of IP Multicast.

In the next sections we will shift to other component of mVPN MPLS solution and further will describe mVPN technology itself.

Part 2. Multicast VPN

MVPN technology is used to deploy multicast services in existing VPNs or as part of any other infrastructure. Typically, multicast data is forwarded between private networks over a VPN infrastructure by encapsulating the initial multicast packets[26].

Layer 3 BGP-MPLS VPNs are widely arrayed in today's networks. RFC 4364, which supersedes RFC 2547, describes protocols and procedures for building BGP-MPLS VPNs for forwarding VPN unicast traffic only. An incremental approach for deploying multicast services can use the same technology as used for deploying Layer 3 VPN for unicast services. This approach can reduce the operational and deployment risk, as well as qualification cost and operating expenditures (OpEx), resulting in a higher ROI (Return of Investment). As multicast applications, such as IPTV and multimedia collaboration, gain popularity, and as the number of networks with different service needs merge over a shared MPLS infrastructure, the demand for a scalable, reliable MVPN service is rapidly increasing[6].

2.1 Basic Implementation Model

For MVPNs, a control plane is required to carry MVPN multicast routing information that is carried from the provider edge (PE) routers. PE routers are connected to the sites that contain receivers to PEs connected to the sites containing the sources. This path allows the receivers to inform the sources that the receiver sites want to receive traffic from the sender sites[6].

As well, a mechanism is needed to carry multicast traffic (data plane). This information is carried from the PE routers connected to the sites that contain the sources to the PEs connected to the sites that contain the receivers. This path enables the flow of multicast traffic from the sources to the receivers[6].

Typically, an MVPN provider network carries multicast routing information from the receivers to the sources and carries multicast data traffic from the sources to the receivers. Different MVPNs may use the same address space (RFC 1918), including an IP multicast addressing space. It would be ideal to use the same route distinguisher mechanism as used by 2547 VPN for unicast to support the overlapping address space[6].

We will not describe the model and internals of BGP IP/MPLS L3VPNs and go directly to multicast VPN implementation. We will use terminology, used in description of MPLS networks.

Draft-Rosen http://www.ietf.org/internet-drafts/draft-rosen-vpn-mcast-xx.txt defined a mechanism to carry multicast traffic within the context of BGP MPLS VPNs and enable the transport of multicast traffic between VRFs of a participating VPN. Now this draft became proposed state as RFC 6037 (xx=15). We will call this Multicast VPN Draft Rosen RFC for the rest of this document. Transporting customer multicast traffic over a shared carrier infrastructure brings its own set of complexities[27]:

- Ensuring that multicast protocols used by the customer are supported by the carrier[27].
- The subscription to the Multicast VPN service should not require any redesign to the customer multicast infrastructure—unless it is mutually agreed—for benefits that are evident to the customer[27].

2.2 mVPN Architecture

Section 2 of Draft Rosen Internet RFC explains the theory of multicast domains in which CE routers control a PIM adjacency with their local PE router only, and not with other CE routers. As previously mentioned, this adjacency quality is indistinguishable to that used in MPLS VPNs. Enterprise clients can keep their existing multicast arrangements, like PIM-SM/PIM DM and any RP discovery tools, and they can upgrade to an mVPN service employing multicast domains without configuration changes. State information is not held by P routers for individual customer source trees, instead hold a single state entry for every VPN (assuming that PIM Bi-Dir is deployed) notwithstanding of the amount of multicast groups in that VPN[31].

When a service provider utilizes PIM -SM in the core (as opposed to PIM Bi-Dir), then the highest volume of state data that a P router would required would be; equivalent to the amount of PE routers in the chine multiplied by the amount of VPNs defined on those PE routers, which should be a lot less than the quantity of likely customer multicast groups. Although you can decrease P-network state data, with multicast domains, regardless of which multicast state the service provider is using (PIM-SM, Bi-Dir, SSM), the volume of state information in the core is deterministic and does not rely on the customer's multicast deployment[31].

Customer networks are also free to use whatever multicast groups they need without the possibility of overlap with other VPNs. These groups are invisible to the P router network, in the same manner that VPN unicast routes are invisible to P routers in an MPLS VPN network[31].

2.3 Multicast Domain Overview

It is necessary to be familiar with these terms as they are essentially the building blocks of this solution[27].

- Multicast domain: A set of BGP/MPLS VPNs (VRFs) that can send multicast traffic to each other, that is, within the context of their respective sites that are part of the same VPN[27].
- Multicast VPN: A VRF that supports both unicast and multicast forwarding tables [27].
- Multicast Distribution Tree: Used to carry customer multicast traffic (C-MCAST packets) among PE routers in a common Multicast Domain[27].

A multicast domain is a set of multicast-enabled virtual routing and forwarding instances (VRFs) that can send multicast traffic to each other. These multicast VRFs are referred to as mVRFs. Multicast domains map all of a customer's multicast groups that exist in a particular VPN to a single unique global multicast group in the P-network. This is achieved by encapsulating the original customer multicast packets within a provider packet by using GRE.

The destination address of the GRE packet is the unique multicast group that the service provider has allocated for that multicast domain. The source address of the GRE packet is the BGP peering address of the originating PE router. A different global multicast group address is required for every multicast domain. Therefore, the set of all customer multicast states (*, G1)_(*, GN) can be mapped to a single (S, G) or (*, G) in the service provider network[31].

Some of the important aspects of the Multicast Domain operation within the context of a Multicast VPN are detailed in the following[27]:

- A MVPN is assigned to a Multicast Domain [27].
- A P-Group address is defined per Multicast Domain, and this address needs to be unique[27].
- C-PACKETS are encapsulated on the PE routers connected to the customer sites and sent on the MDT as P-packets. This ensures that the carrier network does not need to possess any knowledge of customer multicast routing information[27].
- The source address of the P-packet is always the address of the MP-BGP source[27].
- Destination Address is the P-Group address (this address is assigned during the configuration of the MVPN and is also known as the VPN-Group-Address)[27].
- The encapsulation is typically GRE[27].

NOTE

Using Generic Routing Encapsulation (GRE) in a multicast domain is not the same as the network virtualization overlay solution where CE routers are used between point-to-point generic routing encapsulation. These generic routing encapsulation tunnels link PE routers in a multicast configuration. It would be correct to consider tunnels to be point-to-multipoint connections if PIM Sparse Mode is used or even multipoint-to-multipoint if deploying PIM Bi-Dir. Therefore, the use of generic routing encapsulation for multicast domains is inherently more efficient than generic routing encapsulation overlay[31].

Every PE router that supports an mVPN client forms a section of the multicast domain for that particular customer. Multiple end customers can append to a particular PE router, meaning that the PE router can be a member of many multicast domains—one per mVPN customer connected to it[31].

One of the primary appeals regarding multicast domain solutions must be the P routers that have no need for software upgrades to permit new multicast features to maintain mVPNs. Native multicast is the only necessity for the core network system to support multicast domains. The benefit of this is that a native multicast is a developed technology in Cisco IOS; accordingly, operational risks are minimized when the service provider network deploys multicast domains[31].

The P-network is employed in building a default multicast distribution tree (Default-MDT), linking PE routers for every multicast domain by utilizing the single multicast group address that the service provider designates. These multicast groups are unique and named multicast distribution tree -Groups. Each mVRF belongs to a default multicast distribution tree. Therefore, the volume of state data that a P router must hold is not the capacity of the number of client multicast groups in the network; but rather the number of VPNs. This significantly decreases the measure of state data claimed in a P router. If the multicast distribution tree is configured as a bi-directional tree, then a single (*, G) multicast state entry for each VPN is possible[31].

A P router only communicates with the PE router source addresses and the multicast distribution trees-Group addresses that form the multicast distribution trees. CE router traffic traveling along a multicast distribution tree is forwarded in a GRE-encapsulated packet (P-packet) using the multicast distribution tree-group address as the target (more on this later in the "MDTs - multicast distribution tree" section). The GRE P-packet only uses IP, while no MPLS label headers are applied to multicast distribution trees traffic. In the core, there is only pure IP multicast[31].

Multicast Distribution Tree (MDT) is used to carry the customer multicast traffic in a distinct manner—over a shared carrier infrastructure. In simple terms, an multicast distribution tree is a unique multicast tree that is created per MVPN. The two most commonly deployed models are Shared Trees using PIM-SM, also referred to as the Any Source Multicast (ASM) model, and Source Trees using PIM-SSM. A multicast distribution tree for a given customer can use the ASM, SSM, or a combination of both the models. This is predominantly a design-related decision taken by the carrier based on various factors that are beneficial to both the carrier and in certain cases to the customer as well. Most of the questions that arise when reading this section will be addressed as we progress through the chapter[27].

The Multicast distribution tunnels consist of three components:

- Default multicast distribution trees—An MVPN uses this multicast distribution trees to broadcast low-bandwidth multicast traffic or traffic intended for a wide distribution set of receivers and is always used to transmit multicast control traffic between linked PE routers in a multicast domain and is created for every MVPN[27].
- Data multicast distribution tunnels—this type of multicast distribution tunnels is used to tunnel high-bandwidth source traffic through the P-network to interested PE routers. They bypass redundant flooding of client multicast traffic to all of the PE routers within a multicast domain[27].
- Multicast Tunnel Interface (MTI)—It is a representation of access to the multicast domain[27].

Here're the goals of multicast domain solutions:

- Delivering Enterprise Multicast to clients that subscribe to existing MPLS VPN services[31].
- Reducing to a minimum the amount of state information in the service provider core and rendering optimal routing[31].
- Allow clients to choose their own multicast groups, multicast operations mode, RP placement, etc[31].
- Allows multicast in the P-network to be isolated from the operation of multicast in the network[31].

The components used to deploy multicast domains are explained next.

2.4 Multicast Distribution Trees (MDTs) - default MDT

When a Multicast VPN is created, it must also be associated with a Default-MDT. The PE router always builds a Default MDT to peer PE routers that have MVPNs with the same configured MDT-Group address, which is also known as the VPN group address. (Note: Terms will be used interchangeably within this chapter. For instance, MDT group address and VPN group address identify the Default MDT)[27].

Every MVPN is connected to a Default MDT. An MDT is constructed and maintained in the P-network by using standard PIM mechanisms. Consider that a PIM-SM (ASM Mode) was used in the P-network. PE routers in that specific multicast domain will identify each other by connecting with a shared tree for the MDT-Group rooted at the Rendezvous Point (RP) of the service provider, when using PIM SM (ASM model). After a Default MDT is configured for the MVPN, an MTI is created within the specific MVPN, which provides access to the configured MDT-Group. In a case of other PE routers in the network being configured within the very group, then a Default MDT is built between those PE routers. To enable multicast on a VRF holds no guarantee that there is some multicast action on a CE router interface, only that there might exist sources and receivers. In a case that multicast has been allowed on a VRF, and the Default MDT was configured, the PE router joins the Default MDT for that domain unmindful if sources or receivers are active. This facilitates PE routers to build PIM links with PE routers within the same domain; this will at the very least allow MVPN control information to be exchanged[27].

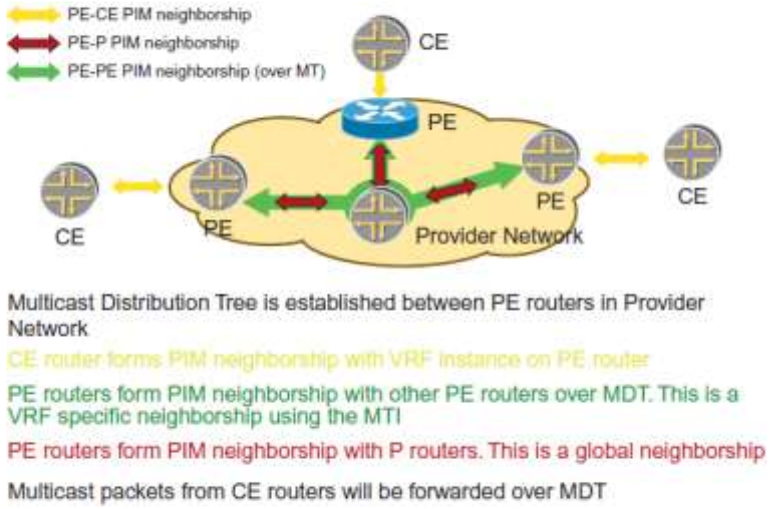
A PE router needs to join an MDT to become the root of that particular tree and the remote peer PE routers and each become a leaf of the MDT. Contrarily, the local PE routers become the leaves of the MDT rooted at remote PE routers. Because the PE router is the root and a leaf of the same tree, it allows participation in a multicast domain as both a sender and receiver[27].

For example, when we have 17 PE routers and each of these PE routers hosts three MVPNs. Because the Default MDT is created on a per-PE and per-MVPN basis—there are a total of 17 (S, G) states per VPN that need to be maintained in the carrier core—each PE can be a sender site and receiver site at the same time. This multiplied by 3 (number of MVPNs) makes it 51 in total. This number will greatly increase in proportion to the number of MVPNs deployed in the network[27].

Before we move to the next stage, it is worthwhile mentioning that the CE router within the MVPN would still use PIM on its interface toward the PE router for exchanging its multicast routing information. Now this information needs to be propagated to the remote CE routers at each site of the same customer or MVPN. This information is exchanged between PE routers via the MTI and using GRE encapsulation as discussed earlier. The MTI ensures that the customer multicast routing information is masked from the provider core (P routers). The VPN group address used for the default MDT encapsulates all C-MCAST traffic. Therefore, the provider network associates all C-MCAST traffic (irrespective of the number of customer sources/groups) using only this single VPN group address. Consequently, all PE routers establish PIM adjacencies with the provider routers inside the carrier core, which is completely separate from the PE-CE and PE-PE (MTI) PIM adjacencies[27].

Figure 2.4.1 explains this in more detail.

Figure 2.4.1 [27].



Multicast Distribution Tree is established between PE routers in Provider Network

CE router forms PIM neighborship with VRF instance on PE router

PE routers form PIM neighborship with other PE routers over MDT. This is a VRF specific neighborship using the MTI

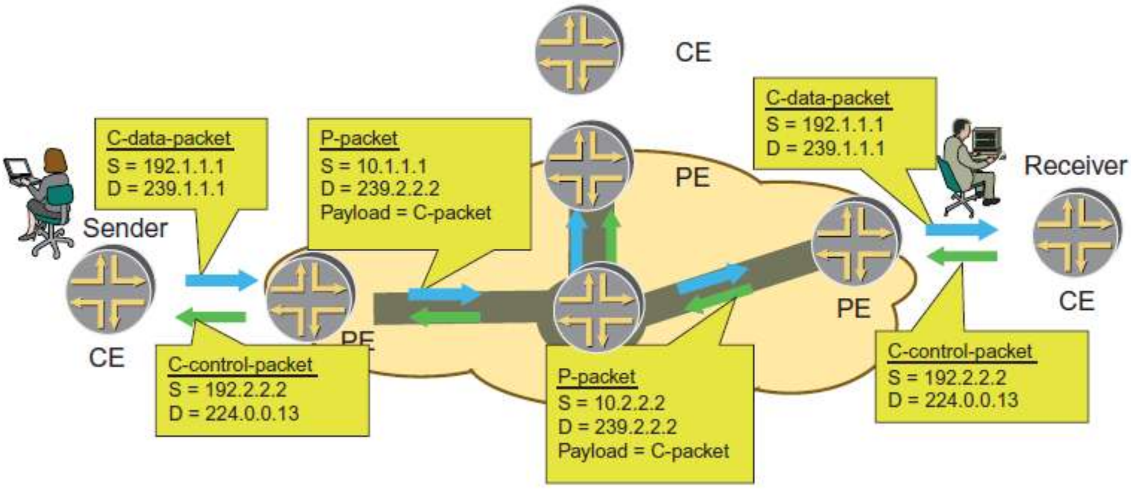
PE routers form PIM neighborship with P routers. This is a global neighborship

Multicast packets from CE routers will be forwarded over MDT

Examine the actual traffic flow in a Multicast VPN environment better to understand the concepts explained so far. In our example, there is a C-MCAST (Customer Multicast) source with an IPv4 address “192.1.1.1” sending traffic to a C-MCAST group at address “239.1.1.1.” (Figure 2.4.2). The VPN Group address assigned for this MVPN in the provider network is “239.2.2.2.” Hence, when looking at Figure 2.4.2, we see that the customer multicast packet is encapsulated using GRE encapsulation, into a P-packet and the source address is the ingress PE (connected to the C-MCAST source) and the multicast group address is the VPN group address[27].

As previously stated, when a PE router sends a client multicast packet onto a multicast distribution tree, it is encapsulated with Generic Routing Encapsulation. This enables the multicast group of a particular VPN to be mapped to a single multicast distribution tree group in the P-network. The PE Multiprotocol BGP local peering address is defined by the source address of the outer IP header, and the destination address is the multicast distribution tree group address allocated to the multicast domain. Therefore, the P-network is only involved with the IP addresses in the Generic Routing Encapsulation header (allocated by the service provider, PE addresses), not the customer-specific addressing[27].

Figure 2.4.2 [27]. Example of default MDT traffic flow



Both customer control and data traffic are sent over the multicast tunnel

P routers only see P packets, so they won't build state for traffic inside the VPN

P packets will go to each PE router that is in the multicast domain

The data packet then gets transmitted via the P-network utilizing the MDT-Group multicast address in the same manner like any other multicast packet. Normal RPF checks are done on the source address, (the originating PE). Once the data package reports at a PE router from a multicast distribution tree, the encapsulation is extracted and the original client multicast packet is forwarded to the corresponding MVPN. The target MVPN is obtained from the MDT-Group address in the target of the encapsulation header. Consequently, using this process causes customer multicast packets to be routed to the P-network to the relevant MDT leaves. Each MDT is a net or mesh of multicast tunnels forming the multicast domain[27].

Figure 2.4.2 provides two traffic flows between the two sites within the MVPN. The first flow is the multicast traffic, which travels between the sender to the receiver (and indicated as C-data-packet for the ease of reading). The egress PE removes the P-packet in order to forward the multicast traffic into the customer domain. The second flow is control traffic, which is encapsulated into a P-packet in the carrier core. The second flow is a PIM Hello message sent from the CE to discover neighboring PIM routers to 224.0.0.13 (ALL-PIM-ROUTERS-group)[27].

The MDT on all PE routers is connected in a fully meshed manner. This infers that the traffic flowing on the Default MDT inevitably reaches all the participating PE routers. When CE starts sending traffic, PEs sends the traffic over the Default MDT. Therefore, the traffic reaches every other PE router that is part of this MDT. This can be very suboptimal and is a serious limitation, especially if there are large volumes of traffic and few interested receivers. This model can result in potentially large volumes of bandwidth being utilized since PE routers without interested receivers would still be receiving traffic only to be dropped. So a better way is to have a more optimal way to have traffic delivered only to interested receivers, yet over a shared tree; in other words, to emulate an SSM model without the cost of much state information injected into the network. Remember that an SSM model requires unique multicast groups to distinguish between traffic flows, whereas we only use a single VPN group address within the context of the Default MDT. So we are going to concept, which is intended to alleviate the situation with excessive traffic - Data MDT[27].

2.5 Multicast Distribution Trees (MDTs) - Data MDT

To surmount the constraints of the Default MDT, a special MDT group called a Data MDT can be created within the MVPN to decrease the flooding by transmitting data only to PE routers that have active VPN receivers. The Data MDT can be created dynamically if a particular customer multicast stream exceeds a bandwidth threshold. Each MVPN can have a pool (group range) of Data MDT groups allocated to it. Note that the Data MDT is only created for data traffic. Default MDT hosts all multicast control traffic to assure that all PE routers receive control information[27].

As soon as traffic limit is exceeded on the Default MDT, the PE router linked with the multicast traffic through the VPN source and can change the (S, G) from the Default MDT to a group associated with the Data MDT. The rate the threshold is checked is a fixed value, which is user configurable. The PE router checks the bandwidth threshold on the basis of (S, G) customer multicast streams rather than all of the traffic on the Default MDT. The group selected for the Data MDT is then taken from a pool that has been configured on the MVPN. For all origins that exceed the configured bandwidth limit, a new Data MDT is created from the available pool for that MVPN. If there are more high-bandwidth sources in the pool than groups available, then the group that is referenced the least is selected and reused. This entails that when the pool includes few groups, the Data MDT might be used by more than one high-bandwidth source. Smaller Data MDT pools ensure that the volume of state data in the P-network is reduced. A large Data MDT means it is less likely for sources to share the same Data MDT resulting in optimal results at the expense of increased state information in the P-network (carrier network)[27].

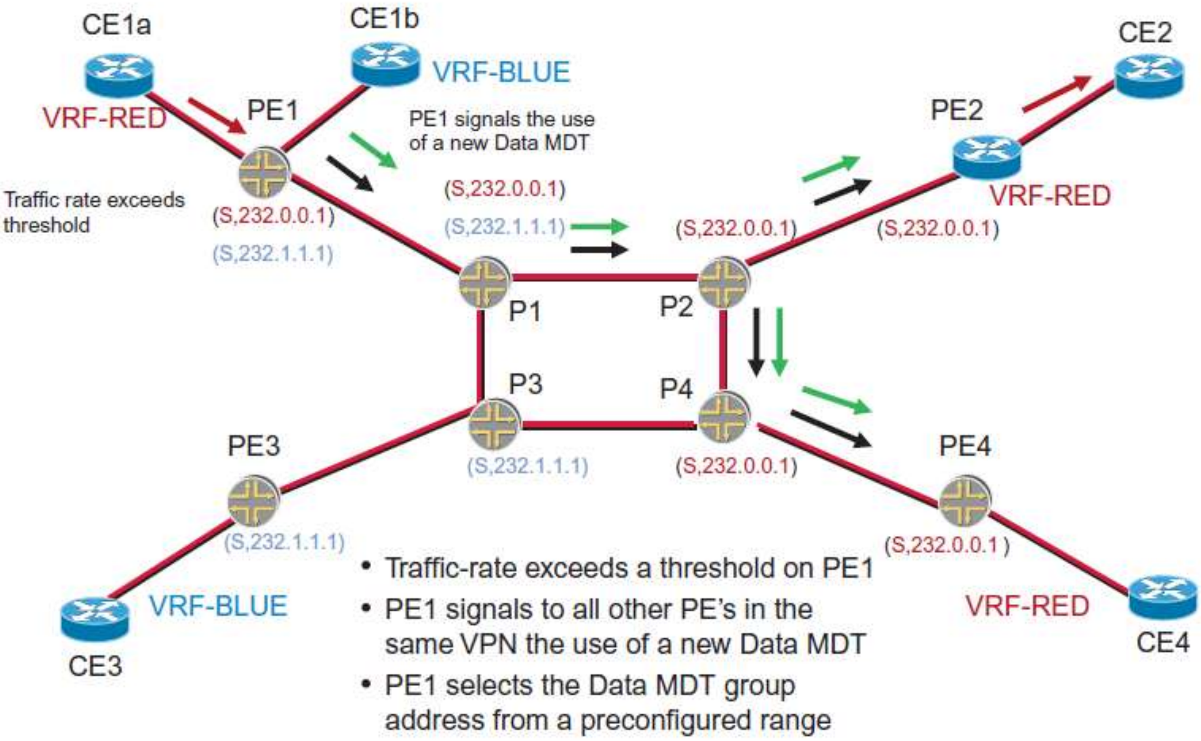
The multicast source traffic is encapsulated in an identical way as the Default MDT when a PE router creates a Data MDT, but the destination group is taken from the Data MDT pool. Any PE router that has affected receivers needs to issue a P-join for the Data MDT; else, the receivers cannot identify the C-packets as they are no longer active on the Default MDT. For this to happen, the origin PE router must notify every other PE router in the multicast domain of the presence of the newly created Data MDT. This can be achieved by sending a special PIM-like control message on the Default MDT containing the customer's (S, G) to Data MDT group mapping. This message is called a Data MDT join. It is an invitation to peer PE routers to join the new Data MDT if they have interested receivers in the corresponding MVPN. A UDP packet conveys the message destined for the ALL-PIM-ROUTERS group (224.0.0.13) with UDP port number 3232. The (S, G, Data MDT) mapping is advertised by using the type, length, value (TLV) format[27].

Any PE router that receives the (S, G, Data MDT) mapping joins the Data MDT if they have receivers in the MVPN for G. The source PE router that initiated the Data MDT waits several seconds before sending the multicast stream onto the Data MDT. The delay allows time for receiving PE routers to establish a path back to the Data MDT root and avoid packet loss when switching from the Default MDT. The Data MDT is a transient object that exists as long as the bandwidth limit is exceeded. If the traffic bandwidth drops below the threshold, the source is switched back to the Default MDT. To circumvent changes among the MDTs, traffic only reverts to the Default MDT if the Data MDT is at least one minute old. PE routers that do not have MVPN receivers for the Data MDT will cache the (S, G, Data MDT) mappings in an internal table so that the join latency can be minimized if a receiver appears at some given point in time. Figures 2.5.1, 2.5.2 and 2.5.3 are examples of the Data MDT setup[27].

In the previous example, MVPN RED has a Default MDT on group address “232.0.0.1,” and MVPN BLUE has a group address of 232.1.1.1. This state information is maintained on all of the routers in the carrier network. Now traffic from a given C-MCAST source exceeds the threshold configured. (Remember, it was mentioned that the Data MDT can be initiated based on a given traffic threshold of a multicast stream.) PE1 is the source from which PE immediately signals to all of the other PEs that are part of the Default MDT by using Data MDT. The subsequent steps are seen in more detail in Figure 2.5.2[27].

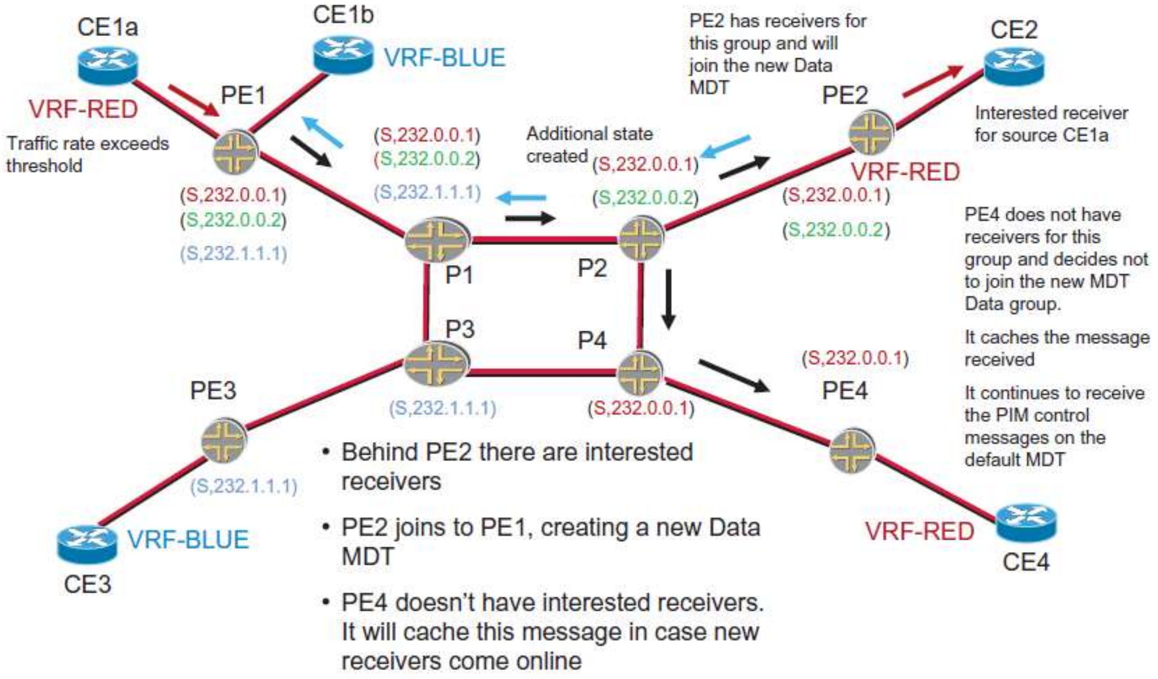
The Data MDT address chosen in this example is “232.0.0.2,” which is signaled to all the PEs. Now only PE2 chooses to join the Data MDT, since it has interested receivers for the group to which the multicast stream is being sent. PE4, on the contrary, does not join the stream and only caches the information, which can be reused in the event of future interests received from its connected CE router. The next set of events is illustrated in Figure 3 where traffic now actually starts flowing on the Data-MDT (Group 232.0.0.2), and only PE2 is actually receiving the traffic[27].

Figure 2.5.1 [27]. Signaling the new multicast distribution path - Data MDT



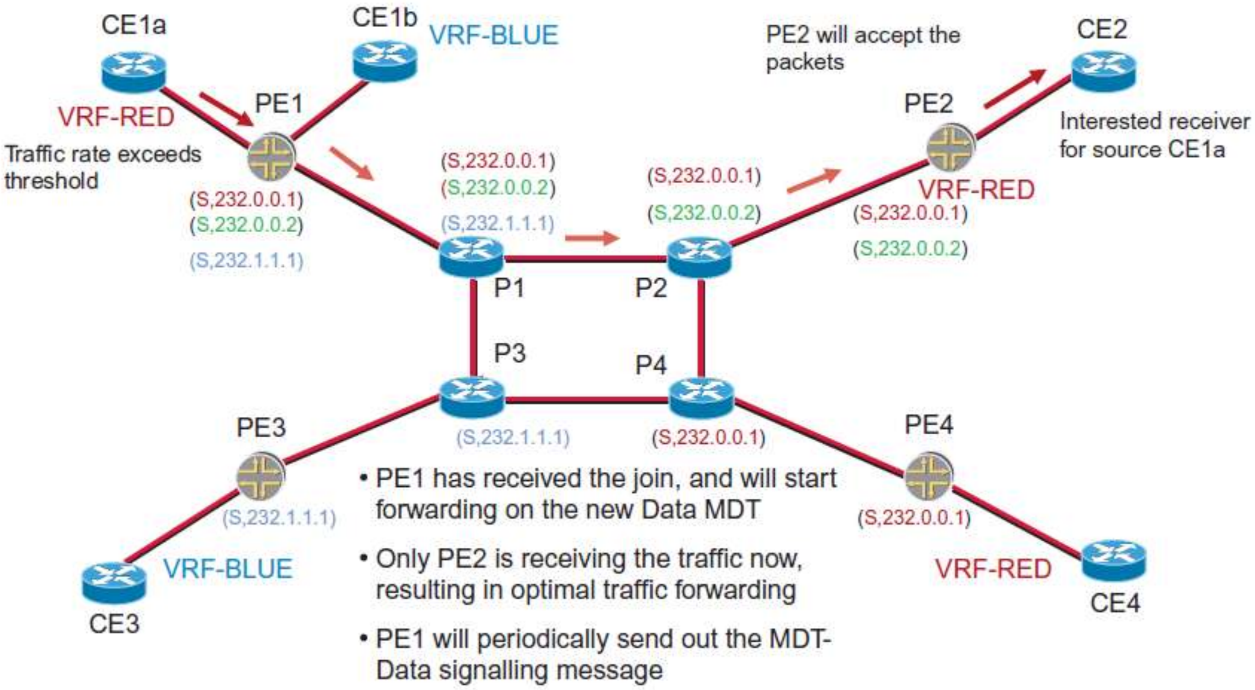
- Traffic-rate exceeds a threshold on PE1
- PE1 signals to all other PE’s in the same VPN the use of a new Data MDT
- PE1 selects the Data MDT group address from a preconfigured range

Figure 2.5.2 [27] Building the Data MDT


CE1a
CE1b
VRF-BLUE
PE2 has receivers for this group and will join the new Data MDT
CE2
VRF-RED
PE1
PE2
Traffic rate exceeds threshold
(S,232.0.0.1)
(S,232.0.0.2)
Additional state created
(S,232.0.0.1)
(S,232.0.0.2)
(S,232.1.1.1)
VRF-RED
Interested receiver for source CE1a
(S,232.0.0.1)
(S,232.0.0.2)
(S,232.1.1.1)
(S,232.0.0.1)
(S,232.0.0.2)
PE4 does not have receivers for this group and decides not to join the new MDT Data group.
It caches the message received
It continues to receive the PIM control messages on the default MDT
P1
P2
P3
P4
PE3
PE4
(S,232.1.1.1)
(S,232.0.0.1)
(S,232.0.0.1)
(S,232.1.1.1)
VRF-BLUE
CE3
VRF-RED
CE4

- Behind PE2 there are interested receivers
- PE2 joins to PE1, creating a new Data MDT
- PE4 doesn't have interested receivers. It will cache this message in case new receivers come online

Figure 2.5.3 [27]. Forwarding starts using Data MDT



- PE1 has received the join, and will start forwarding on the new Data MDT
- Only PE2 is receiving the traffic now, resulting in optimal traffic forwarding
- PE1 will periodically send out the MDT-Data signalling message

2.6 Multicast VRF

Each VRF on a PE router is a vessel to configure multicast routing, the same can be said for a forwarding table also called multicast VRF (mVRF) which is the PE router's view into the enterprise VPN multicast network, and it carries the multicast routing information required to configure that VPN. The relevant data carries the state entries for distribution trees or RP-to-group mappings (for PIM SM). When multicast data or control packets are received by a PE router, multicast routing such as RPF checks and forwarding are performed on the associated mVRF, the same is true for a CE router interface in a VRF[31].

The PE router can configure and set multicast features or protocols in the setting of the mVRF as well. Consider this, when a customer network is using static RP configurations (in other words not using Auto-RP to distribute RP information), the PE router needs to be configured the same as a static RP entry information used in a C-network. The protocols for multicast routing in Cisco IOS such as PIM, IGMP, and MSDP has over time been modified to work in the setting of an mVRF and as such only modify data structures and states within that mVRF[31].

2.7 Multicast Tunnel Interface

The MTI appears in the MVPN as an interface called “Tunnelx” or “MT” depending on the vendor and platform used. For every multicast domain in which an MVPN participates, there is a corresponding MTI. An MTI is a gateway that in essence connects the MVPN to the carrier’s MDT(global environment). Each C-packets transferred to the MTI is encapsulated in a P-packet utilizing GRE and sent along the MDT. Whenever the PE router sends to the MTI, it means that is the root of that MDT. However, a leaf of that MDT means the PE router draws traffic from an MTI. The MT forms PIM adjacencies to all other PE routers in the multicast domain[27].

Therefore, the PIM neighbors of the PE routers are reachable through the same MTI. An instance of MVPN PIM treats the MTI is like an LAN interface: therefore, over MTI, all PIM LAN procedures are valid[27].

The PE router sends PIM control broadcasts over the MTI so that multicast forwarding trees is created between client sites separated by the P-network. The forwarding trees referred to are visible in the C-network, and not on the P-network. To enable multicast transmitting between a client's sites, the MTI forms a part of the outgoing interface list for the (S, G) or (*, G) states that originate from the MVPN. The MTI is generated dynamically upon configuration of the Default-MDT and cannot be explicitly configured. PIM Sparse-Dense (PIM-SD) mode is automatically enabled so that multiple customer group modes can be supported. For example, if the customer was exclusively using PIM-DM, then the MTI would be added to the outgoing list (olist) in the MVPN, but the entry would be labeled Forward/Dense to allow distribution of traffic to other customer sites. If the PE router neighbors all sent a prune message back, and no prune override was received, then the MTI in the olist entry would be set to Prune/Dense exactly as if it were a LAN interface. [13] If the client network is running PIM-SM, an MTI is added to the olist once an explicit join from a remote PE router in the multicast domain is received[27].

The MTI is inaccessible and invisible to the IGP (such as OSPF or ISIS) operating in the client network. So, no unicast routing is transmitted over the MTI because the interface is not listed in the unicast routing table of the associated VRF. Traffic received through an MTI has direct implications on current RPF procedures because RPF check is performed on the unicast routing table for PIM[27].

2.8 RPF Check

RPF is a basic element of multicast routing. The RPF check assures that multicast traffic has correctly arrived from the interface that spans back to the source. Once this check is passed the multicast packets are ready to be distributed away from the origin to the appropriate interfaces. RPF carries the RPF neighbor and the RPF interface.[27]

The RPF interface performs the RPF check by ensuring the interface receives the multicast packet as determined by the unicast routing table as it is supposed to.

The IP address is the RPF neighbor of the PIM adjacency and transmits messages like PIM joins for the (*, G) or (S, G) entries (backwards to the root of the tree where the RP resides). While setting up the control plane of a (*, G) or (S, G) entry, the RPF interface and neighbor are created. When data is forwarded, the RPF check is performed employing the cached RPF interface in the state entry. [27]

We can categorize the RPF check in an mVPN environment into three different kinds of multicast packets;[27]

- A PE router customer interface sends C-packets in the mVRF
- A P router or a PE router interface in the global routing table transmits to P-packets.
- a multicast tunnel interface in the mVRF sends to C-packets

In the first two categories, The RPF checks are executed as per legacy RPF procedures. The interface data gathered from a unicast routing table is cached or stored in a state entry.

The C-source lookup for C-packets returns a PE router interface in the VRF unicast routing table associated with that VRF.[27]

The P-source lookup for P-packets in the global routing table renders back an interface joined to another P router or PE router, and the results of these lookups form the RPF interface.[27]

C-packets that are received from an MTI in the third category is treated somewhat otherwise and demands some alteration to the way the (S, G) or (*, G) states are created. The origin of C-packets in this category is remote PE routers within the network and the travel along the MDT across the P-network. Consequently, the C-packets from the mVRF's perspective must be received on the MTI. Not withstanding, a lookup of the C-source in the VRF does not return the tunnel interface because the MTI does not engage in unicast routing. Rather, the route to the C-source is administered by Multiprotocol BGP as a VPNv4 prefix from the remote PE router – this are normal client prefixes, existing in customer VPN[15]. This means the receiving interface is really in the P-network and when this is true, the RPF procedure has been adjusted to facilitate Multiprotocol BGP that has acquired a C-source address prefix, the RPF interface is set to the MTI that is linked with that mVRF. [27]

NOTE

The altered RPF interface procedure is only applicable to mVRFs in a single multicast domain. Although the Cisco implementation currently limits an mVRF to one domain, the multicast domain architecture can support multiple domains in an mVRF.[27]

The method for defining the reverse path forwarding (RPF) neighbor has also been modified.

If the reverse path forwarding (RPF) interface is set to the MTI, then the reverse path forwarding neighbor must be a remote PE router. (Note that PE router forms PIM adjacencies to other PE routers via the MTI.) The reverse path forwarding neighbor is then determined by examining this criteria; [27]

Firstly, the RPF neighbor needs to be the BGP next-hop to the C-source as recorded in the routing table for that VRF. [27]

Secondly, an identical BGP next-hop address has to be a PIM neighbor in the adjacency table for the mVRF. For this reason, the PIM must use the local BGP peering address when it transmits hello packets over the MDT. Reference to the BGP table is established during the setup in the control plane for creation of the RPF entries and is not repeated. When multicast data is sent, verification is required on the cached RPF information.[27]

2.9 Multiprotocol BGP MDT Updates and SSM

When a Default-MDT group is created by a PE router, all its peers is updated using Multiprotocol BGP, and this process provides two pieces of information: the created MDT-Group as well as the source address of the tree. This is also the BGP peering address of the PE router where the message originated. Presently, this information is only used to maintain P-networks that use SSM. In case the MDT-Group range is enabled for SSM, the source tree is immediately joined. Contrary to PIM SM, as with this, the shared tree is rooted at the RP initially joined[31].

In an MDT-Group range configured to operate in SSM mode on a PE router, the PE router requires information on the MDT root's source address to establish the (S, G) states. This is rendered in the Multiprotocol BGP update with special NLRI type. [31]

The received information is cached in the BGP VPNv4 table For PE routers that do not use SSM. One of the principal benefits is that SSM does not rely on RPs, which eliminates the RP as a single point of failure. BGP update message carries the MDT-Group as an extended community attribute by using the type code of 0x0009[31].

The BGP MP_REACH_NLRI attribute (AFI=1 and SAFI=128) carries the root address of the MDT and uses the same format as a VPN-IPv4 address. It is generally referred to as an mVPN-IPv4 address. However, the NLRI portion of the attribute carries no label information. Information about Data-MDTs isn't transmitted through Multiprotocol BGP messages. The Data-MDT PIM join messages are used for this purpose. Finally let's summarize the information how multicast forwarding occurs[16][31]:

2.10 mVPN Forwarding

We can divide Forwarding into two categories: C-packets received from a PE router client interface in mVRF but excluding MTI, and PE router global multicast interfaces that send and P-packets. In the interest of simplifying things, we can assume that all control checks such as time-to-live (TTL) and RPF are successful, meaning the network is in a converged state[31].

C-Packets Received from a PE Router Customer Multicast Interface

Here is what we need to know about the steps for a router to take when a VRF interface sends a multicast packet to the PE router:

Step 1 A C-Packet arrives on a VRF-configured PE router interface, assuming the source behind it[31].

Step 2 The VRF that is configured for that interface implicitly identifies the mVRF[31].

Step 3 A RPF check is done on the C-packet, and if successful the C-packet is replicated based on the contents of the olist for the (S, G) or (*, G) entry in the mVRF. The olist might contain multicast-enabled interfaces in the same mVRF, in which case packet forwarding follows standard multicast procedures. The olist might also contain a tunnel interface (default or data MDT) that connects the multicast domain[31].

Step 4 If the olist contains a tunnel interface, then the packet is encapsulated by using GRE, with the source being the BGP peering address of the local PE router and the destination being the MDT Group address. The decision on whether the Default-Group or the Data-MDT group is selected depends on whether the y flag is set on the (S, G) entry in the mVRF. The Type-of-Service byte of the C-packet is copied to the P-packet[31].

Step 5 The C-Packet is now a P-Packet in the global multicast routing table. So it can be called multicast in multicast encapsulation[31].

Step 6 The P-packet is forwarded all the way through the P-network by using standard multicast procedures. P routers are unaware of any mVPN activity and treat the packet as native multicast[31].

P-Packets Received from a PE Router Global Multicast Interface

The following describes the steps that the router takes when a multicast packet arrives at the P router from another P router or PE router in the global routing table:

Step 1 A P-packet arrives from a PE router interface in the global network[31].

Step 2 The P-packet's corresponding (S, G) or (*, G) entry is looked up in the global mroute table, and a global RPF check is done[31].

Step 3 If the RPF check is successful, the P-packet is replicated out any P-network interfaces that appear in the olist for its (S, G) or (*, G) entry. At this point, the P-packet is still being treated as native multicast, so packet is encapsulated in GRE, but not uses MPLS labels for forwarding[31].

Step 4 If the (S, G) or (*, G) entry has the Z flag set, then this is a Default- or Data-MDT with an associated mVRF; therefore, the P-packet must be de-encapsulated to reveal the C-packet[31].

Step 5 The destination mVRF of the C-packet is derived from the MDT-group address in the P-packet. The incoming MTI is also resolved from the MDT-group address[31].

Step 6 The C-packet is presented to the target mVRF, with the appropriate MTI set as the incoming interface. The RPF check verifies this tunnel interface[31].

Step 7 The C-packet is once again a native multicast packet, but it resides in the customer's network. The C-packet is replicated to all multicast-enabled interfaces in the mVRF that appears in the olist for the (S, G) or (*, G) entry on this PE[31].

Part 3. Next-generation Multicast VPN (BGP Based NG mVPN, RFC 6513).

3.1 NG MVPN Approach

The NG MVPN RFCs introduced a BGP-based control plane that is modeled after its highly successful counterpart of the VPN unicast control plane. NG MVPNs adopted two important properties of unicast BGP-MPLS VPNs: The BGP protocol is used for distributing all necessary routing information to enable VPN multicast service. This protocol allows service providers to leverage their knowledge and investment in BGP-MPLS VPN unicast services to offer VPN multicast services[26].

The use of BGP for distributing C-multicast routes results in the control traffic exchange being out-of-band from the data plane. This implementation allows for the separation of the control and data plane protocols and makes it easier to leverage newer transport technologies, such as point-to-multipoint (P2MP) MPLS, in delivering MVPN services[26].

The BGP-based NG MVPN control plane lends itself naturally to supporting flexible topologies, such as extranet and hub and spoke, as well as IPv6 support. IPv6 NG MVPN provides the ability to naturally use MPLS encapsulation for IPv6 multicast. It also uses the same model as IPv6 VPN (as defined in RFC 4659) for unicast. Thus, service providers are ensured of a smooth integration of IPv6 multicast services with an existing IPv4 NG MVPN or IPv6 unicast VPN model. BGP MVPNs also provide multihoming support for connecting a multicast source to two PEs, thus enabling sub-second failover from a PE node failure. The autodiscovery of MVPN members available with the BGP approach provides a high degree of automation for establishing provider tunnels that are used for carrying MVPN data among PEs[26].

3.2 BGP/MPLS NG mVPN – carrying traffic across the provider network using inter-PE MPLS tunnels

Just like in the unicast case, the PEs identify traffic arriving from the other PEs as belonging to that particular VPN based on the tunnels over which these packets arrive.9 In the simple case, assuming there is a single sender in the entire customer VPN and assuming that separate distribution trees are used within the service provider for each VPN, the distribution tree itself can be used to identify the VPN. This is similar to the use of the MDT in the PIM/GRE mVPN solution. However, while the draft-rosen approach restricts the MDTs to be PIM signaled GRE-based tunnels, the NG mVPN solution allows for a wide range of tunneling technologies in the provider network. Of these, the most interesting is the use of MPLS P2MP LSPs as tunnels for transporting mVPN traffic between the PEs servicing the source and the PEs servicing the receivers. Using MPLS for the inter-PE tunnels is advantageous for two reasons: (1) the same protocol, namely RSVP-TE or LDP, can be used to establish MPLS tunnels for the purpose of carrying multicast traffic as for unicast traffic, resulting in a reduction in the number of protocols in the service provider network, and (2) when RSVP-TE is used as the signaling protocol, traffic engineering and protection can be achieved for the multicast traffic in the service provider network. It is interesting to note that PHP (as described in Part 1) must not be used when using P2MP MPLS tunnels in this context. This is because a PE must be able to identify traffic arriving at the PE as being associated with a particular VRF on that PE, and the only information that provides such identification is the MPLS label of the LSP that carries the traffic[32].

The use of MPLS as a transport technology gives its name to the BGP/MPLS mVPN solution. However, other options for inter-PE tunnels (including PIM-signaled GRE-based tunnels) are not precluded by the NG mVPN solution. The ability to use different tunneling mechanisms in the provider network brings much-needed flexibility to the service provider, whether it is in the context of a migration scenario, a legacy network or the optimization along a different set of parameters, and also makes this solution consistent with the unicast model of BGP/MPLS VPNs[32].

3.3 NG mVPN – control plane

The NG-MVPN control plane within the Provider network is based on BGP signaling. In other words, BGP is used for exchanging both VPNv4 unicast and multicast information, thus replacing the need for PIM with the Provider IP/MPLS network. The use of a single control plane protocol for invariably all IP/MPLS-based services such as IPv4 Internet prefixes, VPNv4 for both unicast and multicast, and IPv6 results in decreased operational overhead and also offers a simplified and converged control plane infrastructure. In the context of NG-MVPNs, BGP is used for the following functions[27]:

● Auto-discovery of PE routers within a given NG-MVPN instance[27].

● Exchange of Data plane (from this point onward the Data plane will be referred to as Provider- Tunnel or P-tunnel) information between Provider Edge routers. In the context of NG-MVPN, details on the type and identifier of the tunnel used for transmitting C-MCAST traffic is advertised from the ingress PE to all relevant egress PE routers[27].

● Exchange of C-MCAST routing information. All joins from the Customer domain (CE routers) are announced to the relevant PE routers within a context of a given NG-MVPN[27].

The introduction of the BGP control plane does not impose any restrictions in the customer multicast domain. CE routers continue to use PIM between the CE-PE links similar to Draft-Rosen. Therefore the introduction of NG-MVPNs or migration of customers using Draft-Rosen to the NG-MVPN scheme (which will be discussed in detail later in this chapter) does not warrant any redesign or changes to the customer infrastructure. One of the advantages that NG-MVPN offers is a seamless migration for customer multicast infrastructures. Coming back to the BGP control plane, the RFC 4364 introduces a new BGP address family called MCAST-VPN for supporting NG-MVPN control plane operations. The new address family is assigned the subsequent address family identifier (SAFI) of 5 by IANA. A PE router that participates in a BGP-based NG-MVPN network is required to send a BGP update message that contains an MCAST-VPN NLRI, which contains route type, length, and variable fields (illustrated in detail a little later in this chapter). The value of the variable field depends on the route type. Seven types of NG-MVPN BGP routes, also known as MVPN routes, are specified. The first five route types are called auto-discovery (AD) MVPN routes. This chapter also refers to Type 1–5 routes as non-C-multicast MVPN routes. Type 6 and Type 7 routes are called C-multicast MVPN routes[27].

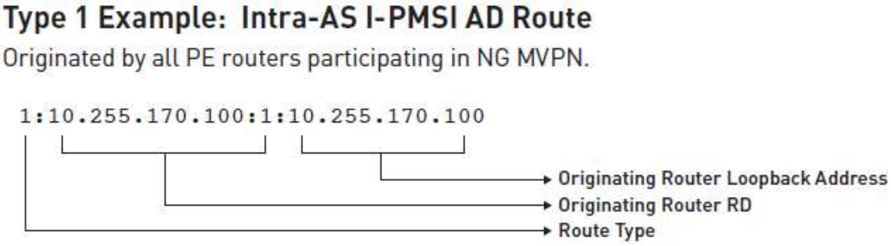
The table below provides details about the various MVPN routes used[27]:

Route type	Definition
Type 1 Intra-AS I-PMSI	Originated by all PE routers and used for advertising and learning Intra-AS MVPN membership information.

Route type	Definition
Type 2 Inter-AS I-PMSI AD	Originated by NG-MVPN ASBR routers and used for advertising and learning Inter-AS MVPN membership information.
Type 3 S-PMSI AD	Originated by Ingress PE routers and used for initiating a selective P-tunnel for a given C-Source and C-Group multicast stream.
Type 4 Leaf AD	Originated by Egress PE routers in response to a Type 3 announcement. It is used for indicating interest for a given C-Source and C-Group multicast stream.
Type 5 Source Active AD	Originated by a PE router (Ingress PE) when it learns about an active Multicast Source. The Type 5 route is announced to all Egress PE that belongs to a given NG-MVPN.
AD Type 6 Shared Tree Join	Originated by an Egress PE when it receives a PIM Shared Tree join (C-*, C-G) from the CE device.
Type 7 Source Tree Join	Originated by an Egress PE when it receives a Source Tree Join or when it receives a Type 5 route announcement from an Ingress PE.

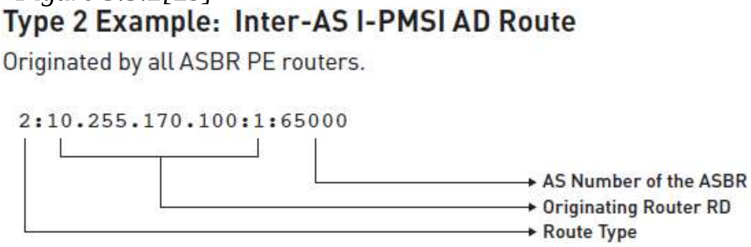
These BGP NLRI have very sharp structure. Let's look into definitions for these update types[27]:

Figure 3.3.1[29]


Type 1 Example: Intra-AS I-PMSI AD Route
Originated by all PE routers participating in NG MVPN.
1:10.255.170.100:1:10.255.170.100
Originating Router Loopback Address
Originating Router RD
Route Type

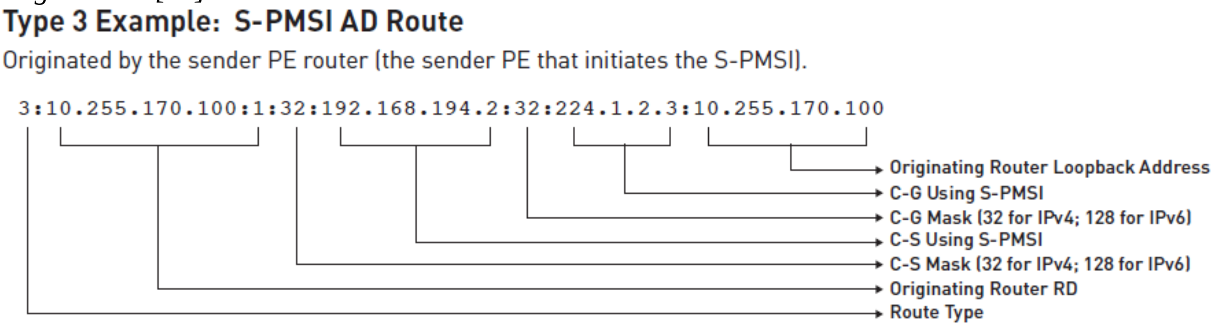
This Update is the first step of creation NG MVPN - all participating PEs exchange this NLRI type regardless of customer routes or traffic. It is just a message "I am here".

Figure 3.3.2[29]



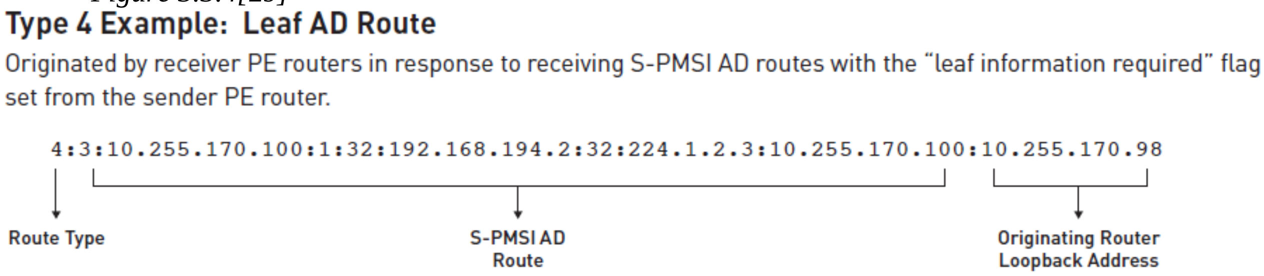
Just as Type 1, but advertises MVPN sources (ASBRs) external for AS.

Figure 3.3.3[29]


Type 3 Example: S-PMSI AD Route
Originated by the sender PE router (the sender PE that initiates the S-PMSI).
3:10.255.170.100:1:32:192.168.194.2:32:224.1.2.3:10.255.170.100
Originating Router Loopback Address
C-G Using S-PMSI
C-G Mask (32 for IPv4; 128 for IPv6)
C-S Using S-PMSI
C-S Mask (32 for IPv4; 128 for IPv6)
Originating Router RD
Route Type

This type of route is generated when Sender PE signal an Selective Path, for example if it detects significant bandwidth growth.

Figure 3.3.4[29]



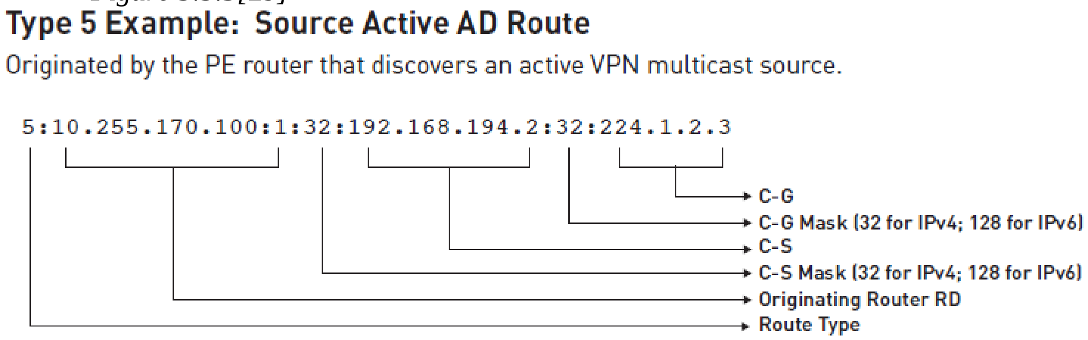
Type 4 Example: Leaf AD Route

Originated by receiver PE routers in response to receiving S-PMSI AD routes with the "leaf information required" flag set from the sender PE router.

4:3:10.255.170.100:1:32:192.168.194.2:32:224.1.2.3:10.255.170.100:10.255.170.98

An Ingress PE router signals a Type 3 BGP route for the creation of the S-PMSI. The creation of the S-PMSI can be defined based on a variety of criteria such as traffic thresholds (as used in the Draft- Rosen Data MDT creation). Upon receipt of the Type 3 BGP routes, Egress PE routers with interested receivers will respond to the announcement by advertising a Type 4 route. After this step, an Ingress PE router creates the appropriate P-tunnel[34].

Figure 3.3.5[29]



Type 5 Example: Source Active AD Route

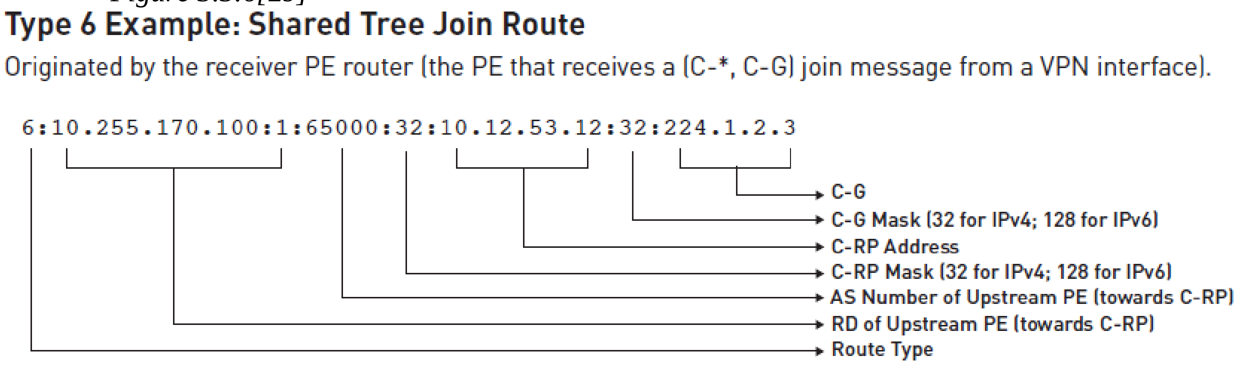
Originated by the PE router that discovers an active VPN multicast source.

5:10.255.170.100:1:32:192.168.194.2:32:224.1.2.3

C-G
C-G Mask (32 for IPv4; 128 for IPv6)
C-S
C-S Mask (32 for IPv4; 128 for IPv6)
Originating Router RD
Route Type

Type 5 routes carry information about active VPN sources and the groups to which they are transmit- ting data[24].

Figure 3.3.6[29]



Type 6 Example: Shared Tree Join Route

Originated by the receiver PE router (the PE that receives a (C-*, C-G) join message from a VPN interface).

6:10.255.170.100:1:65000:32:10.12.53.12:32:224.1.2.3

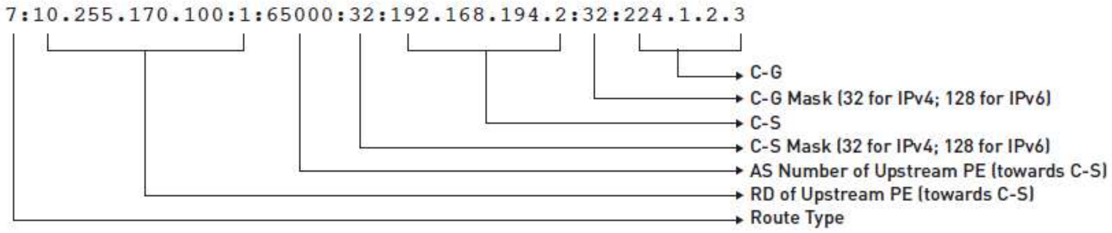
C-G
C-G Mask (32 for IPv4; 128 for IPv6)
C-RP Address
C-RP Mask (32 for IPv4; 128 for IPv6)
AS Number of Upstream PE (towards C-RP)
RD of Upstream PE (towards C-RP)
Route Type

This type of route generated when PE router receives request some group from Customer network, on PE-CE interface (vrf).

Figure 3.3.7[29]

Type 7 Example: Source Tree Join Route

Originated by the receiver PE router (the PE that receives a local (C-S, C-G) join message or the PE that already has a Type 6 route and receives a Type 5 route).


7:10.255.170.100:1:65000:32:192.168.194.2:32:224.1.2.3
C-G
C-G Mask (32 for IPv4; 128 for IPv6)
C-S
C-S Mask (32 for IPv4; 128 for IPv6)
AS Number of Upstream PE (towards C-S)
RD of Upstream PE (towards C-S)
Route Type

The C-multicast route exchange between PE routers refers to the propagation of C-joins from receiver PEs to the sender PEs. In an NG-MVPN, C-joins are translated into (or encoded as) BGP C-multicast MVPN routes and advertised via a BGP MCAST-VPN address family toward the sender PEs. Two types of C-multicast MVPN routes are specified[35].

- Type 6 C-multicast routes are used in representing information contained in a Shared Tree (C-*, C-G) join[35].
- Type 7 C-multicast routes are used in representing information contained in a source tree (C-S, C-G) join[35].

NG-MVPN provides an optimization in handling additional state information created in a Shared Tree environment. In a Shared Tree environment, every C-MCAST Source needs to register itself with the C-RP and the receivers need to join the shared tree via the RP. Therefore traffic initially flows over this shared tree prior to moving to the SPT or shortest path to the Source. Prior to joining the SPT, the receivers send PRUNE messages to the RP to stop the traffic fl owing through it for the group they have joined via the SPT. The RP further triggers another PRUNE toward the source, and now traffic flows via the Source Tree or SPT. This process increases the state information in the net- work and involves additional complexity, which adds no value since traffic eventually ends up flowing via the SPT. NG-MVPN by default provides a solution for this RPT to SPT switchover. Whenever an egress PE router generates a Type 6 BGP route for every PIM join (C-*, C-G) it receives a multicast domain from the customer; it does not advertise this route to remote PE routers unless it receives information of an active source via a Type 5 route. Sources by default do not register themselves with the RP; it is their locally connected router (CE device for NG-MVPN) that sends a unicast packet to the RP with the source's data packets encapsulated within. For the PE routers to learn about active sources, two conditions need to be met[27]:

1. One of the PE routers needs to be designated as the Customer-RP.[27]
2. An MSDP session needs to be established between the PE router and customer RP.[27]

It is only through one of the above procedures that a PE router (e.g., RP-PE) will learn of an active source and generate a BGP Type 5 route. PE routers with interested receivers will generate a Type 7 route toward the Ingress PE (and not toward the RP) forming the SPT[27].

NOTE: we see that there BGP Updates are translation for C-PIM messages and all PIM procedures and algorithms are working as the MPLS Network is just transparent box.

3.4 Ingress and Egress PE

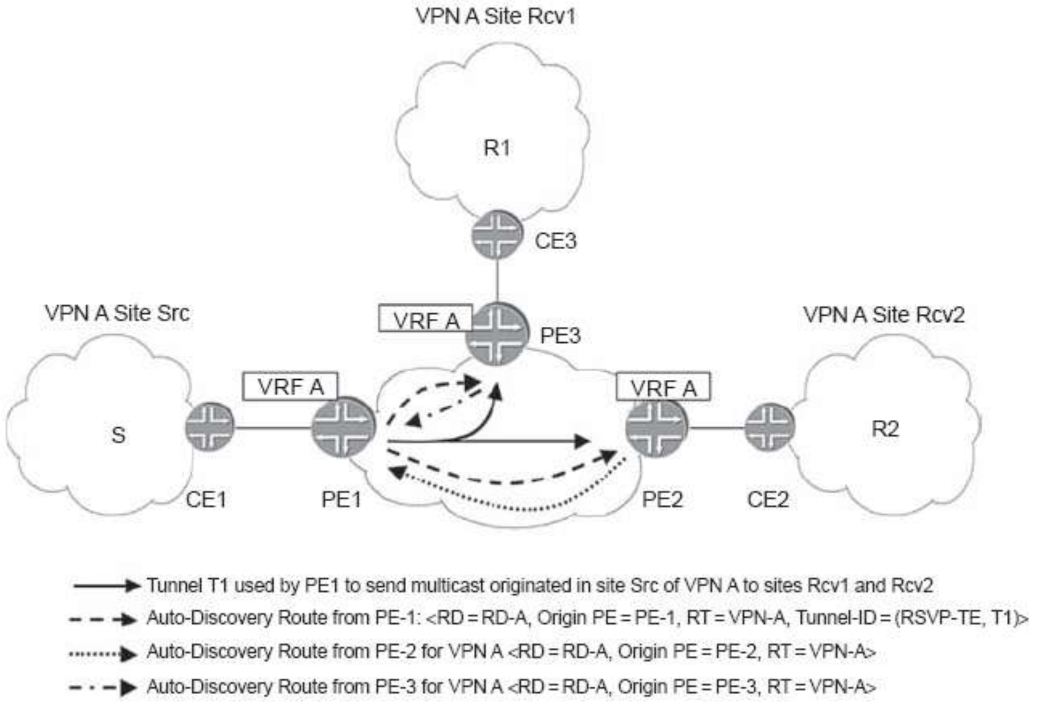
The term Ingress PE router refers to a PE router that has an active C-MCAST source for a given NG-MVPN. Egress PE routers are referred also as leaf nodes. This indicates that they are end points for a given C-MCAST traffic flow. One of the advantages of the NG-MVPN architecture is the ability to designate a PE router as a “Sender and Receiver” site or “Receiver Only” site. The former is an indication of the specified PE router to be able to both originate and receive C-MCAST traffic, while the latter is only able to receive traffic. This is agreed with normal MPLS PE definition - ingress PE - is router, where traffic enter MPLS network, egress PE - where traffic exits MPLS network[27].

3.5 Provider Multicast Service Interface (Inter-PE tunnels – inclusive and selective tunnels).

In the previous discussion, we focused on the role of the inter-PE tunnels in identifying the VPN to which the traffic belongs, but did not touch on which PEs the tunnel extends to or which multicast groups within the VPN use the tunnel. By default, inter-PE tunnels carry all multicast traffic for a particular VPN and extend to all PEs that have sites in the VPN, regardless of whether the sites have receivers for that traffic or not. Such tunnels are referred to as inclusive tunnels, or default tunnels (compare to default MDT). Inclusive tunnels may be wasteful of bandwidth, because traffic is forwarded to PEs which may end up discarding it. Imagine, for example, that in the network from Figure 1, an additional PE, PE4, was connected to a site of VPN A, site 4, which contains only unicast destinations and has no sources or receivers. The inclusive tunnel rooted at PE1 would span PE2, PE3 and PE4 even though PE4 has no interest in receiving any multicast traffic[32].

In contrast to inclusive tunnels, which carry traffic for all multicast groups in a VPN, selective tunnels carry traffic from just some multicast groups and can extend only to the PEs that have receivers for these groups. If in Figure 1, R2 is a receiver for group G1 and PE4 is connected to site 4 of VPN A, which contains receiver R4 for group G1, then the selective tunnel for group G1 would span from PE1 to PE2 and PE4, and not to PE3. This approach may be beneficial for high-bandwidth groups, where bandwidth optimization in the service provider network is required, but comes at the cost of having to create additional state for the extra trees in the service provider network. Note that the data-MDT of draft-Rosen is nothing but a type of selective tunnel[32].

Figure 3.5.1 [26]. Use of autodiscovery routes.


VPN A Site Rcv1
R1
CE3
VPN A Site Src
VRF A
PE3
VPN A Site Rcv2
VRF A
VRF A
S
R2
CE1
PE1
PE2
CE2
Tunnel T1 used by PE1 to send multicast originated in site Src of VPN A to sites Rcv1 and Rcv2
Auto-Discovery Route from PE-1: <RD = RD-A, Origin PE = PE-1, RT = VPN-A, Tunnel-ID = (RSVP-TE, T1)>
Auto-Discovery Route from PE-2 for VPN A <RD = RD-A, Origin PE = PE-2, RT = VPN-A>
Auto-Discovery Route from PE-3 for VPN A <RD = RD-A, Origin PE = PE-3, RT = VPN-A>

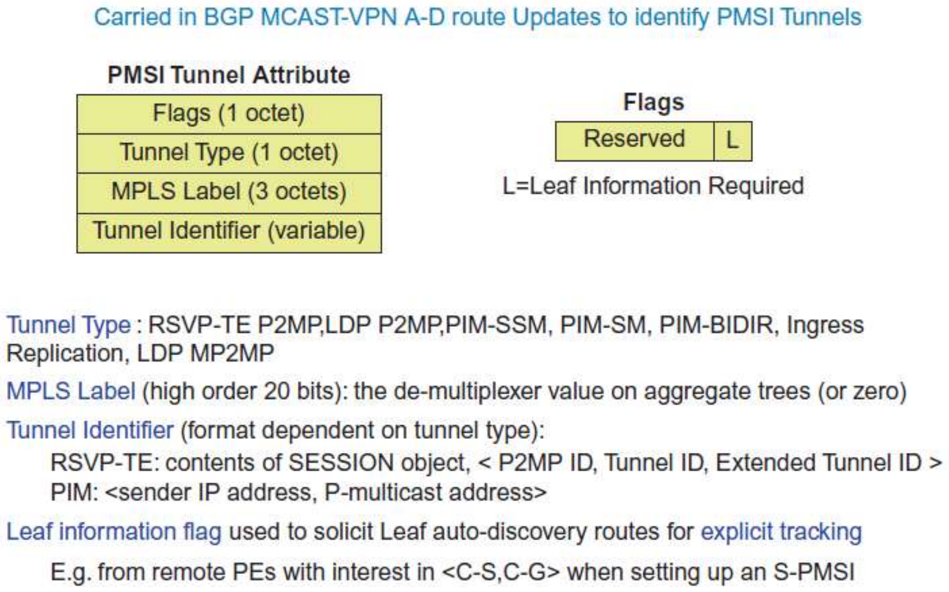
While looking at the various BGP route types for NG-MVPNs, references were made to a term known as PMSI (Provider Multicast Service Interface). Let us discuss this concept further, since it forms a key part of the MVPN architecture. The NG-MVPN architecture uses a PMSI to simplify and generalize different options for the MVPN solution. The PMSI distinguishes between services and the transport mechanism (P-tunnels) that support and realize the concept. When a PE gives a packet to PMSI, the underlying transport mechanism, P-tunnels, delivers the packet to some or all of the other PEs. A PMSI is a conceptual "overlay" on the Provider network with the following property: a PE in a given MVPN can give a packet to the PMSI, and the packet will be delivered to some or all of the other PEs in the MVPN, such that any PE receiving the packet will be able to determine the MVPN to which the packet belongs. For instance, an Ingress PE router may wish to send C-MCAST traffic only to given set of PE routers who express interest for the traffic or send it to all PE routers that participate in a given MVPN—irrespective of whether they have interested receivers or not. This is achieved by attaching an appropriate PMSI attribute to the BGP routes. The various PMSI types are[27]:

- Inclusive PMSI (I-PMSI)
- Selective PMSI (S-PMSI)

I-PMSI may be considered as a unidirectional P2MP connection between an Ingress PE and all Egress PE routers. Therefore, if a BGP route announcement has an attached I-PMSI attribute (e.g., a Type 1 BGP route used for auto-discovery may have an I-PSMI attribute attached), all traffic from the Ingress PE router is delivered to every other PE router participating in a given NG-MVPN. This behavior can be compared to the operation of the Default MDT in a Draft-Rosen MVPN[27].

Selective PMSI (S-PMSI) may be considered to be a subset of the I-PMSI, because a packet is delivered to a subset of PE routers that is participating in a given MVPN. This behavior can be com- pared to the Data MDT in Draft-Rosen. The format of the PMSI Tunnel Attribute is displayed in Figure 3.5.2[27].

Figure 3.5.2 [29]



Carried in BGP MCAST-VPN A-D route Updates to identify PMSI Tunnels

PMSI Tunnel Attribute
Flags (1 octet)
Tunnel Type (1 octet)
MPLS Label (3 octets)
Tunnel Identifier (variable)

Flags	
Reserved	L

L=Leaf Information Required

Tunnel Type : RSVP-TE P2MP,LDP P2MP,PIM-SSM, PIM-SM, PIM-BIDIR, Ingress Replication, LDP MP2MP

MPLS Label (high order 20 bits): the de-multiplexer value on aggregate trees (or zero)

Tunnel Identifier (format dependent on tunnel type):

RSVP-TE: contents of SESSION object, < P2MP ID, Tunnel ID, Extended Tunnel ID >
PIM: <sender IP address, P-multicast address>

Leaf information flag used to solicit Leaf auto-discovery routes for explicit tracking

E.g. from remote PEs with interest in <C-S,C-G> when setting up an S-PMSI

The Tunnel type indicates the various options available within the context of NG-MVPNs. Compared to Draft-Rosen where PIM-GRE was the only option available for transport of multicast traffic, the NG-MVPN provides options including RSVP-TE, MLDP, and also traditional PIM GRE- based schemes as options for Provider Tunnels (transport mechanisms). Each of these options will be discussed in subsequent sections[27].

The MPLS Label indicates the Label assigned. It is always set to “zero,” since both I-PMSI and S-PMSI tunnels always correspond to a single MVPN instance. MVPN differentiation is actually based on other values used in the BGP MVPN routes such as RD values, which uniquely identify a BGP/MPLS VPN[27].

3.6 Carrying traffic from several mVPNs onto the same inter-PE tunnel

Similar to the unicast case, where the tunnels in the provider network can carry traffic from multiple VPNs, the NG mVPN solution allows the P2MP trees in the provider network to aggregate traffic from multiple VPNs. This is not possible in the draft-Rosen approach, where one of the fundamental design decisions was the use distinct per-VPN MDTs. (For data-MDTs, these are distinct MDTs per particular set of (S, G)s within the same VPN.) In other words, draft-Rosen does not support the ability to aggregate traffic that belongs to different VPNs into a single tunnel[32].

From the provider’s point of view, the primary goal for aggregation is the flexibility of trading off bandwidth efficiency against the amount of multicast state in his network. Let us take a look at two options that a PE may use for aggregating multiple VPNs present on the PE onto the same distribution tree[32]:

1. *A single shared MDT rooted at the PE*. All VPNs present on the PE share the same distribution tree, resulting in a single tree in the service provider network. This reduces the

amount of state in the core of the network, at the expense of potentially wasting bandwidth, by sending traffic to PEs that do not need to receive it. For example, in a network that supports two VPNs, VPN A and VPN B, if routers PE1, PE2 and PE4 are servicing sites that belong to VPN A and routers PE1, PE3 and PE4 are servicing sites that belong to VPN B, then with this option multicast traffic arriving at PE1 from a site in VPN A would be sent to PE2, PE3 and PE4. This would be the case even though PE3 definitely does not need to receive the traffic, not having any site of VPN A attached to it[32].

2. *Multiple shared MDTs rooted at the PE.* An alternative to a single distribution tree is to have multiple distribution trees, all rooted at the same PE, each carrying a different subset of VPNs. One criterion for sharing trees could be a similar geographical distribution, which implies an overlapping the PEs that service the VPNs in question. Yet another alternative would be to have multiple distribution trees, each carrying a subset of multicast groups. Therefore, from a given VPN, some multicast groups could be carried on one tree, other multicast groups on another tree and so on, regardless of which VPN the groups belong to. An example of where this is useful is if a service provider has POPs in key major cities and other smaller cities. If several VPNs have some multicast groups that only have members in the major cities, it may be advantageous to have a multicast tree dedicated to serving those particular multicast groups[32].

Recall from the previous section that the MDT in the provider network was used to identify to which VPN traffic belongs to. In the cases where traffic from multiple VPNs shares the same distribution tree, the receiving PE must be able to identify the VPN to which a packet arriving on a MDT belongs. This is achieved by using a VPN label, as in the unicast case[32].

Because all egress PEs must use the same label, this has to be an upstream assigned MPLS label [RFC5331]. The context identifying the label is the P2MP tunnel over which traffic is aggregated. The label binding is advertised via a new type of BGP update originated by the ingress PE, as we will see in next Section[32].

The efficient use of aggregation depends on the ability to identify that two (or more) VPNs can be mapped onto the same distribution tree in the provider network. Ideally, a perfect overlap would exist for the two VPNs, but as long as the distribution tree is a superset of the PEs servicing both VPNs, this is not a strict requirement. In that case, traffic may be delivered to PEs that will immediately discard it, but even in such a case, it may still be beneficial to maintain the aggregation in the provider network. At which point this benefit vanishes depends on the wasted bandwidth as well as on other network-specific considerations, such as the existence of a small capacity path to a PE, which may get clogged with unnecessary traffic[32].

3.7 Creating inter-PE tunnels using BGP autodiscovery routes

In the previous sections, we have seen that inter-PE tunnels are used to carry traffic from the PE connected to the site containing the source to the PEs connected to the sites containing the receivers, thus providing the data plane for mVPN. Two things are required to set up the data plane. First, to enable the construction of such tunnels, it is necessary to discover to which PEs the tunnel must extend. This information must be acquired in an automatic way, because both mVPN membership and receiver information are likely to change, for example with the addition of a new site to the VPN or with a receiver leaving a particular group.

Second, to allow for the same flexibility provided by the data-MDT in the draft-Rosen solution, where some multicast streams are mapped to a separate distribution tree which spans only a subset of the PEs in the VPN, it is desirable to be able to specify which (S, G) streams map to a particular inter-PE tunnel (this is a many-to-one binding)[32].

Rather than inventing a new protocol for accomplishing these two tasks, the NG mVPN solution uses BGP for the automatic discovery and distribution of the information required for enabling the setup of inter-PE tunnels. Before describing the solution, let us first see what information is required for building an inter-PE tunnel[32]:

- The set of PEs to which the tunnel must extend. This information is required to enable the setup of the tunnel[32].
- Information identifying the tunnel. This information enables all PEs in the VPN to discover what tunnels other PEs will be sending traffic on for a particular VPN, and install forwarding state accordingly. This information is also used by the tunnel signaling protocols (e.g. RSVP-TE,LDP) to set up the actual inter-PE tunnels. Several pieces of information may be required: (1) the type and identity of the tunnel used to carry traffic between PEs, (2) the (S, G) streams which may be mapped to this tunnel (if this is a selective tunnel that carries traffic for just some(S, G)) and (3) the upstream-assigned VPN label if aggregation is used.(If aggregation is not used the identity of the transport tunnel can be used to determine to which VPN traffic belongs to.)[32].

To distribute this information with BGP, the MCAST-VPN NLRI is reused for advertising what is known as BGP autodiscovery routes between PEs. Similar to the unicast case, because this information is VPN-specific, the advertisements are made unique by attaching an RD and their distribution is constrained to the right VPN by tagging them with an RT. By default, the unicast RT can be used for tagging the autodiscovery routes, thus creating congruent unicast and multicast topologies, but a different RT could be used to control just the multicast topology. Note that the same BGP mechanisms available for VPN routes, such as RT filtering[RFC4684] and route reflectors can be reused for autodiscovery routes. Let us now take a look at the actual information that is carried in the autodiscovery route. The membership information is implicitly carried by including the address of the advertising PE and the relevant RT in the autodiscovery route. In addition to the membership information, the auto discovery route needs to carry information required to set up the inter-PE tunnel. This information is carried in a new BGP attribute called the P multicast Service Interface (PMSI) tunnel attribute and includes the type and identity of the tunnel. If aggregation is used, the upstream-assigned VPN label is also included. The type of the transport tunnel determines the protocol used for signaling it, for example mLDP or P2MP RSVP-TE. The identity of the tunnel depends on the type of the tunnel and is used by a particular tunnel signaling protocol for setting up the tunnel. For example, if the tunnel type is P2MP LDP, the tunnel identifier is the <Root Node Address, Generic LSP identifier> that is carried in the LDP P2MP FEC. Note that the tunnel identifier may be allocated by a PE before the tunnel is actually instantiated. In the case of mLDP, for example, the P2MP LSP is leaf-initiated. In this case, the tree may not yet exist at the time when the root sends out its autodiscovery route, but the root could pre-allocate and advertise a tunnel identifier[32].

Note that a PE connected to a site containing a source for a given mVPN must always generate an autodiscovery route with a PMSI tunnel attribute. However, PEs connected to sites containing receivers need not include the PMSI tunnel attribute in their advertisements, and in fact they do not need to generate the autodiscovery route at all, unless the tunnel type used for the mVPN is P2MP RSVP-TE (as in that case the identity of the leaves must be known to the source in order to enable the setup of the tunnel).On receipt of an autodiscovery route by a PE, assuming that the import RTs for a particular VRF present on the PE matches the RT attached to the autodiscovery route, the PE imports the route into the VRF and performs the following two types of actions[32]:

1. Instantiation of the tunnel, if required. For leaf-initiated trees like PIM or mLDP, the auto-discovery route generated by the root allows the receiver to find out the tree identifier and attach itself to the tree. For root-initiated trees, like P2MP RSVP-TE, autodiscovery routes received by the root allow it to identify the leaves that it should build its tree to. Note that in this case, the tree may be instantiated after the leaf PE has received the autodiscovery route from the root. Similarly, a new branch may be added to it by the root based on the receipt of an autodiscovery route from the leaf-PE[32].

2. Creation of the forwarding state for mapping traffic arriving on the tunnel to a particular VRF. The determination into which VRF to map the traffic is done based on the RT attached to the autodiscovery route. If no label is advertised in the PMSI tunnel attribute, then all traffic arriving on the tunnel is forwarded in the VRF, otherwise only traffic labeled with the VPN label is forwarded in the VRF. An interesting question arises regarding what happens when there are no receivers in a site attached to a particular PE, but the tunnel extends to all PEs in the mVPN(this could be the case when inclusive trees are used or when a tunnel aggregates traffic from multiple VPNs). Does this mean that it will be forwarded to the customer site? The answer is no, as there will be no forwarding state in the VRF for this multicast traffic (as there has no C-multicast route advertisement sent by the CE in this case)[32].

In addition to the basic functionality described above, autodiscovery routes can provide other features as well. One example is the dynamic creation or teardown of tunnels based on external triggers. For example, it is easy to create a selective tree (the equivalent of a draft-Rosen data-MDT tree) when traffic flowing for a particular (S, G) increases to a certain level, by the ingress PE simply generating the appropriate BGP autodiscovery route when the traffic reaches a given threshold. The autodiscovery route contains the identity of the selective tunnel and the (S, G) that is bound to it. Conversely, the tree can be torn down when no longer required by withdrawing the advertisement. In the case of a root-initiated tunnel, such as an RSVP-signaled P2MP, the ingress router needs to know which PEs have interested receivers for (S, G). This is achieved by those PEs sending corresponding leaf-autodiscovery routes[32].

To summarize, using BGP as an autodiscovery mechanism accomplishes two tasks: (1) it provides the information needed for the dynamic creation and teardown of P2MP inter-PE tunnels and (2) it allows the PEs to identify traffic arriving from other PEs as belonging to a particular VPN, thus making it possible to forward the traffic in the appropriate VRF[32].

3.8 Requirements for support of PIM ASM in an mVPN

Having seen the basic operation of BGP/MPLS mVPNs when the VPN customer is using the PIM-SM SSM mode of operation, let us now look at a few more advanced topics that arise in the context of PIM ASM. In the ASM mode of operation, PIM-SM provides a service model where there are multiple sources and multiple receivers for the same group. An example of an application using this model is a video-conferencing service, where many sources come and go and the locations of all these sources must be known to all receivers[32].

To understand how VPN customers running PIM-SM in the ASM mode can be supported, let us first look at the ASM mode of operation in the context of a plain IP (non-VPN) scenario. There are two important concepts that distinguish PIM-SM in the ASM mode from PIM-SM in the SSM mode. The first one is the concept of RP, the second is the concept of (multicast) domains. In PIM-SM in ASM mode, the discovery of the multicast sources is accomplished by introducing the concept of an RP as a centralized entity that knows about all the active sources. Designated routers connected to active multicast sources register the sources with the RP using PIM register messages. Receivers join an RP Tree (RPT) for the sole purpose of discovering the sources. However, receiving traffic over the RPT may not be ideal, and receivers may switch from the RPT to a shortest path tree (SPT) (typically as soon as the first packet is received from the source). This RPT/SPT interaction introduces a fair amount of additional complexity to the ASM mode of operation[32].

PIM-SM in the ASM mode defines the concept of a ‘multicast domain’. PIM-SM in the ASM mode supports interdomain operations by using the Multicast Source Discovery Protocol (MSDP) [RFC3618] to exchange information about active multicast sources among RPs in different domains, as described before. Since MSDP exchanges information about active multicast sources, it follows that only (S, G) information is exchanged among domains, even if within each domain both (*, G) and (S, G) information is exchanged. As a result, even if within each domain PIM-SM in the ASM mode is used; interdomain operations effectively look like PIM-SM in the SSM mode[32].

Given that at the interdomain level PIM-SM in ASM mode relies on the PIM-SM in SSM mode procedures, it follows that if an mVPN running PIM-SM in the ASM mode could be modeled as a collection of multicast domains interconnected by a service provider network, then the same mechanisms described in the previous sections for supporting mVPNs running PIM-SM in the SSM mode could be leveraged to support the ASM deployment. The next section describes how to accomplish this[32].

3.9 Carrying mVPN active source information using BGP source active autodiscovery routes

One can think of an mVPN as a collection of PIM-SM ASM multicast domains that must be interconnected across the service provider mVPN infrastructure, where each domain consists of all the mVPN sites connected to a given PE, plus the PE itself. As mentioned in the previous section, to create such an interconnection in a plain IP network, the information about the active sources is distributed between the RPs in each domain by running MSDP between the RPs[32].

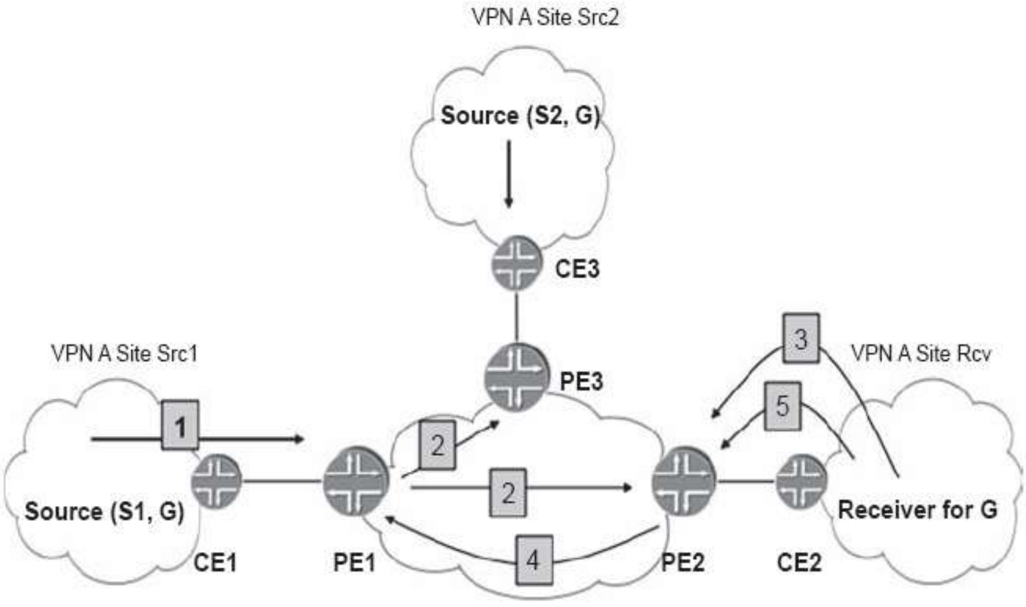
To interconnect these domains in the context of an mVPN, RPs are placed on the PEs. That is, a PE acts as a customer RP (C-RP) for the multicast domain formed by all the sites of a given mVPN connected to a given PE. Because the C-RP and the PE are the same entity, this mode of operation is referred to as the collocated RP/PE model. (Note that in this model every PE that has sites of a given mVPN connected to it becomes an RP for that mVPN.) A mechanism must be set in place to exchange information about active (multicast) sources among these PEs/C-RPs. This is done using a new type of BGP route, the BGP source active autodiscovery route, carried in the MCAST-VPN NLRI[32].

Figure 3.9.1 shows the mVPN deployment for VPN A where two sources, S1 and S2, located in sites Src1 and Src2 respectively, send traffic to receivers in site Rcv of the VPN. A PE that acts as a collocated C-RP finds out about the active sources among the sites connected to the PE by using the same procedures as the RP in a plain IP multicast scenario. In the example, PE1 finds out about S1 through the receipt of a PIM register message from the Designated Router connected to S1. This is the exchange labeled 1 in the figure. The PE then advertises this information to the other PEs (who also happen to be C-RPs) using a BGP source active autodiscovery route. In the example, PE1 sends source active autodiscovery routes to PE2 and PE3, the two exchanges labeled 2 in the figure[32].

As with C-multicast routes, the BGP source active autodiscovery routes carry in addition to the (S, G) information, two additional pieces of information:

(1) an RD to make advertisements unique per VPN and (2) an RT to constrain the distribution of the route to all the VRFs of a given mVPN (this RT can be the same as the RT used by unicast). Based on the receipt of source active route for a particular (S, G) from a remote PE, the local PE knows to which PEs to generate C-multicast routes when a Join (*, G) arrives from one of the receivers it services. The PE generates several C-multicast routes, one per each received source active autodiscovery route that has G. In the example in Figure 1, when PE2 receives a Join (*, G) from CE2 (exchange 3 in the figure), it generates a single C-multicast route (exchange 4), since it had received only a single source active autodiscovery route so far. When a receiver switches from the RPT to the SPT, the switch is localized to that site. In the example, if R switches to the SPT (exchange 5) by sending a Join (S1, G), no further advertisements need to be propagated in the service provider network. As a result, there is no shift in the traffic patterns in the service provider network when receivers in an mVPN switch from shared (RP-based) to source-based trees[32].

Figure 3.9.1 [27]



The above scheme works as long as an mVPN is willing to outsource its RPs to the mVPN service provider. What about mVPNs that do not want to outsource their RPs to the mVPN service provider? To answer this question note that as long as there is a way to provide one or more PEs with the information about active (multicast) sources of a given mVPN, it really does not matter whether these PEs act as fully functional C-RPs or not. In the scenarios where the RP is maintained within the mVPN customer's network the existing IP multicast mechanisms could be used to communicate information about active sources (S, G) from the mVPN RP to one or more PEs by either (a) running MSDP between the RP and the PE or (b) sending a PIM Register message between the RP and the PE[32].

In both cases, the PE maintains information about the active sources for a given mVPN (just like in the collocated RP/PE model), but the PE does not act as the RP. However, just like in the collocated RP option, the PE generates a BGP source active autodiscovery route based on the sources it knows about. Referring back to the example in Figure 1 and assuming that PE1 does not act as the RP, but instead RP1 is maintained within site Src1 of VPN A, only the exchange labeled 1 will differ as compared to the collocated PE/RP model. Namely, instead of S1 sending the PIM-register message to PE1, S1 sends the PIM-register message to RP1, and assuming MSDP is used between RP1 and PE1, RP1 sends an MSDP message to PE1. From this point on, the same exchanges as in the collocated model will take place. When S2 becomes active, the Designated Router connected to S2 sends PIM Register to the RP. Note that even if S2 and RP are in different sites, all what is required to propagate the PIM Register is unicast connectivity19 (as PIM Register are unicast messages)[32].

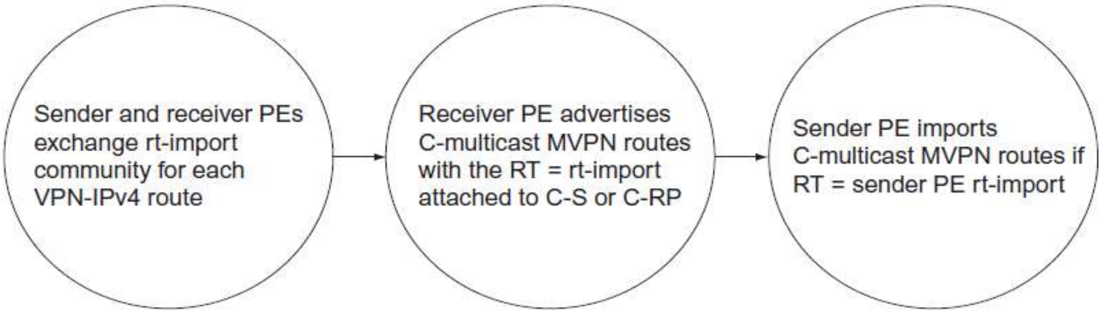
So far we covered just the exchange of multicast routing information. But what is about exchange of multicast data traffic? It is easy to see that all the mechanisms developed for the exchange of multicast data traffic for mVPNs running PIM-SM in the SSM mode work 'as is' in the context of mVPNs running PIM-SM in the ASM mode[32].

To summarize, to support PIM-SM in the ASM mode, information about active sources must be advertised between the PEs servicing sites of the mVPN. The PEs find about the active sources in the sites attached to them either by acting as the RP or by communicating with the RP (e.g. using MSDP). To distribute the source information among the PEs in the mVPN, a new type of route, the BGP source active autodiscovery route is used. Based on the receipt of this information, the PE will generate separate C-multicast routes towards each one of the PEs servicing the sources, when a Join (*, G) arrives from a receiver that it services[32].

3.10 Customer Multicast Routing Information and Route Targets

Based on the details discussed in the previous sections, it is now clear that C-multicast MVPN routes (Type 6 and Type 7) are only useful to the PE router connected to the active C-S or C-RP. Therefore, C-multicast routes need to be installed only in the VRF table on the active sender PE for a given C-G[18] in the case of SPT in the PE router closest to the C-RP or acting as the C-RP in the RPT mode where Type 6 routes are installed. To accomplish this, 2547bis-mcast proposes to attach a special and dynamic RT to C-multicast MVPN routes[27]:

Figure 3.10.1[27]


Sender and receiver PEs exchange rt-import community for each VPN-IPv4 route
Receiver PE advertises C-multicast MVPN routes with the RT = rt-import attached to C-S or C-RP
Sender PE imports C-multicast MVPN routes if RT = sender PE rt-import

The RT attached to C-multicast routes are also referred to as C-multicast import RT and should not be confused with the rt-import used for importing Unicast routing information. Heed that C-multicast MVPN routes vary from other MVPN routes in one fundamental way: they carry a dynamic RT whose value depends on the identity of the active sender PE at a given time and may change if the active PE changes. A local C-join that sends to a PE router defines the identity of the active sender PE router by executing a unicast route lookup for the C-S or C-RP in the unicast VRF table. It chooses the appropriate upstream PE also known as the active sender. After the active sender (upstream) PE is selected, the receiver PE constructs the C-multicast MVPN route corresponding to the local C-join. When the C-multicast route is created, the receiver PE needs to attach the correct RT to this route, aiming for the active sender PE. As stated, each PE router creates a unique VRF route import (rt-import) community and attaches it to the VPN-IPv4 routes. When the receiver PE does a route lookup for C-S or C-RP, it can extract the value of the rt-import associated with this route and set the value of C-multicast import RT to the value of rt-import. On the active sender PE, C-multicast routes are imported only if they carry an RT whose value is the same as the rt-import that the sender PE generated. Let us look at an output of this value, displayed on a sender PE[19][27].

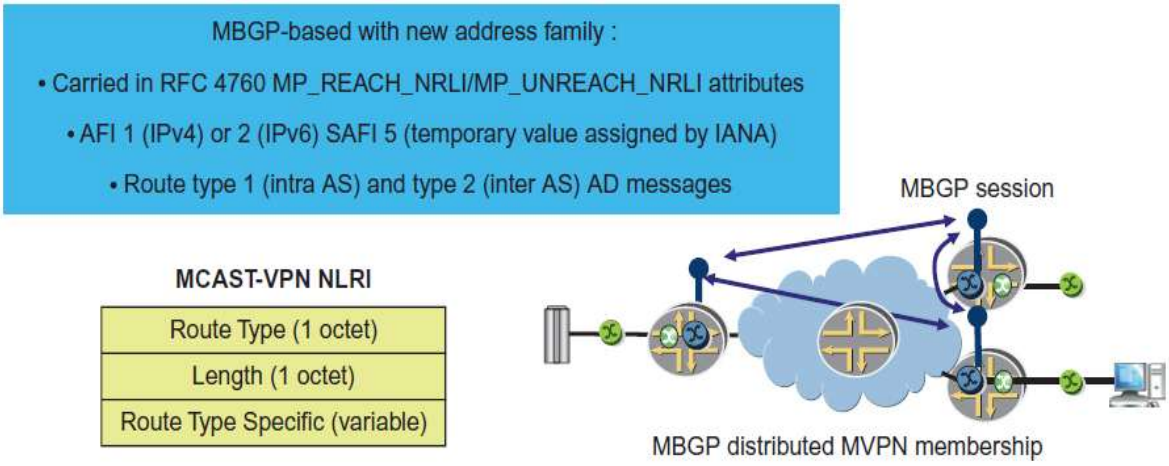
3.11 Putting the Building Blocks into Perspective

Here is a summary of the steps associated in the BGP control plane used to enable multicast traffic flows within an NG-MVPN[27].

1. MVPN Membership and Autodiscovery[27].
2. MVPN Membership and Auto-discovery I-PMSI setup[27].

In Figure 3.11.1 we use PIM-SM as the P-tunnel, since we have yet to introduce other MPLS-based P-tunnels such as RSVP-TE since we have yet to introduce other MPLS-based P-tunnels such as RSVP-TE [27].

Figure 3.11.1[27]



MBGP-based with new address family :

- Carried in RFC 4760 MP_REACH_NRLI/MP_UNREACH_NRLI attributes
- AFI 1 (IPv4) or 2 (IPv6) SAFI 5 (temporary value assigned by IANA)
- Route type 1 (intra AS) and type 2 (inter AS) AD messages

MCAST-VPN NLRI

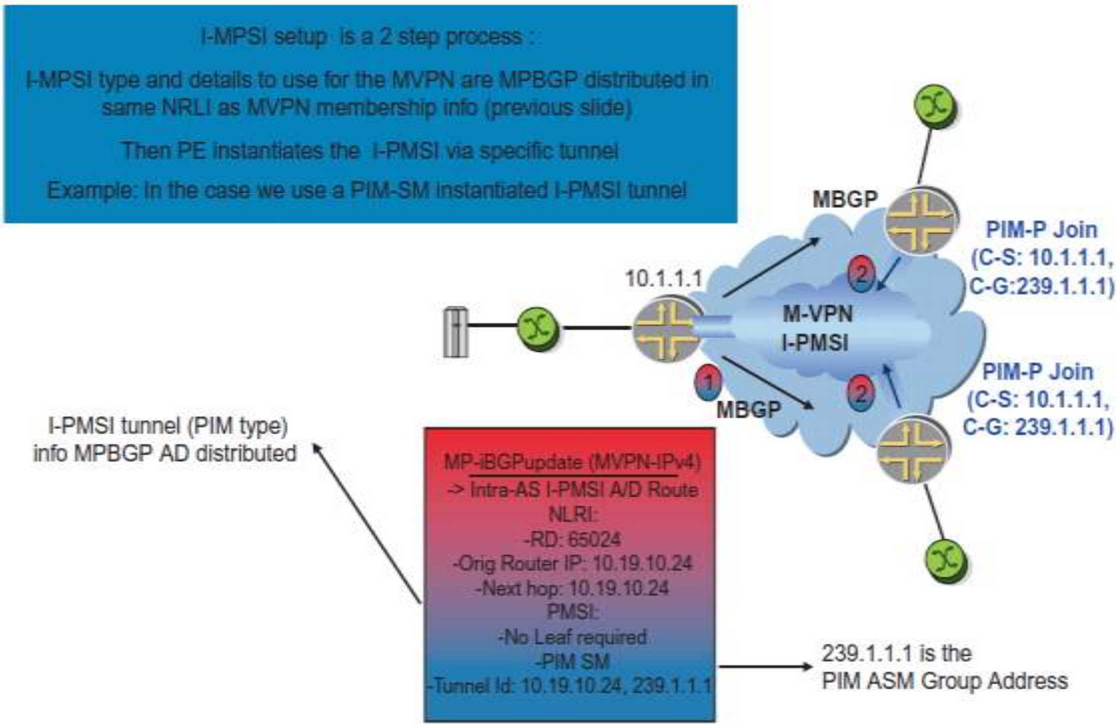
Route Type (1 octet)
Length (1 octet)
Route Type Specific (variable)

A PIM-SM P-tunnel uses an ASM Group Address (similar to Default MDT in Draft-Rosen), such as 239.1.1.1, which we used earlier. An NG-MVPN with PIM-SM or PIM-SSM uses GRE encapsulation and is similar to Draft-Rosen with the exception of using the superior BGP control plane. Some providers opt for this model as a first step to migrate to NG-MVPNs, wherein the control plane is changed and the Data plane is still preserved. It can be considered as an option for a phased migration. However, Figure 3.22 demonstrates the setup of a BGP Type 1 route along with the I-PMSI setup. An I-PMSI field with a PIM-SM P-tunnel (see Figures 2,3) includes the following[27]:

Receivers come online and C-JOIN messages are sent to the Receiver PE routers. The Receiver PE routers perform a route lookup for C-S and C-RP, respectively, and extract the RD, rt-import, and src-as associated with each route. PE with an interested receiver for a given group originates a Type 7 route upon receipt of a Type 5 route carrying RT information (value matching rt-import); PE without an interested receiver creates a Type 6 route, but does not advertise it[27].

1. The Ingress PE compares the received Type 7 routes with its import-RT and, based on a match, the route gets accepted and passed onto the C-PIM infrastructure[27].

Figure 3.11.2[27]



I-MPSI setup is a 2 step process :

I-MPSI type and details to use for the MVPN are MPBGP distributed in same NRLI as MVPN membership info (previous slide)

Then PE instantiates the I-PMSI via specific tunnel

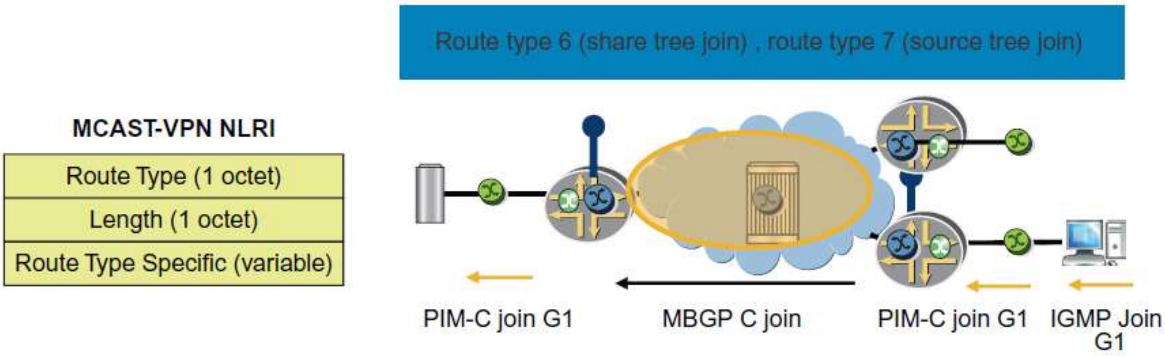
Example: In the case we use a PIM-SM instantiated I-PMSI tunnel

MP-iBGPupdate (MVPN-IPv4)
-> Intra-AS I-PMSI A/D Route
NLRI:
-RD: 65024
-Orig Router IP: 10.19.10.24
-Next hop: 10.19.10.24
PMSI:
-No Leaf required
-PIM SM
-Tunnel Id: 10.19.10.24, 239.1.1.1

I-PMSI tunnel (PIM type) info MPBGP AD distributed

239.1.1.1 is the PIM ASM Group Address

Figure 3.11.3[27]



Route type 6 (share tree join) , route type 7 (source tree join)

MCAST-VPN NLRI

Route Type (1 octet)
Length (1 octet)
Route Type Specific (variable)

3.12 NG mVPN Data-plane

In this section, we move on to the Data plane setup, where one of the various supported options may be used for setting up Provider Tunnels in the network to facilitate C-MCAST traffic flows. The NG-MVPN framework currently provides support for the Provider Tunnels seen in next table[27]:

- Tunnel Type 0 = No Tunnel Information is Present
- Tunnel Type 1 = RSVP-TE Point-to-Multipoint LSP
- Tunnel Type 2 = MLDP Point-to-Multipoint LSP
- Tunnel Type 3 = PIM-Source Specific Multicast
- Tunnel Type 4 = PIM-Sparse Mode Tree
- Tunnel Type 5 = PIM-Bidirectional Tree
- Tunnel Type 6 = Ingress Replication

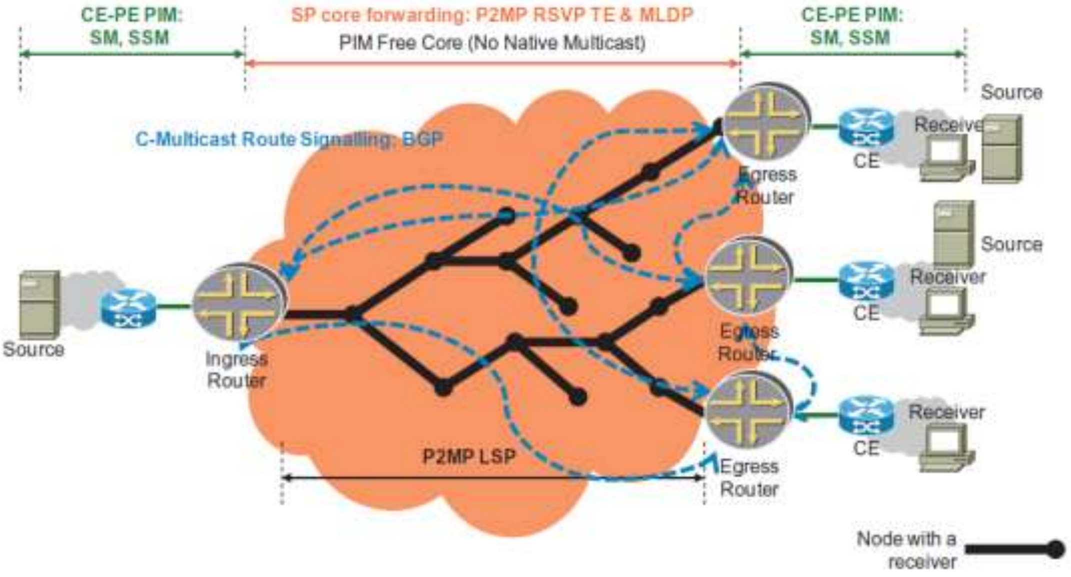
- Tunnel Type 7 = MLDP MP2MPLSP

In theory, each NG-MVPN can be set up to use a different Provider Tunnel or Data plane. However, this is very unlikely since an operator would prefer to use a Provider Tunnel because all unicast traffic uses a single Data plane such as RSVP-TE. The NG-MVPN framework permits an operator to use different Data plane protocols for unicast and multicast traffic, respectively. P-tunnels are rooted at the Ingress PE (Sender) and receiver PEs join a given P-tunnel that signaled based on the NG-MVPN they belong to or based on C-MCAST receiver interest. It is worthy to note that the sender PE goes through two steps when setting up the Data plane. One step includes using the PMSI attribute; it advertises the P-tunnel it will be using via BGP using a Type 1 route. In step two it actually signals the tunnel using whatever tunnel signaling protocol is configured for that VPN. This allows receiver PE routers to bind the tunnel signaled to the VPN that imported the Type 1 intra-AS AD route. Binding a P-tunnel to a VRF table enables a receiver PE router to map the incoming traffic from the core network on the P-tunnel to the local target VRF table[27].

Point-to-Multipoint LSPs

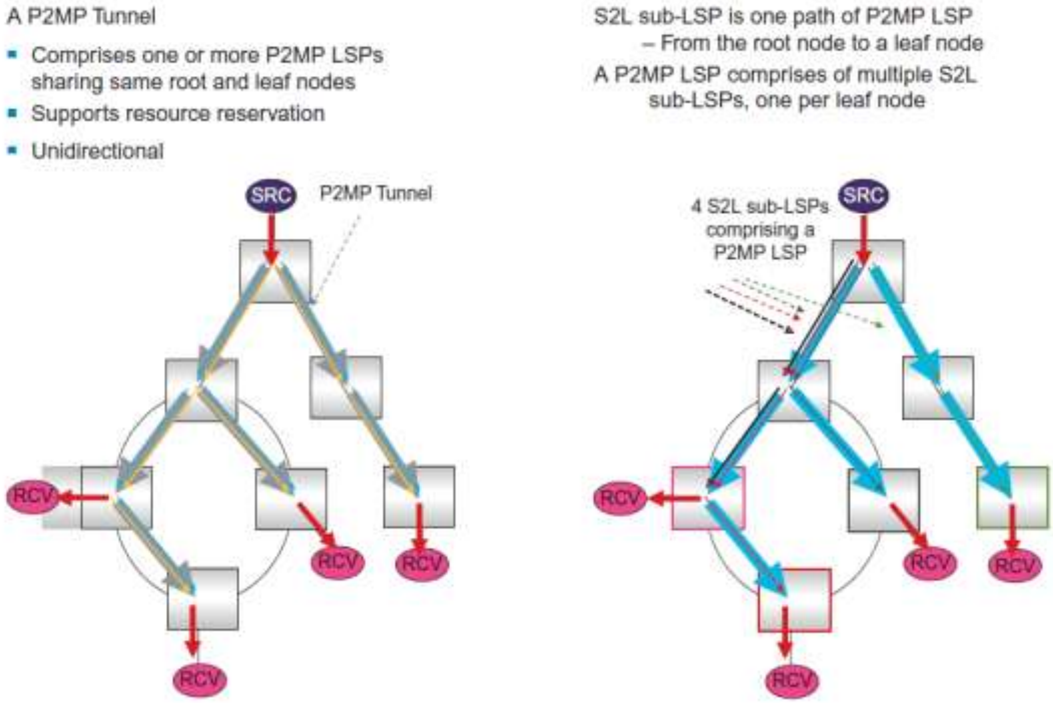
In Table above there is a reference to P2MP LSPs. The need for an MPLS-based transport for MVPN traffic against PIM GRE used as an overlay was reviewed. One of the key benefits discussed was the need for a converged platform for all traffic types such as Multicast and Unicast, which uses MPLS-based transport and a PIM free core. From an NG-MVPN point of view, there are two Provider Tunnel options that use MPLS as a transport mechanism: RSVP-TE and MLDP (Multicast LDP). All of the other Provider Tunnels use GRE-based transport similar to Draft Rosen, with the exception of the control plane, which is based on BGP (except ingress replication, which we would discuss later in this chapter). The hierarchy from an NG-MVPN BGP-based control plane with P2MP LSPs is illustrated in Figure 3.12.1[27].

Figure 3.12.1. [27]



The P2MP LSP and associated sub-LSPs are signaled by the ingress PE router. The information about the P2MP LSP is advertised to egress PEs in the PMSI attribute via BGP. The ingress PE router signals P2MP sub-LSPs by originating P2MP RSVP PATH messages toward egress PE routers. The ingress PE learns the identity of the egress PEs from Type 1 routes. Every RSVP PATH message carries an S2L_Sub_LSP Object along with the P2MP Session Object. The S2L_Sub_LSP Object is a vessel to a 4-byte sub-LSP destination (egress) IP address. Sub-LSPs associated with a P2MP LSP can be signaled automatically by the system or via a static sub-LSP configuration. When they are automatically signaled, the system chooses a name for the P2MP LSP and each sub-LSP associated with it using the following naming convention (see Figure 3.12.2)[27].

Figure 3.12.2[27]. Point-to-multipoint LSP



A P2MP Tunnel

- Comprises one or more P2MP LSPs sharing same root and leaf nodes
- Supports resource reservation
- Unidirectional

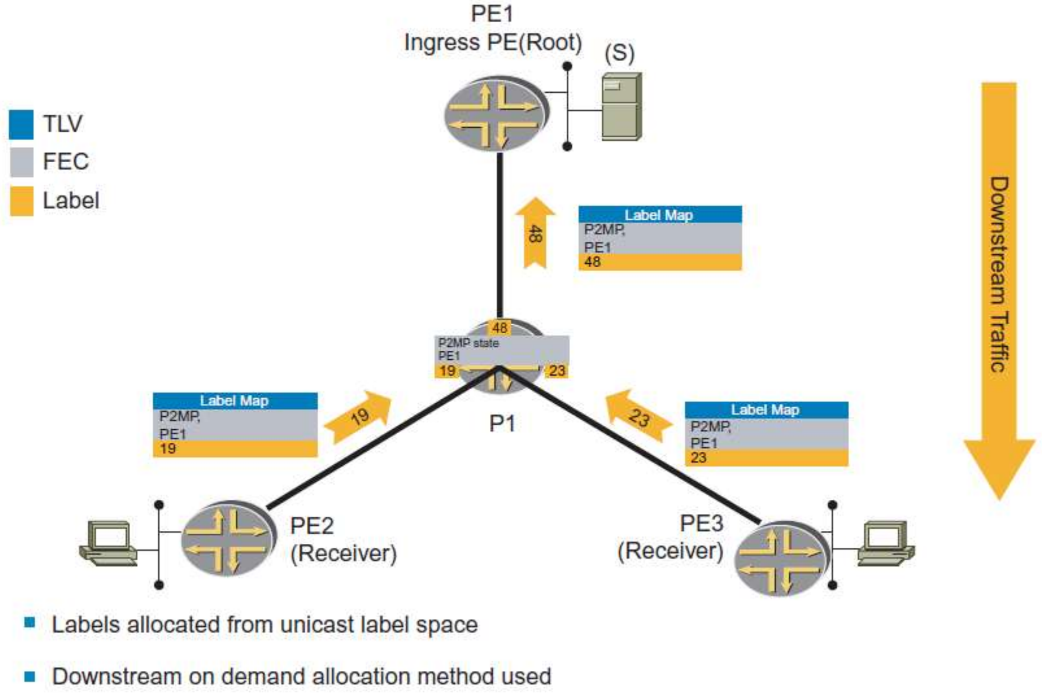
S2L sub-LSP is one path of P2MP LSP
– From the root node to a leaf node
A P2MP LSP comprises of multiple S2L sub-LSPs, one per leaf node

3.13 MLDP Provider Tunnels

The details for using MLDP (Multicast LDP) as Provider Tunnels are defined in RFC 6826. In this section we will discuss the setup and signaling of MLDP P2MP LSPs. From a functional standpoint, MLDP LSPs can be used to signal both I-PMSI and S-PMSI tunnels. The only difference between MLDP and RSVP-TE is that the former cannot provide any TE-specific features such as bandwidth guarantees, built-in Link Protection, and user-defined paths based on constraints. The setup of I-PMSI or S-PMSI tunnels is similar to the process defined in RSVP P2MP configurations since the attributes are attached within the BGP MVPN routes. Hence there is no additional process involved. An MLDP P2MP LSP allows traffic from a single root (or ingress) node to be delivered to a number of leaf (or egress) nodes. Similar to RSVP-TE, only a single copy of the packet will be sent on any link traversed by the Multi Point (MP) LSP. This is accomplished without the use of a multicast protocol in the network. There can be several MP LSPs rooted at a given ingress node, each with its own identifier. The leaf nodes (Egress PEs) of the MP LSP come to know about the root node (Ingress PE) and identifier of the MP LSP to which they belong via the BGP control plane and routing announcements. While using RSVP-TE we noticed that the Ingress PE initiates the P2MP LSP using RSVP PATH messages, which Egress PE routers respond to with appropriate Label information. With MLDP P2MP LSPs, the leaf nodes initiate P2MP LSP setup and teardown. For instance, if an Egress PE router receives a BGP Type 1 route from an Ingress PE router with an I-PMSI attribute, the Leaf initiates the setup of the P2MP LSP and also installs a forwarding state to deliver the traffic received on a P2MP LSP to wherever it needs to go. Transit nodes install the MPLS forwarding state and propagate the P2MP LSP setup (and teardown) toward the root, and the root node installs the forwarding state to map traffic into the P2MP LSP. For the setup of a P2MP LSP with LDP, we define one new protocol entity, the P2MP FEC Element, to be used in the FEC TLV[27].

From the context of NG-MVPNs, the LDP Opaque Value Element is not used for any applications. However, they are used in certain vendor (Non-Next-Generation Multicast VPN) mVPN implementations that use MLDP as the forwarding plane as well. Now we can look at the setup of MLDP P2MP LSPs. As mentioned a bit earlier, an Egress PE knows the next hop for the Ingress PE connected to the multicast source, based on the BGP announcements. To receive the LSPs it needs to tell the upstream router what label it needs to use for this multicast stream. To advertise the label it will send a Label mapping to its upstream router for this multicast source. The label mapping will contain the label to be used. Since the upstream router does not need to have any knowledge of the source, it only contains an FEC to identify the P2MP tree. If the upstream router does not have any FEC state, it will create it and install the assigned downstream outgoing label. If the FEC state was created and this router is not the LSP ingress of the P2MP tree, it needs to forward a label mapping upstream. This operation continues until we reach the LSP ingress router. This process is illustrated in Figure 3.12.3[27].

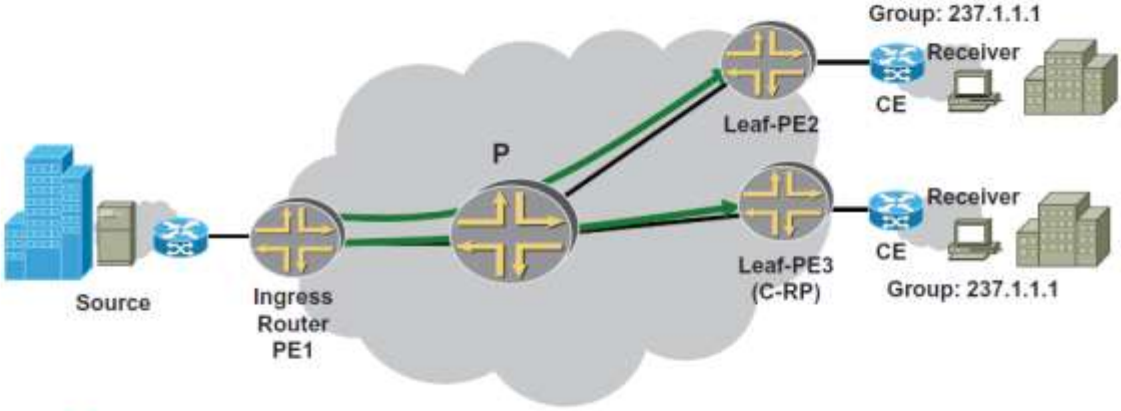
Figure 3.13.3[27] Label allocation for MLDP



3.14 PIM SSM/PIM SM Provider Tunnels

In this section, we look at a network setup using PIM-SSM of PIM-SM -based Provider Tunnels as the Data plane along with the BGP control plane. The following Figure will be used to demonstrate the functionality. From a data plane view PIM SM and SSM are not different - they are both using tunnels using multicast MDT group address.

Figure 3.14.1[27]



- The router "P" does not indicate a single Provider router, and indicates a Carrier Provider core that may contain many devices. A single device is just given ONLY for illustrative reasons.

3.15 Migration Options

Network operators aspire to guarantee their existing MVPN deployments can easily migrate from Rosen mVPNs to the next-generation solution . The versatility of NG MVPN enables the BGP control plane many scaling advantages. Consider as a section of one migration strategy, an MVPN VRF on a PE could run the Draft-Rosen and the BGP-MVPN control plane whereas the data plane proceeds to be a PIM-SM GRE. This path allows all the PEs to be upgraded to the BGPMVPN control plane while they proceed to execute the Draft-Rosen control plane. After that the Draft-Rosen control plane can be deconfigured one <PE, VRF> at a time to migrate the PEs to the BGP-MVPN control plane seamlessly[26].

Thus, providers could migrate to NG MVPN one MVPN at a time and/or one PE at a time. The data plane can then be migrated seamlessly to P2MP MPLS over time. NG MVPN also inherently supports the Carrier-of-Carrier model very similar to the 2547 Layer 3 Unicast VPN model[26].

This effective technique can be used for migrating from Draft-Rosen islands to NG MVPN. The Draft-Rosen VPN islands can seamlessly be carried over NG MVPN, which gives providers the option to gain experience with the NG MVPN without disruption of existing deployments. This approach allows providers to grow the NG MVPN carrier AS while limiting the growth of the Draft-Rosen VPN islands[26].

There are several multicast applications driving the deployment of new MVPNs and requiring the level of scale that is particularly addressed by the NG MVPN solution. Some of the key emerging applications being discussed include the following[26]:

- Layer 3 VPN multicast service offered by service providers to enterprise customers[26]
- Video transport applications for separation/virtualization between different customers[26]
- IPTV wholesale[26]
- Multiple content providers attached to the same network[26]
- Distribution of media-rich financial services or enterprise multicast services[26]
- Multicast backhaul over a metro network[26]

3.16 Next-Generation Multicast in VPLS

VPLS, a key enabler for delivering multipoint Ethernet services and Layer 2 backhaul of DSL or mobile traffic, requires an efficient mechanism to transport multicast traffic. By default, VPLS implementations use ingress replication, which does not offer bandwidth efficiency for multicast traffic. The use of P2MP LSPs with BGP VPLS allows replication on the network only where it is required[26].

In this approach, each PE needs to tell the other PEs the identity of the RSVP-signaled P2MP LSP on which it will send the traffic (multicast, broadcast, or unknown) for a particular BGP-VPLS instance. It sets up the control plane by putting the RSVP session object of the P2MP LSP in a BGP update. VPLS makes use of the BGP control plane that is used for NG MVPN to enable P2MP LSPs, and this is one of the main benefits. Further, BGP auto-discovery allows for the setup of dynamic P2MP LSPs such that the leaves of the P2MP RSVP-TE LSP do not need to be statically configured. In other words, when PE auto-discovers new PE members of a VPLS instance through BGP, it automatically adds a new leaf to the corresponding P2MP LSP[26].

Providers who want to offer multicast virtualization services for Layer 3 VPN and VPLS have an option to use a common control plane (BGP) and data plane (MPLS) framework that leads to a consistent service model and simplified operations[26].

Part 4. Comparison of PIM/GRE (Rosen)and BGP/MPLS mVPNs

4.1 Intro

As described before, based on Draft-Rosen that specifies a virtual router architecture, VPN multicast traffic is overlaid on top of a BGP-MPLS network, and uses PIM-SM for swapping control plane information and setting up multicast forwarding state on the Layer 3 VPN infrastructure. The C-multicast (customer domain multicast) protocol information which is typically client PIM (C-PIM) join/prune messages, received from local CE routers is propagated to other PEs using these PE-PE PIM sessions over the VPN-specific virtual network. The crucial issue here is that the PIM sessions are between the VRFs. Therefore, for a distributed MPVN, a PE controls a PIM session with each of the other PE that has membership in that MVPN and poses a complex and important scaling challenge. Consider that there are a 1000 MVPNs per PE, and there are 100 sites per MVPN, this would mean that there would be 100,000 PIM neighbors per PE, with a result of 3300 PIM hellos/second.[25]

Then, across the MVPN provider network, VPN multicast control as well as data traffic are transmitted via multipoint GRE tunnels. Different instances of PIM protocol running on the provider side (P-PIM) will signal these tunnels. The provider assigns a multicast group address called a GRE header, and this is used for tunneling VPN multicast data and control traffic. These tunnels are VPN multicast distribution trees (MDTs), and the GRE is given multipoint property by this multicast group address[26].

With this virtual router model, control and data plane scaling issues arise, and MVPN providers must maintain two different routing and forwarding mechanisms for VPN unicast and multicast services. As a result, there is an effort by the IETF Layer 3 VPN working group to specify the next-generation MVPNs (also referred to as NG MVPNs). The working group published an IETF RFC 6513 that is a superset of the previously mentioned specifications for MVPNs.[26] There are two main IETF standards: RFC 6513, which outlines the procedures of the BGP NG MVPN, and RFC 6514, which describes the building blocks for the different options[26].

As stated when the chapter started, the L3VPN IETF working group draft on multicast support, [VPN-MCAST] covers both the PIM/GRE (draft-rosen) and the BGP/MPLS (NG) approaches, and deployments of both exist in several large networks. Having described both solutions in this chapter, let us compare them by looking at (a) the VPN model implemented by each of the solutions, (b) the control-plane protocol used in each case, (c) the data-plane mechanisms, (d) the operation in an inter-AS scenario and (e) deployment considerations for a service provider. Let us examine these aspects separately[32].

4.2 VPN model used

Some of the most significant differences between the two solutions are rooted in the underlying VPN models used: the Virtual Router model, used by the PIM/GRE solution, and the Aggregated Routing model, used by the BGP/MPLS solution. Therefore, in order to better understand the differences between the two approaches, let us first examine the differences between the two models[32].

In the Virtual Router model, the exchange of VPN routing information among PEs is accomplished by operating separate instances of routing protocols among the PEs, one instance for each VPN. The exchange of VPN data traffic among PEs is accomplished by setting up VPN-specific tunnels between PE devices, where logically these tunnels are between the VRFs which are within the PE devices. These tunnels are used as if they were normal links between normal routers, and therefore routing protocol data for each customer VPN is also tunneled over them, creating a very tight coupling between the control and data planes[32].

In contrast to the Virtual Router model, the Aggregated Routing model uses a single instance of a routing protocol for carrying VPN routing information among the PEs, and the routing information for multiple different VPNs is aggregated into this instance. Just like with the Virtual Router model, the Aggregated Routing model uses VPN-specific tunnels set up between PE devices to carry data traffic between the PEs. However, in contrast to the Virtual Router model, these tunnels are used solely by the data plane, and routing protocol data for the VPN is not forwarded over them. As a result, the exchange of VPN routing information among PEs (control plane) is fully decoupled from transporting VPN user data traffic between PEs (data plane). This, in turn, facilitates support for various tunneling technologies with the same common control plane[32].

Let us compare the two models in terms of two different properties:

1. *Number of routing adjacencies maintained.* Exchange of VPN routing information in the Virtual Router model requires establishment of a distinct control plane operating across the service provider network for each VPN, which results in requiring PEs to maintain a potentially large number of routing peers and routing adjacencies. The Aggregated Routing model greatly reduces the number of routing peers and adjacencies which the PEs must maintain relative to the Virtual Router model, as there is no longer any need to maintain more than one such adjacency between a given pair of PEs.[32]
2. *Support of different tunneling technologies for forwarding traffic.* In the Virtual Router model, there is a tight coupling between the control and the data planes, as the data plane is also used for forwarding the control-plane information. This makes it difficult to support other technologies for setting up the inter-PE tunnels. In contrast, in the Aggregated Routing model there is no such dependency.[32]

Table 1[26]: Comparison of Draft-Rosen and NG MVPN

	DRAFT-ROSEN	**NG MVPN**
Transport	PIM-SM GRE	Different MVPNs can use different tunneling technologies (P2MP MPLS or PIM-SM GRE).
Signaling	PIM	BGP, same model as unicast

	DRAFT-ROSEN	NG MVPN
		Layer 3 VPN. Supports auto-discovery of routes.
PE-PE signaling sessions	Each PE needs a separate PIM adjacency with each remote PE per VRF.	Each PE uses existing IBGP sessions, which may only require sessions with the route reflectors.
VPN traffic aggregation	No ability to aggregate multiple MVPNs into a single inter-PE tunnel.	It is possible to aggregate multiple (S,G) of a given MVPN into a single selective tunnel and aggregate multiple P2MP LSPs using P2MP LSP hierarchy.
Inter-AS operations	Inter-AS/inter-provider operations options B and C (as defined in RFC 4364) require PEs in different ASes/providers to have direct PIM routing peering.	NG MVPN seamlessly works with all three options (A, B, and C as defined in RFC 4364) available for inter-AS unicast. It also has the concept of segmented inter-AS trees that allows each AS to independently run a different tunneling technology.
Provider Tunnel (P-tunnel) mesh requirement	Required between the PEs, which forces providers to sell an MVPN service where every customer site can be a source and a receiver.	P-tunnel mesh requirement is removed. Allows providers to support MVPN customers where multicast sources can be limited to a subset of its sites. Provides the flexibility to build pricing models for an MVPN service based on sites connected to either sources or receivers or both.

It is easy to see that the PIM/GRE solution is nothing but an instance of the Virtual Router model, while the NG mVPN solution (just like unicast BGP/MPLS) is an instance of the Aggregated Routing model. Therefore, the drawbacks of the Virtual Router model are applicable ‘as is’ to the PIM/GRE solution. Likewise, all the benefits of the Aggregated Routing model are applicable ‘as is’ to the NG mVPN solution[32].

4.3 Protocol used in the control plane

More differences between the two approaches are a consequence of the use of PIM and BGP, respectively, as the control-plane protocol for exchanging mVPN customers multicast routing information. As we will see below, they compound the disadvantages of the Virtual Router model and enhance the advantages of the Aggregated Routing model[32].

The disadvantages of the Virtual Router model are compounded in the draft-Rosen solution by the use of PIM as a control-plane protocol due to the following additional factors:

- The need for periodic refreshes. PIM relies on the periodic exchange of the complete routing information. In contrast, the NG mVPN solution benefits from using BGP to exchange mVPN multicast routing information, as BGP is based on the technique of incremental updates, and therefore is more efficient in terms of control-plane resources than PIM[32].
- The need for direct peerings in the inter-AS /interprovider scenario. The PIM/GRE solution requires PEs in different ASs/providers to have (direct) PIM routing peering, as long as these PEs have at least one mVPN in common. This is one of the direct consequences of following the Virtual Router model. In contrast, the NG mVPN solution allows restricting the exchange of routing information (including mVPN routing information) to only the ASBRs and does not have a direct exchange of routing information among PEs belonging to different ASs/providers[32].

The advantages of the NG mVPN solution over draft-Rosen are enhanced by the use of BGP rather than PIM as a control-plane protocol due to the following factors:

- Support for hierarchical route distribution (hierarchical control plane). BGP has built-in support for hierarchical route distribution using route reflectors. This allows the NG mVPN solution to completely eliminate the PE–PE routing adjacencies and make the number of backbone adjacencies a PE has to maintain into a small constant which is independent of the number of PE devices, which in turn, significantly improves the scaling properties, compared to the Virtual Router model. In inter-AS setups, the ASBRs are also part of the hierarchical control plane of BGP, as the exchange of BGP routes between adjacent ASs is confined to the ASBRs that interconnect these ASs[32].
- Support for scalability mechanisms for route distribution. In contrast to PIM, which has no built-in scalability mechanisms, BGP has support for several of them, for example (1) mechanisms to constrain the distribution of routes (e.g. using RT constraints), (2) ability to do route dampening and (3) aggregation of routing information at the route reflector[32].

4.4 Data-plane mechanisms

In the data plane, the NG mVPN solution has two advantages over draft-Rosen:

1. *Automatic tunnel discovery and tunnel binding*. In the draft-Rosen solution, the construction of the MDT relies on manual configuration of the group address. In the NG mVPN solution, tunnel discovery and binding is automatically accomplished using BGP[32].
2. *Support for aggregation*. One of the main drawbacks of the draft-Rosen solution is the lack of support for aggregation, as a given MDT cannot carry traffic of multiple mVPNs. In contrast, in the NG mVPN solution, aggregation can be easily achieved as explained in before. Because the routers in the core of the network participate in the setup of the inter-PE tunnels, lack of support for aggregation increases the amount of both control and data plane state on these routers[32].

4.5 Service provider network as a 'LAN'

The PIM/GRE solution models the interconnect of Virtual Routers present on the PEs as a LAN, which implies that all these Virtual Routers are equidistant from each other, even in the interprovider scenario where PEs servicing a particular mVPN may be part of different service providers[32].

The LAN model reflects rather poorly the underlying service provider infrastructure. In contrast, the NG mVPN solution provides a straightforward way to reflect the underlying service provider infrastructure in its routing decision, as it does not assume that all the PEs are equidistant from each other and can take into account the inter-PE distance in the VPN route selection procedures[32].

4.6 Pro's and con's for NG mVPN and Draft-Rosen implementation

In this part we will summarize different approaches on Multicast VPN implementation - we will compare Draft-Rosen and NG MVPN.

Draft Rosen:

Advantage	Disadvantage
SP core does not need any code support for mVPN, the core routers only need to support regular PIM multicast protocols	Customer multicast traffic is not label switched, the multicast packets are routed using regular IP multicast tunnel in GRE.
SP core is scalable before it does not need to keep multicast states for customers. Instead, all multicast groups for each customer is configure one multicast group in the SP core. This is archive by encapsulating customers' multicast traffic in point to multipoint GRE tunnel between PE (or Multipoint to multipoint GRE tunnels if BiDir-PIM is used in the SP network)	To create provider tunnels there are additional hardware resources needed due to encapsulation/decapsulation. Some vendors require extra hardware modules to accomplish it and anyway it uses extra CPU control power, usually that process is not implemented on line cards.
customer can keep their multicast design without making any changes. They can keep their RP if running PIM-SM.	SP NOC need to look at Multicast RIB and FIB to troubleshoot multicast issue, and look at LFIB to troubleshoot unicast issue. The troubleshooting is splitted and became more complicated.
customer can use any multicast groups as usual, because the customer multicast groups are hidden from the SP core. No problem with overlapping addresses although multicast in SP core doesn't forwarded as MPLS traffic.	Scalability issue - number of PIM states needed to support mVPN infrastructure is grow significally with number of PEs multiply number of mVRFs, they needed full-mesh for every mVPN. This also creates extra control traffic (like PIM hellos).

	Can't using modern traffic protection schemes for multicast like MPLS FRR and node protection

NG MVPN:

Advantage	Disadvantage
Customer multicast traffic is label switched	Complex configuration, needed to detailed understanding of technology to troubleshoot
No need of PIM and no state in the Core. ISP core can be completely free from BGP and PIM	Need to vendor support - technology and signaling usually accomplished on line cards hardware
Can use traffic protection schemes	Convergence time on failure. PIM, used in Draft-Rosen relies on IGP protocols, which are usually converge fast, but NG depends of BGP, which converge much slower. But this can be leveraged in some cases by using RSVP P2MP-FRR. Also this is implied when failed link returns.
BGP Autodiscovery - nodes are	
Can reuse P2MP for different groups - aggregation of traffic for multiple groups can be forwarded single path	
ISP using the same time proven flexible protocol- BGP, that also helps to organize Inter-AS multicast VPNs	
No extra control CPU burden on VPN customer network changes	
Possibility to choose data transport technology. This also help to make smooth migration to NG MVPN	
Can reserve resources in case of RSVP - for multicast traffic can be reserved special bandwidth, it can be important if packet loss/delay/jitter sensitive traffic is forwarded	

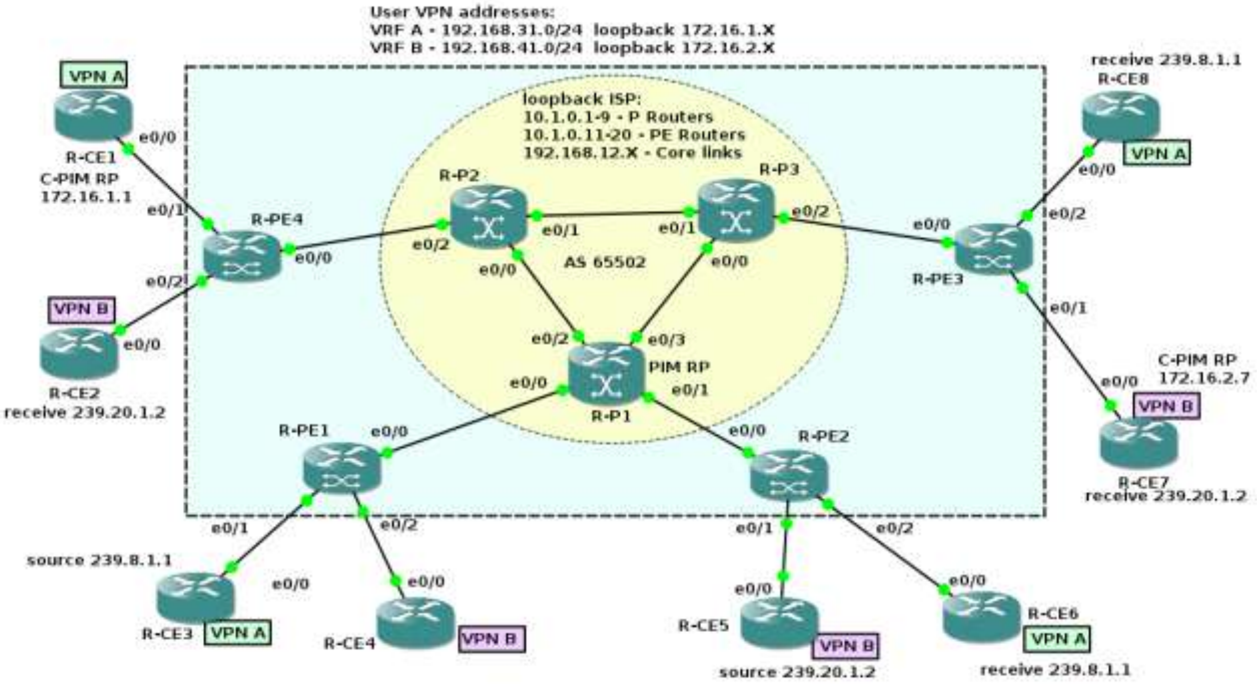
4.7 Summary

Multicast distribution techniques reduce the cost of delivering multi-site, multipoint video and content applications across WAN. It allows carriers to overcome the challenges of delivering high-bandwidth, mission-critical multicast applications in WAN environments. The MVPN service implementation delivers required multicast VPN service attributes such as Privacy, Multi-dimensional Scale, Geographic Reach, Inter-Company Delivery, High Availability, Traffic Engineering, End-to-End Bandwidth Efficiency, Application Assurance and Operational Consistency for next-generation multicast video and content application delivery.

Although the original PIM-based solution described in, offer to carry IP multicast customer traffic over a shared provider infrastructure, this approach departed from the L3VPN unicast model and suffered from scaling limitations both in terms of the maximum number of MVPNs it could support and in terms of the efficiency and intermediate device load which forwards traffic through the provider network. In contrast, the BGP/MPLS-based approach reuses the unicast L3VPN unicast mechanisms with extensions as necessary, thus retaining as much as possible the flexibility and scalability of unicast and use powerful capabilities and leveled approach of IP/MPLS network. However, this scheme requires an extension to the existing Multiprotocol BGP along with advanced operational and troubleshooting knowledge to be deployed in service provider networks[33].

Appendix. Lab demonstration of Multicast VPN (draft Rosen, now RFC 6037).

Lab scenario scene, all routers are Cisco, running IOS version 15.2
This lab network consists of IP/MPLS Network with 2 VPN for Customer A and Customer B.



Main components of this lab scenario:

-routers R-P1, R-P2,R-P3 are MPLS core P-routers, performing only MPLS label switching (LSRs). They run only IGP routing protocol - OSPF single area to exchange internal ISP technical ip-addresses: loopbacks and link subnets. LDP is used for label to FEC mapping.

- routers R-PE1, R-PE2, R-PE3, R-PE4 are edge label routers (PEs). They are boundary routers for MPLS Network, they run OSPF and for exchange external prefixes there is iBGP full-mesh between them.

- other routers: R-CEs (Customer Edge) represent customer network routers, connected to 2 IP/MPLS Layer 3 VPNs: VPN-A and VPN-B. VPN-A used its internal routing protocol - OSPF, also used for PE-CE connectivity, VPN-B uses BGP as PE-CE protocol.

Multicast flow we will organize according the picture: for each VPN we have one source and two receivers:

For VPN A source is R-CE3, group 239.8.1.1 (can be any it is just our case) – receivers – R-CE6, R-CE8
For VPN B source is R-CE5, group 239.20.1.2 (can be any it is just our case) – receivers – R-CE2,
R-CE7
Let's examine the typical configurations of all of these components:

R-CE6 router (OSPF based customer network, VPN A)

```
version 15.2
!
hostname R-CE6
!
ip multicast-routing     # ------- switch on multicast routing, to work in common on multicast traffic
ip cef
!
interface Loopback0
 ip address 172.16.1.6 255.255.255.255
!
interface Ethernet0/0
 ip address 192.168.31.50 255.255.255.240
 ip pim sparse-mode                    # -------------------------- switch on pim sparse mode on interface
!
interface Ethernet0/1
 ip address 192.168.31.129 255.255.255.240   # -- ip address needed to make work PIM and IGMP
 ip pim sparse-mode                                   # ------------- to advertise IGMP subscription to PIM
we need it here
 ip igmp join-group 239.8.1.1  # - IGMP v2 subscription,
request to receive traffic for group 239.8.1.1
!
router ospf 1
 network 172.16.1.6 0.0.0.0 area 0           # ---------------- distribution of loopback address to OSPF
 network 192.168.31.0 0.0.0.255 area 0  # ---------------------- distribution of link address to OSPF
!
ip pim rp-address 172.16.1.1            # ------------  Customer network itself has RP on CE1 it doesn't
know about ISP
!
----------------
```

R-CE2 router (BGP based customer network, VPN B)

```
version 15.2
hostname R-CE2
!
!
ip multicast-routing
ip cef
!
interface Loopback0
 ip address 172.16.2.2 255.255.255.255
!
interface Ethernet0/0
 ip address 192.168.41.2 255.255.255.240
 ip pim sparse--mode
!
router bgp 64500 # -------------------------------- local AS for this Customer is AS64500
 bgp log-neighbor-changes
 neighbor 192.168.41.1 remote-as 65502  # ----------peering with PE router to exchange prefixes
 !
 address-family ipv4                         # --------------------- exchanging only ipv4 prefixes
```

```
  network 192.168.41.0
  redistribute connected
  neighbor 192.168.41.1 activate          # --------------------enable this address for this family
 exit-address-family
!
ip pim rp-address 172.16.2.7              # -------  VPN B has RP at CE7 site
!
----------------------
```

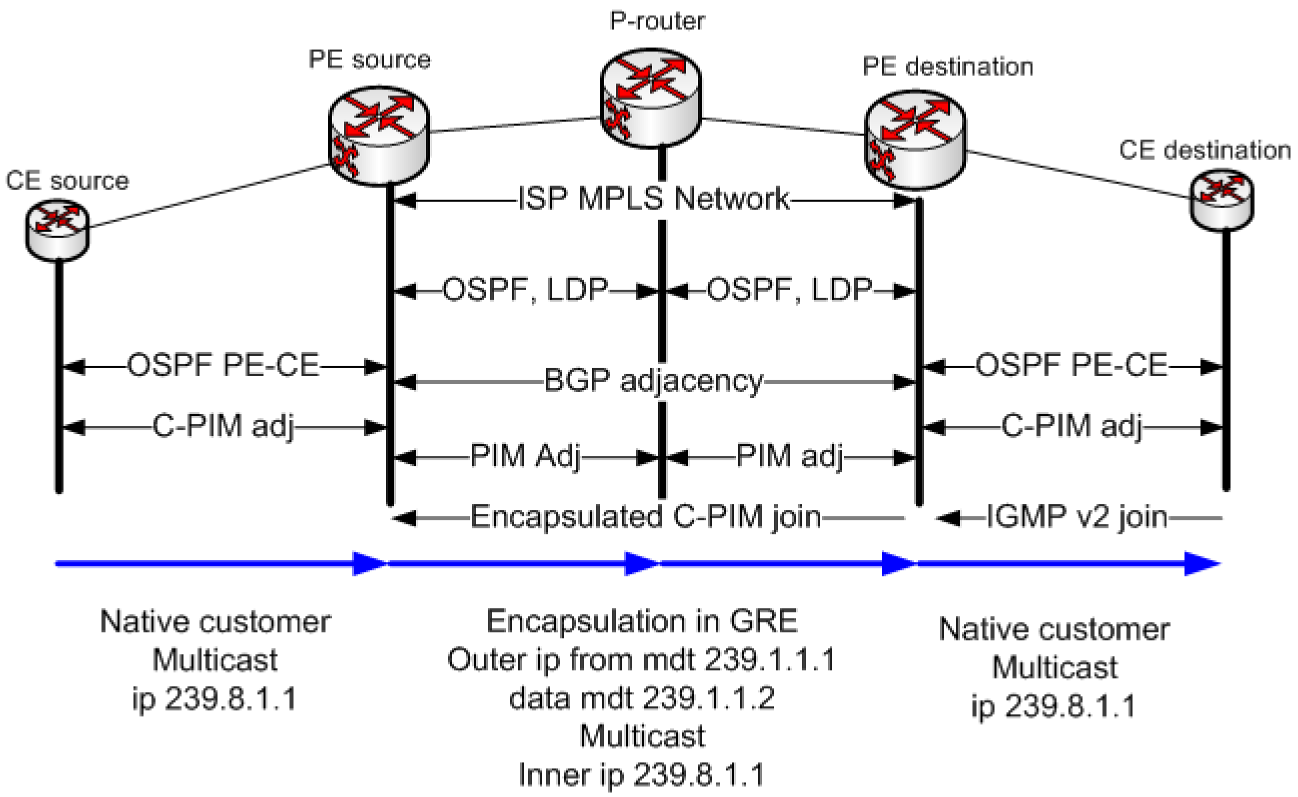
R-PE1 router (PE, MPLS edge providing multicast VPN service)

```
version 15.2
hostname R-PE4
!
vrf definition Vrf-A                          # --------------------- define VPN instance in new style
 rd 65502:1
 !
 address-family ipv4                         # ------------- parameters for this instance for family ipv4
  mdt default 239.1.1.1               # -- define MDT default group address, this creates MTI interface
  mdt data 239.1.1.2 0.0.0.0 threshold 50          # ------------- defining group for data MDT if
  mdt data threshold 50        # ------------- traffic flow exceeds  50kbps it must be switched to it
  route-target export 65502:1   # ------ we will exchange routes only with our VPN, no overlapping
  route-target import 65502:1
 exit-address-family
!
vrf definition Vrf-B                  # -------------- the same as for VPN-A
 rd 65502:2
 !
 address-family ipv4
  mdt default 239.2.2.2                              # -------------- MDT default address for VPN-B
  mdt data 239.2.2.3 0.0.0.0 threshold 50   # ------- MDT data address for VPN-B threshold to
switch also 50kbps
  mdt data threshold 50
  route-target export 65502:2
  route-target import 65502:2
 exit-address-family
!
ip multicast-routing                        # --------enabling multicast routing in common for GRT
ip multicast-routing vrf Vrf-A # --------- enabling multicast routing in common for VPN A
ip multicast-routing vrf Vrf-B # --------- enabling multicast routing in common for VPN A
ip cef
!
!
interface Loopback0
 ip address 10.1.0.14 255.255.255.255
 ip pim sparse-mode          # --------- enabling pim on loopback, for reliability
!
interface Ethernet0/0
 ip address 192.168.12.11 255.255.255.254
 ip pim sparse-mode               # --------- enabling pim towards core to form adjacencies
 ip ospf network point-to-point # ------ set network circuit to PtP because logical nature of this link
 mpls label protocol ldp           # --------- switch on LDP for label mapping
 mpls ip                          # ---------  switch on mpls on this interface
```

```
!
interface Ethernet0/1
 vrf forwarding Vrf-A # --------- this interface belongs to VPN A, customer faced
 ip address 192.168.31.1 255.255.255.240
 ip pim sparse-mode # --------- enable pim sparse mode, this will form C-PIM adjacency PE-CE
!
interface Ethernet0/2
 vrf forwarding Vrf-B   # --------- this interface belongs to VPN A, customer faced
 ip address 192.168.41.1 255.255.255.240
 ip pim sparse-mode
!
router ospf 2 vrf Vrf-A  # ---------  for VPN A  using OSPF IGP this to link with CE1
 redistribute bgp 65502 subnets  # -----   redistribute into OSPF to this site subnets from other sites
 network 192.168.31.0 0.0.0.255 area 0  # --------- mark PE-CE  OSPF interface
!
router ospf 1                   # --------- main OSPF process for ISP network
 mpls traffic-eng area 0        # ------- for MPLS TE, used only for RSVP
 network 10.1.0.14 0.0.0.0 area 0
 network 192.168.12.0 0.0.0.255 area 0
!
router bgp 65502                # ---   main BGP process for IP/MPLS network, full mesh between PEs
 bgp log-neighbor-changes
 neighbor 10.1.0.11 remote-as 65502
 neighbor 10.1.0.11 update-source Loopback0
 neighbor 10.1.0.12 remote-as 65502
 neighbor 10.1.0.12 update-source Loopback0
 neighbor 10.1.0.13 remote-as 65502
 neighbor 10.1.0.13 update-source Loopback0
 !
 address-family ipv4                   # -------  exchanging prefixes in GRT,  ISP internal and external
  redistribute connected
  neighbor 10.1.0.11 activate
  neighbor 10.1.0.11 send-community extended  # ------- this the need to exchange RTs for VPN
  neighbor 10.1.0.11 next-hop-self
  neighbor 10.1.0.11 soft-reconfiguration inbound
  neighbor 10.1.0.12 activate
  neighbor 10.1.0.12 send-community extended
  neighbor 10.1.0.12 next-hop-self
  neighbor 10.1.0.12 soft-reconfiguration inbound
  neighbor 10.1.0.13 activate
  neighbor 10.1.0.13 send-community extended
  neighbor 10.1.0.13 next-hop-self
  neighbor 10.1.0.13 soft-reconfiguration inbound
 exit-address-family
 !
 address-family vpnv4   # ------- exchange of L3VPN prefixes   between PEs
  neighbor 10.1.0.11 activate
  neighbor 10.1.0.11 send-community extended
  neighbor 10.1.0.12 activate
  neighbor 10.1.0.12 send-community extended
  neighbor 10.1.0.13 activate
  neighbor 10.1.0.13 send-community extended
 exit-address-family
 !
 address-family ipv4 mdt  # - mdt family responsible for exchange MDT group addresses for
mVPN
  neighbor 10.1.0.11 activate
  neighbor 10.1.0.11 send-community extended
  neighbor 10.1.0.12 activate
  neighbor 10.1.0.12 send-community extended
  neighbor 10.1.0.13 activate
  neighbor 10.1.0.13 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf Vrf-A        # -- exchange of user prefixes inside VPN – this session is PE-CE
  redistribute connected               # ---- redistribute prefixes for connected subnets in VRF-A
  redistribute static                  # ---- redistribute static routes in VRF-A
  redistribute ospf 2                  # ---- redistribute ospf routes, received from PE-CE session
 exit-address-family
 !
 address-family ipv4 vrf Vrf-B # ------- exchange of user prefixes inside VPN – this session is PE-CE
  redistribute connected
  redistribute static
  neighbor 192.168.41.2 remote-as 64500     # ------- exchange of user prefixes inside VPN – PE-CE
is BGP here
  neighbor 192.168.41.2 activate
  neighbor 192.168.41.2 as-override   # ---- needed to override AS looping because of “PE” crossing
  neighbor 192.168.41.2 soft-reconfiguration inbound
 exit-address-family
!
ip pim rp-address 10.1.0.1               # -------  set PIM RP address statically
ip pim vrf Vrf-A rp-address 172.16.1.1 # -------  to make client PIM connectivity we need to set it
here also
ip pim vrf Vrf-B rp-address 172.16.2.7  # -------  for every VPN, can make automation by Auto-RP
mechanism
!
mpls ldp router-id Loopback0
!
-------------------------------------------+
version 15.2
!
hostname R-P1
!
ip multicast-routing # ------- multicast routing in common
ip cef
!
!
interface Loopback0
 ip address 10.1.0.1 255.255.255.0
 ip pim sparse- mode  # ------- loopback is used for PIM interaction between routers
!
interface Ethernet0/0  # ------- typical mpls interface, MPLS, LDP and PIM for multicast forwarding
```

```
 description to-R-PE1
 ip address 192.168.12.0 255.255.255.254
 ip pim sparse-mode
 ip ospf network point-to-point
 mpls label protocol ldp
 mpls ip
!
interface Ethernet0/1
 description to-R-PE2
 ip address 192.168.12.2 255.255.255.254
 ip pim sparse- mode
 ip ospf network point-to-point
 mpls label protocol ldp
 mpls ip
!
interface Ethernet0/2
 description to-R-P2
 ip address 192.168.12.4 255.255.255.254
 ip pim sparse-mode
 ip ospf network point-to-point
 mpls label protocol ldp
 mpls ip
!
interface Ethernet0/3
 description to-R-P3
 ip address 192.168.12.6 255.255.255.254
 ip pim sparse-mode
 ip ospf network point-to-point
 mpls label protocol ldp
 mpls ip
!
router ospf 1  # ------- here only IGP need to exchange ISP internal information
 network 10.1.0.1 0.0.0.0 area 0
 network 192.168.12.0 0.0.0.255 area 0
!
ip pim rp-address 10.1.0.1  # -- providers PIM RP, in the core need to register MDT group routes
mpls ldp router-id Loopback0
```

Now we will see all the components for multicast forwarding in our lab, the packet flow is shown below:



1) CE1 - this router is RP for Customer A PIM network
Checking connectivity for unicast routers within VPN-A

```
R-CE1#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      172.16.0.0/32 is subnetted, 4 subnets
C        172.16.1.1 is directly connected, Loopback0
O IA     172.16.1.3 [110/21] via 192.168.31.1, 1d07h, Ethernet0/0
O IA     172.16.1.6 [110/21] via 192.168.31.1, 1d06h, Ethernet0/0
O IA     172.16.1.8 [110/21] via 192.168.31.1, 1d07h, Ethernet0/0
      192.168.31.0/24 is variably subnetted, 7 subnets, 2 masks
C        192.168.31.0/28 is directly connected, Ethernet0/0
L        192.168.31.2/32 is directly connected, Ethernet0/0
O IA     192.168.31.16/28 [110/11] via 192.168.31.1, 1d07h, Ethernet0/0
O IA     192.168.31.32/28 [110/11] via 192.168.31.1, 1d07h, Ethernet0/0
O IA     192.168.31.48/28 [110/11] via 192.168.31.1, 1d06h, Ethernet0/0
O IA     192.168.31.128/28 [110/30] via 192.168.31.1, 1d06h, Ethernet0/0
O IA     192.168.31.160/28 [110/30] via 192.168.31.1, 1d06h, Ethernet0/0
```

So we see all routes from other sites - they are seen as OSPF inter-area routes (IA). Is normal for L3VPN, route change type happens when route crosses PE boundary.
Checking how PIM RP working

```
R-CE1#sh ip pim rp
Group: 239.8.1.1, RP: 172.16.1.1, next RP-reachable in 00:00:37
R-CE1#sh ip pim tunnel
Tunnel0
  Type  : PIM Encap
  RP    : 172.16.1.1*
  Source: 192.168.31.2
Tunnel1*
  Type  : PIM Decap
  RP    : 172.16.1.1*
  Source: -
```

Note that PIM tunnels on RP are two types - Encap and Decap. Decap needed only on RP to process multicast signaling join and first packets, non-RPs have only Encap for sending to RP.

```
R-CE1#sh ip pim neighbor
PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      P - Proxy Capable, S - State Refresh Capable, G - GenID Capable
Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
192.168.31.1      Ethernet0/0              1d08h/00:01:25    v2    1 / S P G
```

Look into multicast table and see if any actual traffic coming:

```
R-CE1#sh ip mroute count
Use "show ip mfib count" to get better response time for a large number of mroutes.

IP Multicast Statistics
3 routes using 1564 bytes of memory
2 groups, 0.50 average sources per group
Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kilobits per second
Other counts: Total/RPF failed/Other drops(OIF-null, rate-limit etc)

Group: 239.8.1.1, Source count: 1, Packets forwarded: 586, Packets received: 605
  RP-tree: Forwarding: 586/0/100/0, Other: 586/0/0
  Source: 192.168.31.18/32, Forwarding: 0/0/0/0, Other: 19/0/19

Group: 224.0.1.40, Source count: 0, Packets forwarded: 0, Packets received: 0
```

So we see one service group and our group 239.8.1.1 has source (S,G) and receivers (*,G).

If we look at multicast route table we will see what interfaces packets are flow - here in and out are the same:

```
R-CE1#sh ip mroute
IP Multicast Routing Table
---------cut-----
(*, 239.8.1.1), 1d08h/00:02:58, RP 172.16.1.1, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet0/0, Forward/Sparse, 1d08h/00:02:58

(192.168.31.18, 239.8.1.1), 00:03:00/00:00:03, flags: PT
  Incoming interface: Ethernet0/0, RPF nbr 192.168.31.1
  Outgoing interface list: Null
```

If we will check on CE3 router, which is source will see what there are two responses, because of two receivers:

```
R-CE3#ping 239.8.1.1 repeat 1000000
Type escape sequence to abort.
```

```
Sending 1000000, 100-byte ICMP Echos to 239.8.1.1, timeout is 2 seconds:

Reply to request 0 from 192.168.31.129, 1 ms
Reply to request 0 from 192.168.31.161, 2 ms
Reply to request 1 from 192.168.31.129, 1 ms
Reply to request 1 from 192.168.31.161, 1 ms
-----
```

and look example how RPF check for multicast sources is performed:

```
R-CE6#sh ip rpf 172.16.1.1
RPF information for ? (172.16.1.1)
  RPF interface: Ethernet0/0
  RPF neighbor: ? (192.168.31.49)
  RPF route/mask: 172.16.1.1/32
  RPF type: unicast (ospf 1)
  Doing distance-preferred lookups across tables
  RPF topology: ipv4 multicast base, originated from ipv4 unicast base
```

So multicast packets are flowing and get response from receivers. (Note - the multicast response is used only for debugging purposes it works with *ip igmpjoin-group* cisco IOS command, in production network it can consume too much CPU and not recommended, to statically join group for real networks there is command *ip igmp static-group*).

So - we see what multicast flowing is working, let's move to provider network and see what is happening on PE:

PE1

```
R-PE1#show ip mroute
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 239.1.1.1), 1d08h/stopped, RP 10.1.0.1, flags: SJCFZ
  Incoming interface: Ethernet0/0, RPF nbr 192.168.12.0
  Outgoing interface list:
    MVRF Vrf-A, Forward/Sparse, 1d08h/00:00:40

(10.1.0.12, 239.1.1.1), 1d08h/00:00:56, flags: JTZ
  Incoming interface: Ethernet0/0, RPF nbr 192.168.12.0
  Outgoing interface list:
    MVRF Vrf-A, Forward/Sparse, 1d08h/00:00:40
-----cut-------------
(10.1.0.11, 239.1.1.1), 1d08h/00:03:17, flags: FT
  Incoming interface: Loopback0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet0/0, Forward/Sparse, 1d08h/00:03:14
```

```
(*, 239.2.2.2), 1d08h/stopped, RP 10.1.0.1, flags: SJCFZ
  Incoming interface: Ethernet0/0, RPF nbr 192.168.12.0
  Outgoing interface list:
    MVRF Vrf-B, Forward/Sparse, 1d08h/00:00:40

(10.1.0.13, 239.2.2.2), 1d08h/00:01:00, flags: JTZ
  Incoming interface: Ethernet0/0, RPF nbr 192.168.12.0
  Outgoing interface list:
    MVRF Vrf-B, Forward/Sparse, 1d08h/00:00:40
```

In global table we see routes for MDT default group address 239.1.1.1
Explore PIM core adjacencies:

```
R-PE1#sh ip pim neighbor
PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      P - Proxy Capable, S - State Refresh Capable, G - GenID Capable
Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                             Prio/Mode
192.168.12.0      Ethernet0/0              1d09h/00:01:39    v2    1 / S P G
R-PE1#sh ip pim tunnel
Tunnel0
  Type  : PIM Encap
  RP    : 10.1.0.1
  Source: 192.168.12.1
R-PE1#sh ip pim rp
Group: 239.1.1.1, RP: 10.1.0.1, v2, uptime 1d09h, expires never
Group: 239.2.2.2, RP: 10.1.0.1, v2, uptime 1d09h, expires never
Here we see adjacency to MPLS core, and RPs for MDT groups.
R-PE1#sh ip pim mdt bgp
MDT (Route Distinguisher + IPv4)               Router ID         Next Hop
  MDT group 239.1.1.1
   65502:1:10.1.0.12                           10.1.0.12         10.1.0.12
   65502:1:10.1.0.13                           10.1.0.13         10.1.0.13
   65502:1:10.1.0.14                           10.1.0.14         10.1.0.14
  MDT group 239.2.2.2
   65502:2:10.1.0.12                           10.1.0.12         10.1.0.12
   65502:2:10.1.0.13                           10.1.0.13         10.1.0.13
   65502:2:10.1.0.14                           10.1.0.14         10.1.0.14
R-PE1#sh ip bgp ipv4 mdt all
BGP table version is 15, local router ID is 10.1.0.11
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 65502:1 (default for vrf Vrf-A)
 *>  10.1.0.11/32     0.0.0.0                            0 ?
 *>i 10.1.0.12/32     10.1.0.12                0    100      0 ?
 *>i 10.1.0.13/32     10.1.0.13                0    100      0 ?
 *>i 10.1.0.14/32     10.1.0.14                0    100      0 ?
Route Distinguisher: 65502:2 (default for vrf Vrf-B)
 *>  10.1.0.11/32     0.0.0.0                            0 ?
 *>i 10.1.0.12/32     10.1.0.12                0    100      0 ?
 *>i 10.1.0.13/32     10.1.0.13              0    100      0 ?
 *>i 10.1.0.14/32     10.1.0.14              0    100      0 ?
```

Here we see BGP updates coming between PEs for MDT groups (ipv4 unicast mdt).
And next we will see multicast flows inside VPN A, vrf A

```
R-PE1#show ip mroute vrf Vrf-A
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 239.8.1.1), 00:50:28/stopped, RP 172.16.1.1, flags: SP
  Incoming interface: Tunnel1, RPF nbr 10.1.0.14
  Outgoing interface list: Null

(192.168.31.18, 239.8.1.1), 00:50:28/00:02:49, flags: T
  Incoming interface: Ethernet0/1, RPF nbr 192.168.31.18
  Outgoing interface list:
    Tunnel1, Forward/Sparse, 00:50:28/00:02:31

(*, 224.0.1.40), 1d09h/00:02:55, RP 172.16.1.1, flags: SJPL
  Incoming interface: Tunnel1, RPF nbr 10.1.0.14
  Outgoing interface list: Null
```

Multicast traffic flows from PE-CE interface E0/1, where source CE3 located and going to Tunnel1, which represents MDT default group 239.1.1.1.

Now let's check situation in the core, router R-P1 is the most interesting - more interfaces and it is RP for ISP network.
See what it is pure MPLS LSR and have PIM adjacencies on the same interfaces, no BGP run:

```
R-P1#sh mpls forwarding-table
Local      Outgoing   Prefix           Bytes Label   Outgoing    Next Hop
Label      Label      or Tunnel Id     Switched      interface
20         19         10.1.0.14/32     946434        Et0/2       192.168.12.5
21         20         10.1.0.13/32     607795        Et0/3       192.168.12.7
22         Pop Label  10.1.0.12/32     811299        Et0/1       192.168.12.3
23         Pop Label  10.1.0.11/32     99435860      Et0/0       192.168.12.1
24         No Label   10.1.0.3/32      0             Et0/3       192.168.12.7
25         No Label   10.1.0.2/32      0             Et0/2       192.168.12.5
26         Pop Label  192.168.12.12/31 0             Et0/3       192.168.12.7
27         Pop Label  192.168.12.10/31 0             Et0/2       192.168.12.5
28         Pop Label  192.168.12.8/31  0             Et0/2       192.168.12.5
           Pop Label  192.168.12.8/31  0             Et0/3       192.168.12.7
R-P1#sho bgp summary
% BGP not active

R-P1#sh ip pim neighbor
PIM Neighbor Table
```

```
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      P - Proxy Capable, S - State Refresh Capable, G - GenID Capable
Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                    Prio/Mode
192.168.12.1      Ethernet0/0              1d09h/00:01:32    v2    1 / DR S P G
192.168.12.3      Ethernet0/1              1d08h/00:01:21    v2    1 / DR S P G
192.168.12.5      Ethernet0/2              1d09h/00:01:17    v2    1 / DR S P G
192.168.12.7      Ethernet0/3              1d09h/00:01:42    v2    1 / DR S P G
R-P1#show ip pim tunnel
Tunnel0
  Type  : PIM Encap
  RP    : 10.1.0.1*
  Source: 10.1.0.1
Tunnel1*
  Type  : PIM Decap
  RP    : 10.1.0.1*
  Source: -
R-P1#sh int tunnel 0
Tunnel0 is up, line protocol is up
  Hardware is Tunnel
  Description: Pim Register Tunnel (Encap) for RP 10.1.0.1
  Interface is unnumbered. Using address of Loopback0 (10.1.0.1)
  MTU 17912 bytes, BW 100 Kbit/sec, DLY 50000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 10.1.0.1 (Loopback0), destination 10.1.0.1
   Tunnel Subblocks:
      src-track:
         Tunnel0 source tracking subblock associated with Loopback0
          Set of tunnels with source Loopback0, 2 members (includes iterators), on interface <OK>
  Tunnel protocol/transport PIM/IPv4
  Tunnel TOS/Traffic Class 0xC0,  Tunnel TTL 255
  Tunnel transport MTU 1486 bytes
  Tunnel is transmit only
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
  Last input never, output never, output hang never
  Last clearing of "show interface" counters 1d09h
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
-----cut----------------------------
R-P1#sh int tunnel 1
Tunnel1 is up, line protocol is up
  Hardware is Tunnel
  Description: Pim Register Tunnel (Decap) for RP 10.1.0.1
  Interface is unnumbered. Using address of Loopback0 (10.1.0.1)
  MTU 17920 bytes, BW 100 Kbit/sec, DLY 50000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 10.1.0.1 (Loopback0), destination 10.1.0.1
   Tunnel Subblocks:
      src-track:
         Tunnel1 source tracking subblock associated with Loopback0
          Set of tunnels with source Loopback0, 2 members (includes iterators), on interface <OK>
  Tunnel protocol/transport PIM/IPv4
  Tunnel TTL 255
  Tunnel transport MTU 1494 bytes
  Tunnel is receive only
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
  Last input never, output never, output hang never
  Last clearing of "show interface" counters 1d09h
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
```

See what is in multicast route table:

```
R-P1#show ip mroute
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 239.1.1.1), 1d09h/00:03:21, RP 10.1.0.1, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet0/2, Forward/Sparse, 1d09h/00:02:34
    Ethernet0/0, Forward/Sparse, 1d09h/00:02:36
    Ethernet0/1, Forward/Sparse, 1d09h/00:03:21
    Ethernet0/3, Forward/Sparse, 1d09h/00:03:13

(10.1.0.11, 239.1.1.1), 1d09h/00:02:49, flags: T
  Incoming interface: Ethernet0/0, RPF nbr 192.168.12.1
  Outgoing interface list:
    Ethernet0/3, Forward/Sparse, 1d09h/00:03:13
    Ethernet0/1, Forward/Sparse, 1d09h/00:03:21
    Ethernet0/2, Forward/Sparse, 1d09h/00:03:06

(10.1.0.14, 239.1.1.1), 1d09h/00:01:49, flags: T
  Incoming interface: Ethernet0/2, RPF nbr 192.168.12.5
  Outgoing interface list:
    Ethernet0/1, Forward/Sparse, 1d09h/00:03:21
    Ethernet0/0, Forward/Sparse, 1d09h/00:02:57
```

You can see what there are only MDT default groups, pointing to interfaces towards corresponding PEs.

Now just check on R-PE3, which have receivers what packet flow is coming to R-CE8.

```
R-PE3#show ip mroute vrf Vrf-A count
Use "show ip mfib count" to get better response time for a large number of mroutes.

IP Multicast Statistics
3 routes using 1710 bytes of memory
2 groups, 0.50 average sources per group
Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kilobits per second
```

```
Other counts: Total/RPF failed/Other drops(OIF-null, rate-limit etc)

Group: 239.8.1.1, Source count: 1, Packets forwarded: 2009, Packets received: 2009
  RP-tree: Forwarding: 2/0/100/0, Other: 2/0/0
  Source: 192.168.31.18/32, Forwarding: 2007/0/100/0, Other: 2007/0/0

Group: 224.0.1.40, Source count: 0, Packets forwarded: 0, Packets received: 0
R-PE3#show ip mroute vrf Vrf-A
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 239.8.1.1), 1d08h/00:03:14, RP 172.16.1.1, flags: S
  Incoming interface: Tunnel1, RPF nbr 10.1.0.14
  Outgoing interface list:
    Ethernet0/2, Forward/Sparse, 1d08h/00:03:14

(192.168.31.18, 239.8.1.1), 01:07:27/00:01:47, flags: T
  Incoming interface: Tunnel1, RPF nbr 10.1.0.11
  Outgoing interface list:
    Ethernet0/2, Forward/Sparse, 01:07:27/00:03:14
```

And now we will change the situation to investigate what will happen if traffic flow will grow above MDT data threshold - change the ping parameters on CE3 source:

```
R-CE3# ping 239.8.1.1 repeat 1000000 size 1200 timeout 0
```

Checking on PE1:

```
R-PE1#sh int e0/1
Ethernet0/1 is up, line protocol is up
  Hardware is AmdP2, address is aabb.cc00.0110 (bia aabb.cc00.0110)
  Internet address is 192.168.31.17/28
  MTU 1500 bytes, BW 10000 Kbit/sec, DLY 1000 usec,
     reliability 255/255, txload 250/255, rxload 125/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:07, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 4927000 bits/sec, 520 packets/sec
  5 minute output rate 9828000 bits/sec, 1025 packets/sec
     339042 packets input, 303716763 bytes, 0 no buffer
     Received 19590 broadcasts (316442 IP multicasts)
R-PE1#show ip pim vrf Vrf-A mdt send
MDT-data send list for VRF: Vrf-A
```

```
  (source, group)                   MDT-data group/num   ref_count
  (192.168.31.18, 239.8.1.1)          239.1.1.2            1
```

We see what MDT data group 239.1.1.2 has been activated for VPN A.

```
R-PE3#sh ip pim vrf Vrf-A mdt receive

Joined MDT-data [group/mdt number : source]  uptime/expires for VRF: Vrf-A
 [239.1.1.2 : 0.0.0.0]  00:04:54/00:02:05
R-PE3#sh ip mroute vrf Vrf-A active
Use "show ip mfib active" to get better response time for a large number of mroutes.

Active IP Multicast Sources - sending >= 4 kbps

Group: 239.8.1.1, (?)
   Source: 192.168.31.18 (?)
     Rate: 1102 pps/10582 kbps(1sec), 10582 kbps(last 50 secs), 542 kbps(life avg)
```

So we see that MDT data group has been activating and multicast forwarding using it to deliver packets to subscribers.

Appendix 2. router configurations for Draft-Rosen implementation

R-CE1

```
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R-CE1
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
clock timezone MSK 4 0
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
no ip icmp rate-limit unreachable
ip auth-proxy max-login-attempts 5
ip admission max-login-attempts 5
!
no ip domain lookup
ip multicast-routing
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
!
redundancy
!
ip tcp synwait-time 5
!
interface Loopback0
 ip address 172.16.1.1 255.255.255.255
!
interface Ethernet0/0
 ip address 192.168.31.2 255.255.255.240
 ip pim sparse-mode
!
router ospf 1
 network 172.16.1.1 0.0.0.0 area 0
 network 192.168.31.0 0.0.0.255 area 0
 default-information originate always metric 23040
!
no ip http server
no ip http secure-server
ip pim rp-address 172.16.1.1
!
!
control-plane
```

```
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
 transport input all
!
end
```

R-CE2

```
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R-CE2
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
clock timezone MSK 4 0
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
no ip icmp rate-limit unreachable
ip auth-proxy max-login-attempts 5
ip admission max-login-attempts 5
!
no ip domain lookup
ip multicast-routing
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
!
redundancy
!
ip tcp synwait-time 5
!
interface Loopback0
 ip address 172.16.2.2 255.255.255.255
!
interface Ethernet0/0
```

```
 ip address 192.168.41.2 255.255.255.240
 ip pim sparse-mode
!
router bgp 64500
 bgp log-neighbor-changes
 neighbor 192.168.41.1 remote-as 65502
 !
 address-family ipv4
  network 192.168.41.0
  redistribute connected
  neighbor 192.168.41.1 activate
 exit-address-family
!
ip pim rp-address 172.16.2.7
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
control-plane
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
 transport input all
!
end
```

R-CE3

```
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R-CE3
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
clock timezone MSK 4 0
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
no ip icmp rate-limit unreachable
ip auth-proxy max-login-attempts 5
ip admission max-login-attempts 5
!
no ip domain lookup
ip multicast-routing
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
!
redundancy
!
ip tcp synwait-time 5
!
interface Loopback0
 ip address 172.16.1.3 255.255.255.255
!
interface Ethernet0/0
 ip address 192.168.31.18 255.255.255.240
 ip pim sparse-mode
!
router ospf 1
 network 172.16.1.3 0.0.0.0 area 0
 network 192.168.31.0 0.0.0.255 area 0
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
ip pim rp-address 172.16.1.1
!
control-plane
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
 transport input all
!
end
```

R-CE4

```
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R-CE4
!
boot-start-marker
boot-end-marker
!
logging discriminator EXCESS severity drops 6 msg-body drops EXCESSCOLL
logging buffered 50000
logging console discriminator EXCESS
!
no aaa new-model
!
clock timezone MSK 4 0
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
no ip icmp rate-limit unreachable
ip auth-proxy max-login-attempts 5
ip admission max-login-attempts 5
!
no ip domain lookup
ip multicast-routing
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
!
crypto pki token default removal timeout 0
!
redundancy
!
ip tcp synwait-time 5
!
interface Loopback0
 ip address 172.16.2.4 255.255.255.255
!
interface Ethernet0/0
 ip address 192.168.41.18 255.255.255.240
 ip pim sparse-mode
!
interface Ethernet0/1
 ip address 188.2.2.2 255.255.255.252
 ip pim sparse-mode
 ip igmp join-group 239.20.1.2
!
router bgp 64500
```

```
 bgp log-neighbor-changes
 neighbor 192.168.41.17 remote-as 65502
 !
 address-family ipv4
  network 192.168.41.0
  redistribute connected
  neighbor 192.168.41.17 activate
 exit-address-family
!
 ip forward-protocol nd
!
no ip http server
no ip http secure-server
ip pim rp-address 172.16.2.7
!
control-plane
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
 transport input all
!
end
```

R-CE5

```
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R-CE5
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
clock timezone MSK 4 0
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
no ip icmp rate-limit unreachable
ip auth-proxy max-login-attempts 5
ip admission max-login-attempts 5
!
no ip domain lookup
ip multicast-routing
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
!
redundancy
!
ip tcp synwait-time 5
!
interface Loopback0
 ip address 172.16.2.5 255.255.255.255
!
interface Ethernet0/0
 ip address 192.168.41.50 255.255.255.240
 ip pim sparse-mode
!
router bgp 64500
 bgp log-neighbor-changes
 neighbor 192.168.41.49 remote-as 65502
 !
 address-family ipv4
  network 192.168.41.0
  redistribute connected
  neighbor 192.168.41.49 activate
 exit-address-family
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
ip pim rp-address 172.16.2.7
!
control-plane
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
 transport input all
!
end
```

R-CE6

```
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R-CE6
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
clock timezone MSK 4 0
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
no ip icmp rate-limit unreachable
ip auth-proxy max-login-attempts 5
ip admission max-login-attempts 5
!
no ip domain lookup
ip multicast-routing
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
!
redundancy
!
ip tcp synwait-time 5
!
interface Loopback0
 ip address 172.16.1.6 255.255.255.255
!
interface Ethernet0/0
 ip address 192.168.31.50 255.255.255.240
 ip pim sparse-mode
!
interface Ethernet0/1
 ip address 186.1.1.1 255.255.255.252
 ip pim sparse-mode
 ip igmp join-group 239.8.1.1
!
router ospf 1
 network 172.16.1.6 0.0.0.0 area 0
 network 192.168.31.0 0.0.0.255 area 0
!
ip forward-protocol nd
!
```

```
no ip http server
no ip http secure-server
ip pim rp-address 172.16.1.1
!
control-plane
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
 transport input all
!
end
```

R-CE7

```
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R-CE7
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
clock timezone MSK 4 0
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
no ip icmp rate-limit unreachable
ip auth-proxy max-login-attempts 5
ip admission max-login-attempts 5
!
no ip domain lookup
ip multicast-routing
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
!
redundancy
!
ip tcp synwait-time 5
```

```
!
interface Loopback0
 ip address 172.16.2.7 255.255.255.255
!
interface Ethernet0/0
 ip address 192.168.41.34 255.255.255.240
 ip pim sparse-mode
!
interface Ethernet0/1
 ip address 188.1.1.1 255.255.255.252
 ip pim sparse-mode
 ip igmp join-group 239.20.1.2
!
router ospf 1
 network 172.16.2.7 0.0.0.0 area 0
 network 192.168.41.0 0.0.0.255 area 0
!
router bgp 64500
 bgp log-neighbor-changes
 neighbor 192.168.41.33 remote-as 65502
 !
 address-family ipv4
  network 192.168.41.0
  redistribute connected
  neighbor 192.168.41.33 activate
 exit-address-family
!
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
ip pim rp-address 172.16.2.7
!
control-plane
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
 transport input all
!
end
```

R-CE8

```
version 15.2
```

```
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R-CE8
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
clock timezone MSK 4 0
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
no ip icmp rate-limit unreachable
ip auth-proxy max-login-attempts 5
ip admission max-login-attempts 5
!
no ip domain lookup
ip multicast-routing
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
!
redundancy
!
ip tcp synwait-time 5
!
interface Loopback0
 ip address 172.16.1.8 255.255.255.255
!
interface Ethernet0/0
 ip address 192.168.31.34 255.255.255.240
 ip pim sparse-mode
!
interface Ethernet0/1
 ip address 192.168.31.161 255.255.255.240
 ip pim sparse-mode
 ip igmp join-group 239.8.1.1
!
router ospf 1
 network 172.16.1.8 0.0.0.0 area 0
 network 192.168.31.0 0.0.0.255 area 0
!
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
ip pim rp-address 172.16.1.1
!
control-plane
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
 transport input all
!
end
```

R-PE1

```
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R-PE1
!
boot-start-marker
boot-end-marker
!
vrf definition Vrf-A
 rd 65502:1
 !
 address-family ipv4
  mdt default 239.1.1.1
  mdt data 239.1.1.2 0.0.0.0 threshold 50
  mdt data threshold 50
  route-target export 65502:1
  route-target import 65502:1
 exit-address-family
!
vrf definition Vrf-B
 rd 65502:2
 !
 address-family ipv4
  mdt default 239.2.2.2
  mdt data 239.2.2.3 0.0.0.0 threshold 50
  mdt data threshold 50
  route-target export 65502:2
  route-target import 65502:2
 exit-address-family
!
```

```
no aaa new-model
!
clock timezone MSK 4 0
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
no ip icmp rate-limit unreachable
ip auth-proxy max-login-attempts 5
ip admission max-login-attempts 5
!
no ip domain lookup
ip multicast-routing
ip multicast-routing vrf Vrf-A
ip multicast-routing vrf Vrf-B
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
!
redundancy
!
ip tcp synwait-time 5
!
interface Loopback0
 ip address 10.1.0.11 255.255.255.255
 ip pim sparse-mode
!
interface Ethernet0/0
 ip address 192.168.12.1 255.255.255.254
 ip pim sparse-mode
 ip ospf network point-to-point
 mpls label protocol ldp
 mpls ip
!
interface Ethernet0/1
 vrf forwarding Vrf-A
 ip address 192.168.31.17 255.255.255.240
 ip pim sparse-mode
!
interface Ethernet0/2
 vrf forwarding Vrf-B
 ip address 192.168.41.17 255.255.255.240
 ip pim sparse-mode
!
router ospf 2 vrf Vrf-A
 redistribute bgp 65502 subnets
 network 192.168.31.0 0.0.0.255 area 0
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 network 10.1.0.11 0.0.0.0 area 0
 network 192.168.12.0 0.0.0.255 area 0
!
router bgp 65502
 bgp log-neighbor-changes
 neighbor 10.1.0.12 remote-as 65502
 neighbor 10.1.0.12 update-source Loopback0
 neighbor 10.1.0.13 remote-as 65502
 neighbor 10.1.0.13 update-source Loopback0
 neighbor 10.1.0.14 remote-as 65502
 neighbor 10.1.0.14 update-source Loopback0
 !
 address-family ipv4
  redistribute connected
  neighbor 10.1.0.12 activate
  neighbor 10.1.0.12 send-community extended
  neighbor 10.1.0.12 next-hop-self
  neighbor 10.1.0.13 activate
  neighbor 10.1.0.13 send-community extended
  neighbor 10.1.0.13 next-hop-self
  neighbor 10.1.0.13 soft-reconfiguration inbound
  neighbor 10.1.0.14 activate
  neighbor 10.1.0.14 send-community extended
  neighbor 10.1.0.14 next-hop-self
  neighbor 10.1.0.14 soft-reconfiguration inbound
 exit-address-family
 !
 address-family vpnv4
  neighbor 10.1.0.12 activate
  neighbor 10.1.0.12 send-community extended
  neighbor 10.1.0.13 activate
  neighbor 10.1.0.13 send-community extended
  neighbor 10.1.0.14 activate
  neighbor 10.1.0.14 send-community extended
 exit-address-family
 !
 address-family ipv4 mdt
  neighbor 10.1.0.12 activate
  neighbor 10.1.0.12 send-community extended
  neighbor 10.1.0.13 activate
  neighbor 10.1.0.13 send-community extended
  neighbor 10.1.0.14 activate
  neighbor 10.1.0.14 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf Vrf-A
  redistribute connected
  redistribute static
  redistribute ospf 2
 exit-address-family
 !
 address-family ipv4 vrf Vrf-B
  redistribute connected
  redistribute static
  neighbor 192.168.41.18 remote-as 64500
  neighbor 192.168.41.18 activate
  neighbor 192.168.41.18 as-override
  neighbor 192.168.41.18 soft-reconfiguration inbound
 exit-address-family
!
ip forward-protocol nd
no ip http server
no ip http secure-server
ip pim rp-address 10.1.0.1
ip pim vrf Vrf-A rp-address 172.16.1.1
ip pim vrf Vrf-B rp-address 172.16.2.7
!
mpls ldp router-id Loopback0
!
control-plane
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
 transport input all
!
end
```

R-PE2

```
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R-PE2
!
boot-start-marker
boot-end-marker
!
vrf definition Vrf-A
 rd 65502:1
 !
 address-family ipv4
  mdt default 239.1.1.1
  mdt data 239.1.1.2 0.0.0.0 threshold 50
  mdt data threshold 50
  route-target export 65502:1
  route-target import 65502:1
 exit-address-family
!
vrf definition Vrf-B
 rd 65502:2
 !
 address-family ipv4
  mdt default 239.2.2.2
  mdt data 239.2.2.3 0.0.0.0 threshold 50
  mdt data threshold 50
  route-target export 65502:2
  route-target import 65502:2
 exit-address-family
!
no aaa new-model
!
clock timezone MSK 4 0
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
no ip icmp rate-limit unreachable
ip auth-proxy max-login-attempts 5
ip admission max-login-attempts 5
!
no ip domain lookup
ip multicast-routing
ip multicast-routing vrf Vrf-A
ip multicast-routing vrf Vrf-B
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
!
redundancy
!
ip tcp synwait-time 5
!
interface Loopback0
 ip address 10.1.0.12 255.255.255.255
 ip pim sparse-mode
!
interface Ethernet0/0
 ip address 192.168.12.3 255.255.255.254
 ip pim sparse-mode
 ip ospf network point-to-point
 mpls label protocol ldp
 mpls ip
!
interface Ethernet0/1
```

```
 vrf forwarding Vrf-B
 ip address 192.168.41.49 255.255.255.240
 ip pim sparse-mode
!
interface Ethernet0/2
 vrf forwarding Vrf-A
 ip address 192.168.31.49 255.255.255.240
 ip pim sparse-mode
!
router ospf 2 vrf Vrf-A
 redistribute bgp 65502 subnets
 network 192.168.31.0 0.0.0.255 area 0
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 network 10.1.0.12 0.0.0.0 area 0
 network 192.168.12.0 0.0.0.255 area 0
 network 192.168.13.0 0.0.0.15 area 0
!
router bgp 65502
 bgp log-neighbor-changes
 neighbor 10.1.0.11 remote-as 65502
 neighbor 10.1.0.11 update-source Loopback0
 neighbor 10.1.0.13 remote-as 65502
 neighbor 10.1.0.13 update-source Loopback0
 neighbor 10.1.0.14 remote-as 65502
 neighbor 10.1.0.14 update-source Loopback0
 !
 address-family ipv4
  redistribute connected
  neighbor 10.1.0.11 activate
  neighbor 10.1.0.11 send-community extended
  neighbor 10.1.0.11 next-hop-self
  neighbor 10.1.0.11 soft-reconfiguration inbound
  neighbor 10.1.0.13 activate
  neighbor 10.1.0.13 send-community extended
  neighbor 10.1.0.13 next-hop-self
  neighbor 10.1.0.13 soft-reconfiguration inbound
  neighbor 10.1.0.14 activate
  neighbor 10.1.0.14 send-community extended
  neighbor 10.1.0.14 next-hop-self
  neighbor 10.1.0.14 soft-reconfiguration inbound
 exit-address-family
 !
 address-family vpnv4
  neighbor 10.1.0.11 activate
  neighbor 10.1.0.11 send-community extended
  neighbor 10.1.0.13 activate
  neighbor 10.1.0.13 send-community extended
  neighbor 10.1.0.14 activate
  neighbor 10.1.0.14 send-community extended
 exit-address-family
 !
 address-family ipv4 mdt
  neighbor 10.1.0.11 activate
  neighbor 10.1.0.11 send-community extended
  neighbor 10.1.0.13 activate
  neighbor 10.1.0.13 send-community extended
  neighbor 10.1.0.14 activate
  neighbor 10.1.0.14 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf Vrf-A
  redistribute connected
  redistribute static
  redistribute ospf 2
 exit-address-family
 !
address-family ipv4 vrf Vrf-B
  redistribute connected
  redistribute static
  neighbor 192.168.41.50 remote-as 64500
  neighbor 192.168.41.50 activate
  neighbor 192.168.41.50 as-override
  neighbor 192.168.41.50 soft-reconfiguration inbound
 exit-address-family
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
ip pim rp-address 10.1.0.1
ip pim vrf Vrf-A rp-address 172.16.1.1
ip pim vrf Vrf-B rp-address 172.16.2.7
!
mpls ldp router-id Loopback0
!
control-plane
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
 transport input all
!
end
```

R-PE3

```
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R-PE3
!
boot-start-marker
boot-end-marker
!
vrf definition Vrf-A
 rd 65502:1
 !
 address-family ipv4
  mdt default 239.1.1.1
  mdt data 239.1.1.2 0.0.0.0 threshold 50
  mdt data threshold 50
  route-target export 65502:1
  route-target import 65502:1
 exit-address-family
!
vrf definition Vrf-B
 rd 65502:2
 !
 address-family ipv4
  mdt default 239.2.2.2
  mdt data 239.2.2.3 0.0.0.0 threshold 50
  mdt data threshold 50
  route-target export 65502:2
  route-target import 65502:2
 exit-address-family
!
no aaa new-model
!
clock timezone MSK 4 0
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
no ip icmp rate-limit unreachable
ip auth-proxy max-login-attempts 5
ip admission max-login-attempts 5
!
no ip domain lookup
ip multicast-routing
ip multicast-routing vrf Vrf-A
ip multicast-routing vrf Vrf-B
ip cef
no ipv6 cef
!
```

```
multilink bundle-name authenticated
!
redundancy
!
ip tcp synwait-time 5
interface Loopback0
 ip address 10.1.0.13 255.255.255.255
 ip pim sparse-mode
!
interface Ethernet0/0
 ip address 192.168.12.13 255.255.255.254
 ip pim sparse-mode
 ip ospf network point-to-point
 mpls label protocol ldp
 mpls ip
!
interface Ethernet0/1
 vrf forwarding Vrf-B
 ip address 192.168.41.33 255.255.255.240
 ip pim sparse-mode
!
interface Ethernet0/2
 vrf forwarding Vrf-A
 ip address 192.168.31.33 255.255.255.240
 ip pim sparse-mode
!
router ospf 2 vrf Vrf-A
 redistribute bgp 65502 subnets
 network 192.168.31.0 0.0.0.255 area 0
!
router ospf 1
 network 10.1.0.13 0.0.0.0 area 0
 network 192.168.12.0 0.0.0.255 area 0
!
router bgp 65502
 bgp log-neighbor-changes
 neighbor 10.1.0.11 remote-as 65502
 neighbor 10.1.0.11 update-source Loopback0
 neighbor 10.1.0.12 remote-as 65502
 neighbor 10.1.0.12 update-source Loopback0
 neighbor 10.1.0.14 remote-as 65502
 neighbor 10.1.0.14 update-source Loopback0
 !
 address-family ipv4
  redistribute connected
  neighbor 10.1.0.11 activate
  neighbor 10.1.0.11 send-community extended
  neighbor 10.1.0.11 next-hop-self
  neighbor 10.1.0.11 soft-reconfiguration inbound
  neighbor 10.1.0.12 activate
  neighbor 10.1.0.12 send-community extended
  neighbor 10.1.0.12 next-hop-self
  neighbor 10.1.0.12 soft-reconfiguration inbound
  neighbor 10.1.0.14 activate
  neighbor 10.1.0.14 send-community extended
  neighbor 10.1.0.14 next-hop-self
  neighbor 10.1.0.14 soft-reconfiguration inbound
 exit-address-family
 !
 address-family vpnv4
  neighbor 10.1.0.11 activate
  neighbor 10.1.0.11 send-community extended
  neighbor 10.1.0.12 activate
  neighbor 10.1.0.12 send-community extended
  neighbor 10.1.0.14 activate
  neighbor 10.1.0.14 send-community extended
 exit-address-family
 !
 address-family ipv4 mdt
  neighbor 10.1.0.11 activate
  neighbor 10.1.0.11 send-community extended
  neighbor 10.1.0.12 activate
  neighbor 10.1.0.12 send-community extended
  neighbor 10.1.0.14 activate
  neighbor 10.1.0.14 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf Vrf-A
  redistribute connected
  redistribute static
  redistribute ospf 2
 exit-address-family
 !
 address-family ipv4 vrf Vrf-B
  redistribute connected
  redistribute static
  neighbor 192.168.41.34 remote-as 64500
  neighbor 192.168.41.34 activate
  neighbor 192.168.41.34 as-override
  neighbor 192.168.41.34 soft-reconfiguration inbound
 exit-address-family
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
ip pim rp-address 10.1.0.1
ip pim vrf Vrf-A rp-address 172.16.1.1
ip pim vrf Vrf-B rp-address 172.16.2.7
!
mpls ldp router-id Loopback0
!
```

```
control-plane
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
 transport input all
!
end
```

R-PE4

```
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R-PE4
!
boot-start-marker
boot-end-marker
!
!
vrf definition Vrf-A
 rd 65502:1
 !
 address-family ipv4
  mdt default 239.1.1.1
  mdt data 239.1.1.2 0.0.0.0 threshold 50
  mdt data threshold 50
  route-target export 65502:1
  route-target import 65502:1
 exit-address-family
!
vrf definition Vrf-B
 rd 65502:2
 !
 address-family ipv4
  mdt default 239.2.2.2
  mdt data 239.2.2.3 0.0.0.0 threshold 50
  mdt data threshold 50
  route-target export 65502:2
  route-target import 65502:2
 exit-address-family
!
no aaa new-model
```

```
!
clock timezone MSK 4 0
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
no ip icmp rate-limit unreachable
ip auth-proxy max-login-attempts 5
ip admission max-login-attempts 5
!
no ip domain lookup
ip multicast-routing
ip multicast-routing vrf Vrf-A
ip multicast-routing vrf Vrf-B
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
!
redundancy
!
ip tcp synwait-time 5
!
interface Loopback0
 ip address 10.1.0.14 255.255.255.255
 ip pim sparse-mode
!
interface Ethernet0/0
 ip address 192.168.12.11 255.255.255.254
 ip pim sparse-mode
 ip ospf network point-to-point
 mpls label protocol ldp
 mpls ip
!
interface Ethernet0/1
 vrf forwarding Vrf-A
 ip address 192.168.31.1 255.255.255.240
 ip pim sparse-mode
!
interface Ethernet0/2
 vrf forwarding Vrf-B
 ip address 192.168.41.1 255.255.255.240
 ip pim sparse-mode
!
router ospf 2 vrf Vrf-A
 redistribute bgp 65502 subnets
 network 192.168.31.0 0.0.0.255 area 0
!
router ospf 1
 mpls traffic-eng area 0
 network 10.1.0.14 0.0.0.0 area 0
 network 192.168.12.0 0.0.0.255 area 0
!
router bgp 65502
 bgp log-neighbor-changes
 neighbor 10.1.0.11 remote-as 65502
 neighbor 10.1.0.11 update-source Loopback0
 neighbor 10.1.0.12 remote-as 65502
 neighbor 10.1.0.12 update-source Loopback0
 neighbor 10.1.0.13 remote-as 65502
 neighbor 10.1.0.13 update-source Loopback0
 !
 address-family ipv4
  redistribute connected
  neighbor 10.1.0.11 activate
  neighbor 10.1.0.11 send-community extended
  neighbor 10.1.0.11 next-hop-self
  neighbor 10.1.0.11 soft-reconfiguration inbound
  neighbor 10.1.0.12 activate
  neighbor 10.1.0.12 send-community extended
  neighbor 10.1.0.12 next-hop-self
  neighbor 10.1.0.12 soft-reconfiguration inbound
  neighbor 10.1.0.13 activate
  neighbor 10.1.0.13 send-community extended
  neighbor 10.1.0.13 next-hop-self
  neighbor 10.1.0.13 soft-reconfiguration inbound
 exit-address-family
 !
 address-family vpnv4
  neighbor 10.1.0.11 activate
  neighbor 10.1.0.11 send-community extended
  neighbor 10.1.0.12 activate
  neighbor 10.1.0.12 send-community extended
  neighbor 10.1.0.13 activate
  neighbor 10.1.0.13 send-community extended
 exit-address-family
 !
 address-family ipv4 mdt
  neighbor 10.1.0.11 activate
  neighbor 10.1.0.11 send-community extended
  neighbor 10.1.0.12 activate
  neighbor 10.1.0.12 send-community extended
  neighbor 10.1.0.13 activate
  neighbor 10.1.0.13 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf Vrf-A
  redistribute connected
  redistribute static
  redistribute ospf 2
 exit-address-family
address-family ipv4 vrf Vrf-B
```

```
  redistribute connected
  redistribute static
  neighbor 192.168.41.2 remote-as 64500
  neighbor 192.168.41.2 activate
  neighbor 192.168.41.2 as-override
  neighbor 192.168.41.2 soft-reconfiguration inbound
 exit-address-family
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
ip pim rp-address 10.1.0.1
ip pim vrf Vrf-A rp-address 172.16.1.1
ip pim vrf Vrf-B rp-address 172.16.2.7
!
mpls ldp router-id Loopback0
!
control-plane
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
 transport input all
!
end
```

R-P1

```
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R-P1
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
clock timezone MSK 4 0
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
```

```
no ip icmp rate-limit unreachable
ip auth-proxy max-login-attempts 5
ip admission max-login-attempts 5
!
no ip domain lookup
ip multicast-routing
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
!
redundancy
!
ip tcp synwait-time 5
!
interface Loopback0
 ip address 10.1.0.1 255.255.255.0
 ip pim sparse-mode
!
interface Ethernet0/0
 description to-R-PE1
 ip address 192.168.12.0 255.255.255.254
 ip pim sparse-mode
 ip ospf network point-to-point
 mpls label protocol ldp
 mpls ip
!
interface Ethernet0/1
 description to-R-PE2
 ip address 192.168.12.2 255.255.255.254
 ip pim sparse-mode
 ip ospf network point-to-point
 mpls label protocol ldp
 mpls ip
!
interface Ethernet0/2
 description to-R-P2
 ip address 192.168.12.4 255.255.255.254
 ip pim sparse-mode
 ip ospf network point-to-point
 mpls label protocol ldp
 mpls ip
!
interface Ethernet0/3
 description to-R-P3
 ip address 192.168.12.6 255.255.255.254
 ip pim sparse-mode
 ip ospf network point-to-point
 mpls label protocol ldp
 mpls ip
!
```

```
router ospf 1
 network 10.1.0.1 0.0.0.0 area 0
 network 192.168.12.0 0.0.0.255 area 0
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
ip pim rp-address 10.1.0.1
!
mpls ldp router-id Loopback0
!
control-plane
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
 transport input all
!
end
```

R-P2

```
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R-P2
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
clock timezone MSK 4 0
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
no ip icmp rate-limit unreachable
ip auth-proxy max-login-attempts 5
ip admission max-login-attempts 5
!
no ip domain lookup
```

```
ip multicast-routing
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
!
redundancy
!
ip tcp synwait-time 5
!
interface Loopback0
 ip address 10.1.0.2 255.255.255.0
 ip pim sparse-mode
!
interface Ethernet0/0
 description to-R-P1
 ip address 192.168.12.5 255.255.255.254
 ip pim sparse-mode
 ip ospf network point-to-point
 mpls label protocol ldp
 mpls ip
!
interface Ethernet0/1
 description to-R-P3
 ip address 192.168.12.8 255.255.255.254
 ip pim sparse-mode
 ip ospf network point-to-point
 mpls label protocol ldp
 mpls ip
!
interface Ethernet0/2
 description to-R-PE4
 ip address 192.168.12.10 255.255.255.254
 ip pim sparse-mode
 ip ospf network point-to-point
 mpls label protocol ldp
 mpls ip
!
router ospf 1
 network 10.1.0.2 0.0.0.0 area 0
 network 192.168.12.0 0.0.0.255 area 0
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
ip pim rp-address 10.1.0.1
!
mpls ldp router-id Loopback0
!
control-plane
```

```
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
 transport input all
!
end
```

R-P3

```
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R-P3
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
clock timezone MSK 4 0
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
no ip icmp rate-limit unreachable
ip auth-proxy max-login-attempts 5
ip admission max-login-attempts 5
!
no ip domain lookup
ip multicast-routing
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
!
redundancy
!
ip tcp synwait-time 5
!
interface Loopback0
 ip address 10.1.0.3 255.255.255.0
 ip pim sparse-mode
```

```
!
interface Ethernet0/0
 description to-R-P1
 ip address 192.168.12.7 255.255.255.254
 ip pim sparse-mode
 ip ospf network point-to-point
 mpls label protocol ldp
 mpls ip
!
interface Ethernet0/1
 description to-R-P2
 ip address 192.168.12.9 255.255.255.254
 ip pim sparse-mode
 ip ospf network point-to-point
 mpls label protocol ldp
 mpls ip
!
interface Ethernet0/2
 description to-R-PE3
 ip address 192.168.12.12 255.255.255.254
 ip pim sparse-mode
 ip ospf network point-to-point
 mpls label protocol ldp
 mpls ip
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 network 10.1.0.3 0.0.0.0 area 0
 network 192.168.12.0 0.0.0.255 area 0
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
ip pim rp-address 10.1.0.1
!
mpls ldp router-id Loopback0
!
control-plane
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
 transport input all
!
end
```

References

1. IP Multicast Routing Configuration Guide, Cisco IOS XE ... (n.d.). Retrieved from http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3850/softw
2. IP Multicast Routing Configuration Guide, Cisco IOS XE ... (n.d.). Retrieved from http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3850/softw
3. IP Multicast Routing Configuration Guide, Cisco IOS XE ... (n.d.). Retrieved from http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3850/softw
4. Multicast VPN | Networking Tech's. (n.d.). Retrieved from http://networkstechnote.wordpress.com/category/multicast-vpn/
5. Multicast VPN | Networking Tech's. (n.d.). Retrieved from http://networkstechnote.wordpress.com/category/multicast-vpn/
6. Emerging Multicast VPN Applications. (n.d.). Retrieved from http://www.webtorials.com/main/resource/papers/juniper/paper3/multicast-apps.pdf
7. Emerging Multicast VPN Applications. (n.d.). Retrieved from http://www.webtorials.com/main/resource/papers/juniper/paper3/multicast-apps.pdf
8. mVPN Architecture &gt; Using Multicast Domains. (n.d.). Retrieved from http://www.ciscopress.com/articles/article.asp?p=32100&seqNum=3
9. www.ciscopress.com/articles/article.asp?p=32100&seqNum=3
10. MDTs &gt; Using Multicast Domains - Cisco Press: Source for ... (n.d.). Retrieved from http://www.ciscopress.com/articles/article.asp?p=32100&seqNum=4
11. MDTs &gt; Using Multicast Domains - Cisco Press: Source for ... (n.d.). Retrieved from http://www.ciscopress.com/articles/article.asp?p=32100&seqNum=4
12. mVPN Architecture &gt; Using Multicast Domains. (n.d.). Retrieved from http://www.ciscopress.com/articles/article.asp?p=32100&seqNum=3
13. MDTs &gt; Using Multicast Domains - Cisco Press: Source for ... (n.d.). Retrieved from http://www.ciscopress.com/articles/article.asp?p=32100&seqNum=4
14. MDTs &gt; Using Multicast Domains - Cisco Press: Source for ... (n.d.). Retrieved from http://www.ciscopress.com/articles/article.asp?p=32100&seqNum=4
15. MDTs &gt; Using Multicast Domains - Cisco Press: Source for ... (n.d.). Retrieved from http://www.ciscopress.com/articles/article.asp?p=32100&seqNum=4
16. MDTs &gt; Using Multicast Domains - Cisco Press: Source for ... (n.d.). Retrieved from http://www.ciscopress.com/articles/article.asp?p=32100&seqNum=4
17. MDTs &gt; Using Multicast Domains - Cisco Press: Source for ... (n.d.). Retrieved from http://www.ciscopress.com/articles/article.asp?p=32100&seqNum=4
18. Understanding JUNOS Next-Generation Multicast VPNs. (n.d.). Retrieved from http://kb.juniper.net/library/CUSTOMERSERVICE/GLOBAL_JTAC/technotes/2000320-en.pdf
19. Understanding JUNOS Next-Generation Multicast VPNs. (n.d.). Retrieved from http://kb.juniper.net/library/CUSTOMERSERVICE/GLOBAL_JTAC/technotes/2000320-en.pdf
20. draft-ietf-mpls-ldp-p2mp-07 - Label Distribution Protocol ... (n.d.). Retrieved from http://tools.ietf.org/html/draft-ietf-mpls-ldp-p2mp-07.txt
21. Understanding JUNOS Next-Generation Multicast VPNs. (n.d.). Retrieved from http://kb.juniper.net/library/CUSTOMERSERVICE/GLOBAL_JTAC/technotes/2000320-en.pdf
22. Emerging Multicast VPN Applications. (n.d.). Retrieved from http://www.webtorials.com/main/resource/papers/juniper/paper3/multicast-apps.pdf
23. Emerging Multicast VPN Applications. (n.d.). Retrieved from http://www.webtorials.com/main/resource/papers/juniper/paper3/multicast-apps.pdf
24. Understanding JUNOS Next-Generation Multicast VPNs. (n.d.). Retrieved from http://kb.juniper.net/library/CUSTOMERSERVICE/GLOBAL_JTAC/technotes/2000320-en.pdf

25. Understanding JUNOS Next-Generation Multicast VPNs. (n.d.). Retrieved from http://kb.juniper.net/library/CUSTOMERSERVICE/GLOBAL_JTAC/technotes/2000320-en.pdf
26. Emerging Multicast VPN Applications. (n.d.). Retrieved from http://www.webtorials.com/main/resource/papers/juniper/paper3/multicast-apps.pdf
27. Deploying NG Multicast Enabled applications: authors: Vinod Joseph, Srinivas Mulugu ISBN: 978-0-12-384923-6
28. NaNOG 49 (North American Operators Group), MPLS for dummies (Presentation) Richard A Steenbergen <ras@nlayer.net> nLayer Communications, Inc. Retrieved From http://www.nanog.org Nanog 49 archive.
29. NG MVPN BGP ROUTE TYPES AND ENCODINGS. Juniper Networks application note.
30. Multicast Virtual Private Network Concepts. http://www.cisco.com/en/US/tech/tk828/technologies_white_paper09186a00800a3db6.shtml#wp34608
31. MPLS and VPN Architectures, Volume 2 by Jim Guichard, Ivan Pepelnjak, Jeff Apcar
32. MPLS-Enabled Applications; Emerging Developments and New Technologies by Ina Minei, Julian Lucek
33. Investigation into Layer 3 Multicast Virtual Private Network Schemes by Munir Ibrahim Bazaja
34. Deploying Next Generation Multicast-enabled Applications: Label Switched Multicast for MPLS VPNs, and Wholesale Ethernet
35. Network Convergence Ethernet Applications and Next Generation Packet Transport Architectures by Vinod Joseph and Srinivas Mulugu