Design of MPLS networks VPN and TE with testing its resiliency and reliability

Diploma thesis

Michal Aron

Brno, spring 2014
ZADANIE DP
Declaration

I declare that I have worked on this thesis independently using only the sources listed in the bibliography. All resources, sources, and literature, which I used in preparing or I drew on them, I quote in the thesis properly with stating the full reference to the source.

Michal Aron

Acknowledgement

I would like to say my thank you to doc. Ing. Jaroslav Dočkal, CSc. for his technical assistance, valuable advices and forbearance, he provided to me while I was writing this thesis.
Abstract

Thesis will describe MPLS networks and explain a need for such a technology as well as its contribution to networking in general and specifically the way of supporting multiple customers. Next area of focus will be the effort to provide all the services of such a network while simultaneously keeping customers isolated, network manageable and mainly reliable. Furthermore thesis will include details about MPLS network implementation, traffic engineering along with failover functionality. Lab part consists of real implementation of the MPLS networks under 2 different major vendors, which are Cisco and Juniper. This service provider’s network will be tested to different kind of failure scenarios in order to provide results of multiple methods used for network resiliency and reliability. Emulation techniques will be used to achieve the most real-world implementation while keeping the whole network environment inside one computer. GNS3 software stack with VMware virtualization were chosen for this purpose, as these applications are well suited for emulation of real network environment.
Keywords

MPLS, MPLS-TE, VPN, L3VPN, L2VPN, VPLS, BGP, MP-BGP, VRF, FRR, Path protection, Local protection, Failover functionality.
Contents

1 Introduction ..................................................................................................................... 1

1.1 Historical background ................................................................................................. 1

1.2 Driven factors for deploying MPLS network .............................................................. 2

1.3 Current trends ............................................................................................................ 3

2 Foundation of MPLS networks .................................................................................. 5

2.1 MPLS architecture ..................................................................................................... 6

2.1.1 MPLS header ......................................................................................................... 6

2.1.2 Labels .................................................................................................................... 7

2.1.3 LERs and LSRs .................................................................................................... 8

2.1.4 FEC ...................................................................................................................... 9

2.1.5 LSP ..................................................................................................................... 10

2.1.6 MPLS operating modes ....................................................................................... 13

2.2 Control plane ............................................................................................................ 17

2.3 Forwarding plane ...................................................................................................... 17

2.4 LDP ........................................................................................................................ 18

2.5 RSVP-TE .................................................................................................................. 19

3 Variants of MPLS implementations ......................................................................... 22

3.1 Introduction ............................................................................................................... 22

3.1.1 Virtual Private Network (VPN) ........................................................................... 22

3.1.2 Overlay VPN model ............................................................................................ 23

3.1.3 Peer-to-peer VPN model .................................................................................... 24

3.2 Layer 3 MPLS VPNs ................................................................................................. 26

3.2.1 The BGP/MPLS VPN model .............................................................................. 26

3.2.2 Virtual Routing and Forwarding (VRF) instances ............................................ 27

3.2.3 Distribution of routing information ................................................................. 29
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3</td>
<td>Layer 2 MPLS VPNs</td>
<td>40</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Layer 2 interworking</td>
<td>41</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Layer 2 over MPLS transport principles</td>
<td>41</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Forwarding plane</td>
<td>42</td>
</tr>
<tr>
<td>3.3.4</td>
<td>Control plane</td>
<td>44</td>
</tr>
<tr>
<td>3.4</td>
<td>Virtual Private LAN Services (VPLS)</td>
<td>46</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Forwarding plane</td>
<td>47</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Control plane</td>
<td>47</td>
</tr>
<tr>
<td>3.5</td>
<td>Conclusion</td>
<td>49</td>
</tr>
<tr>
<td>4</td>
<td>MPLS traffic engineering</td>
<td>50</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>50</td>
</tr>
<tr>
<td>4.2</td>
<td>Goals of TE functions</td>
<td>50</td>
</tr>
<tr>
<td>4.3</td>
<td>Setting up TE paths</td>
<td>53</td>
</tr>
<tr>
<td>4.3.1</td>
<td>LSP priorities and preemption</td>
<td>53</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Distribution of TE information</td>
<td>53</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Link coloring</td>
<td>54</td>
</tr>
<tr>
<td>4.3.4</td>
<td>CSPF</td>
<td>55</td>
</tr>
<tr>
<td>4.3.5</td>
<td>Selection of TE paths</td>
<td>56</td>
</tr>
<tr>
<td>4.4</td>
<td>MPLS DiffServ-TE</td>
<td>56</td>
</tr>
<tr>
<td>5</td>
<td>Failover functionality in MPLS</td>
<td>58</td>
</tr>
<tr>
<td>5.1</td>
<td>Introduction</td>
<td>58</td>
</tr>
<tr>
<td>5.2</td>
<td>Path protection</td>
<td>59</td>
</tr>
<tr>
<td>5.3</td>
<td>Local protection</td>
<td>61</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Pre-failure configuration</td>
<td>63</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Failure detection</td>
<td>64</td>
</tr>
<tr>
<td>5.3.3</td>
<td>Connectivity restoration</td>
<td>67</td>
</tr>
<tr>
<td>5.3.4</td>
<td>Post-failure signaling</td>
<td>67</td>
</tr>
</tbody>
</table>
1 Introduction

Multiprotocol Label Switching (MPLS) networks are currently the most used transport technology for service provider networks. This has happened mainly because of the various different features available in a single solution which wasn’t possible to achieve by any other transport technology used that time. In the past technologies such as Asynchronous Transfer Mode (ATM), Frame Relay along with circuit switching were used. These solutions don’t offer diversity or variety in terms of building a well scalable network and are closely related to a single transport technology. Another main factor is that traffic engineering wasn’t developed enough to allow a service provider to optimize the traffic flow and links utilization as it would be demanded of a service provider’s network. MPLS sufficiently satisfies prerequisites of such a network mainly because of protocol structure and the place in OSI model where it operates.

1.1 Historical background

The first working group of the IETF that were supposed to design MPLS and address the problems that service providers were facing that time took place in 1997. This working group still exists and since then MPLS has grown into a protocol which is widely used and dependent in many network environments. The designers initially tried to address the most problematic issues at that time with the potential for further development. They had to come up with an idea which would allow for faster routing decisions with backwards compatibility. This was not only for former protocols, but also for equipment. The protocol which sits in between the data-link layer and network layer was defined. It is sometimes referred to by the community as a layer 2.5 protocol. This allows MPLS to operate above any existing layer 2 protocol with the encapsulation of network layer. It is a very strong feature which allows the interconnecting of different network technologies and grouping them under one solution. The protocol is using its own addressing scheme based on labels. Since we are no longer dependent on the network layer, we can have a much greater performance in forwarding. Switching between labels is faster and easier than doing lookups in the routing table.
Introduction

Generally speaking, looking for an exact match costs much less than looking for the longest prefix. Today this isn’t seen as a big obstacle as today’s devices perform routing decisions in hardware rather than by a central processor. Today’s modern ASICs have mostly eliminated performance issues because it is common for one ASIC to do tens of millions routing lookups per second relatively cheap, but yet taking the significant amount of hardware resources. However back in those days computing power was an issue.

1.2 Driven factors for deploying MPLS network

It has been over a decade since MPLS started to take over other routing protocols in order to provide data transport services as well as routing services. There have to be reasons why this protocol is so successful rather than being deprecated over the years.

Multi-service network

The reason why MPLS networks have become so successful is due to the ability of implementing a multi-service network. Having a network infrastructure which consists of a variety of different technologies, while still being maintained by a single standard that provides everything ever needed to control the traffic has no competition.

Virtual private networks

The second most significant driving factor is the ability to provide private and secure connections, known as virtual private networks (VPNs), for many different customers over the very same network topology. This is a great superiority over other existing solutions.

Traffic engineering functions

Another significant reason for deploying a MPLS network is the implementation of traffic engineering (TE) functions. The ability to have control over the traffic, where and how it flows in the network offers to have overall network capacity wisely utilized, avoiding congestions and prioritizing more sensitive traffic.
Network resiliency and reliability

Network resiliency and reliability capabilities have also become very popular features provided in MPLS. It is required for a service provider’s network to have network resiliency and to be protected against different kinds of failures and outages. The quality of such a network is to continue providing guaranteed service to the customers without any violation of service level agreements (SLAs). Otherwise sanctions may arise. This thesis dedicates the whole of chapter 5 to address resiliency and reliability features available in MPLS networks.

Scalability

Before MPLS, many networks had a core of ATM switches surrounded by routers that were typically fully meshed and had a square number of adjacencies. Enabling a structured MPLS network solves this kind of problem. The devices in the core are not involved in any kind of relationship with the rest of the network and their only purpose is to switch the packets. The virtual connections are built and maintained on the devices surrounding this core part of the network which significantly reduces the amount of virtual paths. On this level, resiliency and reliability of the core trunks are secured. These devices are usually attached to edge devices which act as an entrance point for the outside world in to the network. These are the ones that are providing and securing customers' traffic. They peer directly with individual customers and inject their traffic to the service provider’s network.

By following a structured approach it makes it much simpler for network operators to manage, maintain and troubleshoot the network.

1.3 Current trends

Today’s MPLS networks can be deployed in number of various different implementations. The easiest implementation to achieve is simply implement MPLS service into existing network topology. This would bring into the network benefits of MPLS technology. Nowadays already small corporations have spread their departments across the country or even worldwide. In order to do their business they have to interconnect with each other in a way they
appear to be a part of the same realm. They could place a private line across land in order to achieve desired effect but it would be extremely expensive and almost unreal to implement. Rather what they choose to do is ask available service providers for a private connection between their branches with the same desired effect but much cheaper. However the most widely deployed solution is Layer 3 MPLS VPN. Sometimes it is the only service MPLS network was built for. Layer 3 VPNs offer secured private line without any limitations or restrictions for the customers. Benefit for the customer is that the routing is taken care on service provider’s side so he doesn’t need to dedicate additional resources to maintain connection across branches. Other option for a customer would be to lease a layer 2 connection. Here the connection appears to be simple point-to-point line of chosen technology without any control of above layers meaning the customer is the one who is doing the routing. It can be used to transport customer’s layer 2 traffic such as Ethernet, ATM or Frame Relay. Finally the last option is to ask for a complete Local Area Network services where each branches are connected to service provider’s network with the impression as they would have been attached to a single switch.

For the past decade it has been proven MPLS is a tool of the future. As the idea of moving natively non IP services into IP network also MPLS tries to be reliable transport technology regardless application it will carry. Many service providers are trying to implement connections for applications which are using completely distinct networks such as Public Switched Telephone Network (PSTN), TDM, Broadcasting of TV or connection for legacy transport technologies.
2 Foundation of MPLS networks

Regular routed packet travels across network based on routing decisions made by each individual routing device on a path from source to destination. Each router needs to process an information in the packet header, analyze it and perform routing decisions based on this analyses and destination address. This means every single device on the path has to repeat this process and locally decide where to forward the packet. For example an IP packet contains significantly more information than a simple action of forwarding packet out of some interface requires. However not only that. Header of IP packet contains CRC field which due to change of TTL value needs to be recalculated in order to provide error detection.

Forwarding a packet to its next hop is set of two different functions. First one is to analyze an entire set of all possible packets which can travel across and associate them with forwarding equivalence classes (FECs). All packets belonging to the same FEC\(^1\) are handled same way and will always follow same path to reach their destination. Second function is to associate each FEC with particular next hop\(^2\). Both these actions are required to be run on each device on the path.

By the beginning of 2001 IETF founded MPLS protocol which solves problems of independent network using time consuming operations to forward packets. Forwarding decisions are done based on labels which by the end forms one Label Switched Path (LSP) across the MPLS network. Note that forwarding packet based on incoming label value is reverse logic than forwarding the traffic in the routed environment. Since the devices no longer care about the headers at other layers and forward traffic just based on a label it makes any device which is capable of doing label lookups and their replacement to operate inside MPLS network.

Packet classification and forwarding decision are done only once at the ingress device where the rest of MPLS networks doesn’t inspect inner headers but only perform an exact match lookup for incoming to outgoing label

---

\(^1\) For example if two different IP packets have a destination addresses which both fall under one longest prefix match then they are considered to belong to the same FEC.

\(^2\) In case of load balancing multiple next hops are associated with FEC.
association, swap those two *labels* and forward packet out of particular interface.

Consider a packet which requires additional actions to determine specific FEC. Using the *conventional forwarding* every device would have to go through those actions where in MPLS only first device on the path. Handing over those actions from the core to the edge is reducing the number of actions needed to process a packet. This allows core device to concentrate all their power resources just to perform simple *label* swapping and thus distribution of packets closer to their destination. Given that it is obvious the whole path inside MPLS domain is determined at the edge device which choses particular LSP for determined FEC of a packet.

To have an overall control over the network regarding traffic flow and providing certain guarantees in some cases is desirable to force packets to follow certain route which is explicitly defined. Following this approach provides a desired control over the whole traffic where it flows and how it is handled on its way. Additional constraints can be introduced for LSPs such as bandwidth assurance, priority, link attributes and path options [1]. As for service provider’s network it is essentials to provide certain guarantees to the customers for their applications.

### 2.1 MPLS architecture

This section is going to provide certain level of MPLS basics in order to understand further topics of this thesis.

#### 2.1.1 MPLS header

MPLS is using its own 32 bit long header which logically fits between layer 2 and 3 of OSI model. Header consists only of 4 fields where 20 bits are used for a *label*, next 3 bits were saved to be experimental, but now they are used for quality of service (QoS), another bit marks the bottom of the MPLS *header stack* and last 8 bits are used for time to live (TTL).
Foundation of MPLS networks

As it can be seen in Figure 2-1 MPLS header sits in between layer 2 and 3 giving MPLS characteristic of a layer 2.5 protocol.

2.1.2 Labels

Label is a local identifier of fixed length used for FEC identification. It is an arbitrary number only locally significant. Label association to a particular FEC is driven action of a downstream device generating label value and then advertising it to an upstream neighbor with the expectation to receive this label value for a traffic associated with particular FEC. In other words it is the agreement between adjacent upstream and downstream device anywhere in the network to use this label to FEC binding for moving the traffic from upstream to downstream node. For a label value there is a 20 bit space in MPLS header giving in total 1048576 possible choices. Labels 0 through 15 are reserved for special use. Table 2-1 shows the purpose of all reserved labels. Rest of them from 16 through 1048575 are free to be used.

<table>
<thead>
<tr>
<th>Label</th>
<th>Usage</th>
<th>RFC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Explicit NULL for IPv4</td>
<td>RFC 3032</td>
</tr>
<tr>
<td>1</td>
<td>Router Alert Label</td>
<td>RFC 3032</td>
</tr>
<tr>
<td>2</td>
<td>Explicit NULL for IPv6</td>
<td>RFC 3032</td>
</tr>
<tr>
<td>3</td>
<td>Implicit NULL</td>
<td>RFC 3032</td>
</tr>
<tr>
<td>4 - 6</td>
<td>Unassigned</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Entropy Label Indicator</td>
<td>RFC 6790</td>
</tr>
<tr>
<td>8 - 12</td>
<td>Unassigned</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Generic Associated Channel Label</td>
<td>RFC 5586</td>
</tr>
<tr>
<td>14</td>
<td>OAM Alert Label</td>
<td>RFC 3429</td>
</tr>
<tr>
<td>15</td>
<td>Unassigned</td>
<td></td>
</tr>
</tbody>
</table>

Table 2-1  Reserved labels and RFC where it is documented

![Figure 2-1](https://via.placeholder.com/150)  MPLS header
2.1.3 LERs and LSRs

Devices operating on the edge of a MPLS network which are the touch point for customers are called Label Edge Routers (LERs) also known as Provider Edge (PE) device. Further devices in the core which perform only label swapping from MPLS perspective are called Label Switch Routers (LSRs) or simply Provider (P) device. Finally the devices which are on the other side of the link facing customer are called Customer Edge (CE) devices.

Figure 2-3 demonstrates these positions in the network topology.

Figure 2-2 Example of MPLS network with customers

To able to orientate on a LSP terms as upstream direction and downstream direction were introduced. Those terms also apply to an adjacent LSRs for particular LSP in order to differentiate their relative position against each other. Figure 2-3 demonstrates these positions in the network topology.

For further discussion to have a sense an LSP needs to be defined. Suppose we need to build a LSP across the network in a direction from LER1 towards LER2. Then a flow towards destination has a downstream direction. Opposite flow heading to the source has an upstream direction. If the concern is a neighbor ship between two adjacent LSRs on a particular LSP, to the device closer to the source is being referred as to the upstream node (LSR2 in Figure 2-3) and his neighbor closer to the destination as to the downstream node (LSR3 in Figure 2-3).
2.1.4 FEC

A FEC is a flow of packets that receive the same forwarding treatment along the path. All packets belonging to the same FEC have the same label. However, not all packets that have the same label belong to the same FEC, because their EXP bit values might differ. Because of that the forwarding treatment isn’t necessary the same, and they could belong to a different FEC [2].

Packet classification to the particular FEC is done by the ingress LSR\(^3\). This demonstrates the whole logic of MPLS behavior in contradiction with conventional forwarding. Same FEC can be determined to the packets which:

- Are heading to the same destination.
- Are in the same multicast group.
- Have same treatment according to their precedence or IP DiffServ.
- Belong to the same VC or sub interface at the ingress point of LSP.

To point out the main outcome of FEC classification is that each packet belonging to the same FEC is treated same way receiving\(^4\) identical label at each point before forwarding.

---

\(^3\) Terms ingress LSR and ingress LSP are used interchangeably where ingress LSR is actual device and ingress LSP is a logical point where the LSP have its beginning. Similarly terms are interchangeable for the egress LSR and egress LSP.

\(^4\) In case the packet is reaching egress LSR label is popped.
2.1.5 LSP

LSP is a unidirectional path formed inside MPLS network between two nodes. Nodes don’t necessarily have to be on the edge of the network. In other words a LSP is a sequence of LSRs which forward labeled packet of a certain FEC. Node which initiates a LSP is called a head end of LSP or has an acronym ingress LSP. Likewise to the node where the LSP is terminated is being referred to as tail end of LSP or with acronym egress LSP. The knowledge which label should be used for the downstream node is the matter of control plane where label bindings are being exchanged between nodes with the aid of one label distribution protocol. Example of LSP can be seen in Figure 2-3 as a sequence of nodes LER1-LSR2-LSR3-LER2.

In convention forwarding each device does a packet classification where unlikely in MPLS we have 3 different functions which are performed along the LSP in order to forward a packet.

Figure 2-4 demonstrates a label distribution and packet forwarding of following functions.

---

5 Node where a transition between layer 3 routing to MPLS switching happens (for example IP to MPLS).
6 Node where the transition is done back from MPLS switching to the layer 3 routing (MPLS to IP).
7 When referring generally to all label distribution protocols term LDP (actual label distribution protocol) tries to be avoided.
2.1.5.1 Label imposition

Creating a new header with a particular label value representing certain FEC is referred by the term label imposition or label pushing. Packet of an existing technology which is being transferred through MPLS network is imposed a new label at the ingress LSR. Figure 2-1 shows the label is being imposed inside MPLS header between layer 2 and 3 headers.

2.1.5.2 Label swapping

Label swapping is the process where LSR simply and quickly swap label values of incoming packet for its downstream neighbor. This process is being compared to a hot potato term. Its purpose is to get rid of packet as soon as possible. Following actions are required in order to do so:

1) LSR does a label lookup for the incoming label value of a packet.
2) Outgoing label and interface are determined for a particular label.
3) LSR swaps incoming label with outgoing and forwards the packet out of determined interface.

2.1.5.3 Label disposition

When the labeled packet is being disposed of its label and thus MPLS header terms label disposition or label popping are used to represent this action. There are two methods for disposing a label which differ in a behavior of the next to last (or penultimate) LSR:

- Explicit null – egress LSR advertise to its upstream adjacent neighbor label of a value 0 which represents IPv4 explicit null (for IPv6 explicit null label value of 2 is used). This tells penultimate LSR it is next to last node on the LSP and it will swap an incoming label value to the label of value 0. Having packet labeled at the egress LSR prevents device from keeping any unnecessary information regarding POP labels and necessity to inspect encapsulated payload in case of MPLS QoS in order to determine packet handling from the QoS perspective. An explicit null preserve the LSP Class of Service (CoS) behavior across the entire LSP by keeping MPLS header until it reaches egress LSR.
• Implicit null – egress LSR advertise to its upstream adjacent neighbor a label value of 3 which represents *implicit null*. This tells penultimate LSR instead of performing a swap operation rather to pop a label with the MPLS header with the result of egress LSR receiving an original packet which was delivered to the ingress LSR. It saves an addition action to be taken at egress LSR to get the encapsulated payload. This approach is generally known as *penultimate hop popping*. Since in this case MPLS header isn’t present at the egress LSR, node is forced to handle the packet according to the CoS settings specified in the underlying payload.

### 2.1.5.4 Penultimate hop popping

Using a penultimate hop popping avoids a scenario where the egress LSR would need to do two label lookups. From the logical point of view there is no need to perform label lookup at the egress LSR. This is because the labeled packet of a certain depth of label stack reached its destination. In other words MPLS packet will get rid of one MPLS header at the penultimate node with the result where packet arrives to egress LSR with following label on a label stack or in case there isn’t another label to perform a lookup on, packet is treated based on the original protocol. Now if the egress LSR receives a labeled packet it simply does a *label lookup* and forwards a packet with the new *label* as there wasn’t any overlying tunnel.

### 2.1.5.5 LSP next hop

When referring to a next hop of a LSP it is always assumed to a next hop determined from the forwarding table based on an incoming *label* value. This particular next hop may differ from a next hop which would be determined from the underlying protocol.

### 2.1.5.6 Label stack

Label stack is a succession of labels each embedded in its own MPLS header as shown in Figure 2-5. Last MPLS header on the stack has bit indicating *bottom of the stack* set to 1.
2.1.6 MPLS operating modes

2.1.6.1 Label assignment and distribution

In the MPLS architecture, the decision to bind a particular label \( L \) to a particular FEC \( F \) is made by the LSR which is downstream with respect to that binding. The downstream LSR then informs the upstream LSR of the binding. Thus labels are "downstream-assigned", and label bindings are distributed in the "downstream to upstream" direction [3].

Label assignment starts at the ingress LSR by applying MPLS header with particular label value to the underlying packet what can even be an MPLS labeled packet or a packet of some other protocol which is being transferred across MPLS network. Next downstream node will swap label of an incoming packet with an outgoing label which only matters to the following downstream node. Once labeled packet reaches egress LSR, label is being stripped of with the MPLS header and thus LSP is terminated. All this wouldn’t happen if there weren’t appropriate label values distributed across the MPLS network.

Labels provide only local meaning to the device itself. For a successful communication it is needed some sort of intelligence which will tell devices which particular label value to use for sending particular packet to downstream neighbor otherwise nodes wouldn’t know how to handle incoming packet and which label value is expected by the next hop. This intelligence can be implemented in two different ways:

- It can be installed on an underlying IP routing protocol – so called piggybacking labels.
- It can be provided using a separate protocol – by label distribution protocols.

Figure 2-5 MPLS label stack format
Piggybacking labels on an underlying IP routing protocol

This method relay on an existing IGP protocol for a label distribution rather than introducing separate protocol to be run on LSRs. For this purpose IGP protocol needs to be extended to carry labels. Advantage is that there always exists a label for a particular FEC since the function is embedded.

Since BGP isn’t IGP but it can provide label distribution which is mostly used for MPLS VPN networks.

Using a separate protocol for distributing labels

In many different environments it is more convenient to introduce new intelligence for distributing labels. There are several reasons why to use independent protocol rather than existing IGP:

- Not all platforms provide extension for label distribution.
- MPLS network can be spread out on more Autonomous Systems (AS) where IGP is strictly related to a single AS.
- It would be challenging for some scenarios implement label distribution into link-state protocol.
- Separate protocol provides independent environment where label distribution can be enriched by implementing many other useful features.
- Relaying on an IGP haven’t become very popular idea.

Nowadays most widespread label distribution protocols are:

- Label Distribution Protocol (LDP).
- Resource Reservation Protocol - Traffic Engineering (RSVP-TE) [4].

2.1.6.2 Label distribution modes

In MPLS exists two different approaches for label binding distribution. Figure 2-6 demonstrates both label distribution modes.


**Unsolicited downstream**

In this mode LSR advertise label bindings to the adjacent downstream LSRs without making those LSR sending out request for a particular label bindings.

**Downstream on demand**

In this mode LSRs have to request adjacent upstream LSR for a particular label binding. Then per request specific LSR responds with a label binding for given FEC.

### 2.1.6.3 Label retention mode

Different modes of label retention refer to a case how device handle label bindings which is not currently using.

**Liberal label retention**

Using liberal label retention mode (LLR) device holds all received label bindings in FIB. Only label binding advertised by downstream LSR and used for forwarding for a particular FEC is put into LFIB. Rest of label bindings received from other LSRs associated with given FEC are hold in FIB. This enables LSR to immediately update LFIB of new label binding and start
forwarding once network topology changed or failure related to primary label binding was detected.

**Conservative label retention**

If device is using conservative label retention (CLR) mode it holds only label binding advertised by downstream LSR for a particular FEC.

**Conclusion**

The main difference between LLR and CLR is that LLR adapts quickly to topology changes while CLR stores fewer label bindings in the memory. It is good practice to use CLR mode with downstream on demand label distribution whereas LLR mode goes better with unsolicited downstream label distribution. It doesn’t make much sense for device using CLR mode to receive unsolicited labels since only one will be used as a next hop and rest will get rejected. Similarly for a device using LLR mode is more convenient to get unsolicited label bindings rather than flood the network with requests.

### 2.1.6.4 LSP control mode

The device can create a local label binding for a particular FEC by following two different approaches.

**Independent LSP control mode**

When Independent LSP control mode is in use device will create local label binding independently from any other LSRs. This happens as soon as new FEC has been recognized.

**Ordered LSP control mode**

While ordered LSP control mode is in use, LSP formation is initiated at the ingress LSR and successively progressing through the network till it reaches the egress point.

In Ordered LSP Control mode, an LSR only creates a local binding for a FEC if it recognizes that it is the egress LSR for the FEC or if the LSR has received a label binding from the next hop for this FEC [2].
Conclusion

While using independent LSP control mode it may happen that device will start forwarding the traffic before whole LSP has been established which can result in inconsistent packet forwarding or even dropping the packet.

2.2 Control plane

Control plane of the router is the place where all computations related to the final packet forwarding decision happens. The biggest entity is the routing protocol (or static routes) which are responsible for filling up the routing table. The context of routing protocol allows to configure various set of features determining the final form of routing process. Furthermore router can be configured with additional components which help to achieve desired control over packet flooding through the router. It can be firewall, access-lists, policy based routing, QoS and others.

Components of control plane are:

- Routing protocol – is the intelligence responsible for filling up the routing table.
- Routing Information Base (RIB) (e.g. IP routing table) – is the place, where router checks for destination longest prefix match in order to forward the packet further to the destination. If multiple candidates for the same destination prefix exists the prefix source with lower Administrative Distance (AD) will be installed into RIB.
- Label Distribution Protocol – is the protocol responsible for creating LSP through the MPLS enabled network by exchanging label information.

2.3 Forwarding plane

Forwarding plane or sometimes called data plane is the part of the router actually responsible for packet forwarding. It has a separate table defining final forwarding decision for incoming packets. Usually the forwarding information is logically kept inside a table and is driven from the control plane. Depending on vendor and platform the form of forwarding table varies.
Foundation of MPLS networks

After all the idea is the same. To speed up the process of routing decisions over incoming packets.

Forwarding plane of MPLS enabled router has two forwarding tables:

- **Forwarding Information Base (FIB)** (IP forwarding table) – provides mapping of incoming destination prefix to an outgoing interface.
- **Label Forwarding Information Base (LFIB)** – provides mapping of a local label to an outgoing label along with the outgoing interface.

![Diagram](image)

**Figure 2-7** Management of control plane and data plane

### 2.4 LDP

LDP is the most basic label distribution protocol used for label distribution across MPLS network. While finding LSP from ingress LSR to the egress it always follows underlying IGP path. It is useful if we need to implement MPLS really quickly into network topology. Just by enabling LDP on the interfaces, full-mesh of tunnels is automatically created across the entire topology based on the reachability information of IGP protocol. This makes it really easy to use but unable to handle TE requirements for the MPLS network.
2.5 RSVP-TE

Resource Reservation Protocol is the IntServ protocol for allocating network resources along the path for certain streams or data flows. Its initial purpose was to allow end users to request the network for specific QoS settings and then help nodes on the path to establish and maintain such a connection. Protocol was developed long before MPLS was created. MPLS working group adopted RSVP for the traffic engineering purposes and defined it as RSVP-TE standard. It provides functionality for path signaling with all necessary extensions to add a traffic engineering capabilities to the MPLS network.

RSVP-TE provides following features:

- Path signaling and maintaining
- Constrained Shortest Path First (CSPF) calculation
- Explicit path definition
- Path resource reservation
- Fast rerouting capabilities
- Link coloring
- LSP preemption

Ingress router initiate a unidirectional RSVP session by sending the Path message downstream towards egress point in the network. Path message contains attributes to signal required resources along the path. Afterwards the Resv message is sent upstream to the ingress LSR containing information whether required path attributes were granted or not and also securing the label assignment for a LSP.

RSVP-TE messages used to signal sessions:

- Path – used to create a LSP or do a periodic refresh. It is sent downstream. It has to be processed by all RSVP devices on the path.
- Resv – used to signal reserved resources and takes care about label assignment. It originates from the egress LSR and is processed by all RSVP devices till it reaches ingress LSR.
- PathTear – always travels downstream and is used to delete LSP and thus free its booked resources. Can be sent by ingress LSR or any node which connection timed out.
Foundation of MPLS networks

- ResvTear – used only to delete allocated resources for given LSP.
- PathErr – used to report errors while processing path message and travels upstream direction.
- ResvErr – message used to report errors in reservation resources and travels hop by hop to the egress LSR.

RSVP-TE protocol maintains 3 states for each LSP:

- Soft state – it is used to maintain LSP alive. Path is periodically refreshed by Path and Resv messages. It is called soft because if path isn’t refreshed after certain amount of time it is deleted. RSVP of each router periodically scans soft state of each LSP.
- Path state – node maintain path state for each RSVP session which is passing through. Session because LSP does not necessarily have to be established. Information in path state are driven from Path message.
- Reservation state – information is taken from the Resv message and it holds the reservation request of each RSVP session.

Recall initially RSVP was meant to be host-to-host reservation protocol. Several extensions have been added to RSVP protocol to fully support MPLS-TE functions. This modification created a new router-to-router protocol called RSVP-TE.
RSVP-TE protocol *Path* message objects:

- Mandatory objects:
  - SESSION – used to indicate tunnel type.
  - LABEL_REQUEST – used for passing label information.
- Optional objects
  - EXPLICIT_ROUTE – used for specifying nodes which RSVP message has to traverse through. It can contain strict list which tells exactly through which hops message will go or specifying *loose hops* which needs to be visited before reaching egress LSR.
  - RECORD_ROUTE – contains addresses of each node *Path* message passed through.
  - SESSION_ATTRIBUTE – used to assign specific attributes to the RSVP session.
  - RSVP_HOP – contains the hop IP address of a preceding node.

RSVP-TE protocol *Resv* message objects:

- Mandatory objects:
  - SESSION – used to indicate LSP establishment.
  - LABEL_REQUEST – used for label distribution process.
  - STYLE – specify the style of reservation process.
- Optional objects
  - RECORD_ROUTE – used to return the list of hops back to the ingress LSR.
  - RSVP_HOP – contains the hop IP address of a preceding node.

RSVP-TE is the protocol well suited for handling TE requirements for MPLS network. It allows network operators to configure individual LSP to get proper handling by the nodes on the path. It will use the signaling either to successfully establish LSP or it will provide source information about LSP establishment failure. Chapter 4 is focused on TE functions available in MPLS only via RSVP-TE. Furthermore in chapter 5 thesis describes various use of RSVP-TE protocol to signal *path protection* and restoration.
Variants of MPLS implementations

3 Variants of MPLS implementations

3.1 Introduction

MPLS enabled network allows to implement various different solutions for transporting customers traffic across same network infrastructure. As mentioned in section 1.3 Current trends, MPLS is found nowadays as the most beneficial technology for a service provider network. MPLS provides transparent tunneling between network endpoints. Only endpoints are involved in end user traffic classification which is then being forwarded across the MPLS core. This saves the core devices of running an intelligence needed for managing end users’ traffic. This chapter describes different variants of MPLS implementations used in today’s modern service providers network which are:

- Layer 3 MPLS VPNs
- Layer 2 MPLS VPNs
- Virtual Private LAN Services

3.1.1 Virtual Private Network (VPN)

Essentially VPN is a very generic term. The definition says it’s a network connection between devices which don’t share a single physical connection between each other. In other words it extends a private network across different network to cross physical boundaries. This carrier network can be Internet or a network of a service provider.

Example of VPN connections:

- From the layer 2 perspective:
  - Ethernet VLANs
  - PVCs either Frame Relay or ATM
  - MPLS VPNs or VPLS
- From the layer 3 perspective:
  - GRE Tunnels
  - IPsec Tunnels
  - MPLS VPNs
3.1.2 Overlay VPN model

Then main stroke of Overlay VPN model is that service provider does not participate in routing with its customers. This means the communication needs to be provisioned prior to be actually working. It simply provides a point-to-point connections for a customer.

One of the biggest drawback of overlay VPN model is that it suffers from the scalability issues. In order to provide connections across whole infrastructure using the full-mesh fashion the total number of tunnels between devices is \((n \times (n - 1))/2\) where \(n\) represents the number of sites. Now it is quite simple to imagine how many circuits would have to be provisioned. Just for the network of 10 sites it requires 45 provisioned connections in order to get a full-mesh. The amount of required configuration can be overwhelming. When a new site is added to the topology it requires to be configured with all other sites and vice versa additional configuration has to be done on the rest of the sites to peer with the new one. In the big topologies it isn’t convenient to follow such an approach. Further difficulties are involved with bandwidth reservation since there is lot of provisioned connections across the network which makes very hard to keep an overall track of provisioned bandwidth reservations. When it comes to the act of increasing bandwidth often happens only way is to set up and provision new connections which is not a very convenient solution.

Since service provider is not involved in routing it allows individual customers to use overlapping addressing scheme.

Here belongs:

- PVCs either Frame Relay or ATM.
  - Service provider offers a virtual private connection which is using layer 2 switching in the middle.
- Leased lines.
  - Such as T-carrier or E-carrier used to provide connections. Individual connections can be aggregated into channels in order to cross the core.
Variants of MPLS implementations

- Tunnels – GRE or IPsec.
  - Creating an overlay layer 3 tunnel where service providers does not participate in a routing of a virtual private network. Typical example can be IPsec tunnel across the provided internet connectivity.

3.1.3 Peer-to-peer VPN model

Opposite model to the overlay VPN model is peer-to-peer VPN model where service provider does participate in the customer’s routing. Rather than CE devices peering with each other and thus creating an overlay infrastructure on the top of service provider’s network they only peer with the directly attached PE device. As a result of this concept large mesh of routing peerings between CE routers disappears [5]. This means that static provisioning is no longer required.

Other benefits are available:

- Required configuration is now done only for a particular PE and CE.
- Multiple CE can now attach to a single PE device.
- Bandwidth upgrade is now separated from the core of service provider’s network. When customer needs more bandwidth simply additional links are placed between particular CE and PE. When the core of the network suffers from bandwidth availability new links are only placed there.
- All customers are served by a single logical infrastructure working altogether rather than maintaining separate connections between individual CE devices.
- Maintaining reachability from the customer’s point of view. He only advertise its prefixes and the actual routing between sites is taken care by the provider.

However the big disadvantage is that for service provider it is almost impossible to maintain such a network. This is because customers’ traffic is routed or switched inside service provider’s network and it has to be kept separate from the control traffic of a service provider and the traffic belonging to the other customers. Providing separation and privacy is the key benefit of
Variants of MPLS implementations

VPNs and in peer-to-peer VPN model is achieved by implementing complex filtering policies such as:

- For a layer 2 networks:
  - Private VLANs.
  - Access-list for layer 2.
- For layer 3 networks:
  - Access-lists for layer 3.
  - Policy-based routing.

It is not only complex filtering policies but also lot of administration overhead which comes along with maintaining such a solution.

Important thing to realize is that a separate addressing scheme has to be implemented between service provider and each customer. It isn’t possible for any of them to have overlapping address space. If a service provider has a multiple customers on the same network, even when they are not directly communicating with each other, service provider would have to have implemented unique addressing. This rules out the use of private address space for the IP networks. Every single device in service provider network is involved in making the local forwarding decisions which requires for all of them to follow same addressing scheme and to be familiar with which prefix belongs to which customer. Now it can be seen that peer-to-peer VPN model will properly work only if the distinct address space is ensured across the customers’ and service provider’s network. Distinct address space is also required for traffic filtration where packets are filtered mainly based on their source and destination addresses. This reveals the problems with doing the default routing where customers points their default routes at each other having packets to bounce back and forward unless exact match is being implemented for the particular destination prefixes.

---

8 It doesn’t necessary means customers can’t use private address space. He would need to implement e.g. address translation on the CE in order to get portion of his network across the service provider’s network and then translate back at the opposite CE device.
3.2 Layer 3 MPLS VPNs

3.2.1 The BGP/MPLS VPN model

Layer 3 MPLS VPN implementation also known as BGP/MPLS IP VPNs or simply L3VPNs is the VPN model which combines the best from both worlds, the overlay VPN model and peer-to-peer VPN model. However it can be said BGP/MPLS VPN model was mainly inspired by the peer-to-peer model in order to also provide benefits of the overlay VPN model.

Following features describe the BGP/MPLS VPN model:

- Static provisioning isn’t required.
- Implementing new site doesn’t require additional configuration for others.
- Separate routing instances per customer. This feature allows customers to be no longer limited by the address space they can use.
- Traffic filtering is no longer required. With the separate routing instances per customer there is no possible way for a packet from one customer to leak to another.
- No restrictions for doing default routing.

To achieve this model two new components were introduced to the MPLS service provider’s network:

- Component for separation the customers’ routing information:
  - Virtual Routing and Forwarding (VRF) instance for keeping separate routing instances per customer.
- Component for the exchange of customer’s routing information:
  - Multi-protocol BGP which allows multiple address families to be transferred across the network in parallel.
  - MPLS which is used to label switch the traffic to the BGP next-hop.

---

9 This model was firstly developed by Cisco and later on it was standardized by the IETF.
3.2.2 Virtual Routing and Forwarding (VRF) instances

VRF concept isn’t necessarily directly related to MPLS or Multiprotocol BGP (MP-BGP). VRF simply means that additional VRF instance was created inside the device. VRF instance has absolutely nothing in common with others VRF instances or global routing instance unless specified through the routing policy. It can be imagined as a separate virtual router operating inside actual physical device. VRF instances are assigned per interface basics. It doesn’t necessary means whole physical interface can belong to only one VRF instance. Assignment can be also logical where the term interface can be interpreted as sub-interface, Frame Relay DLCI, ATM VCI/VPI or VLANs.

What VRF instances provide is the separation of routing information. They don’t guarantee the traffic isolation between instances. If inside one VRF instance exists an information how to reach prefix from the other there isn’t any mechanism inside VRF implementation which would prevent this from happening. Clearly there is a need for some mechanism which would control all the information being installed to VRF instances.

Simple example demonstrating how VRF implementation works can be seen in Figure 3-2. Customer 1 and customer 2 are using overlapping addressing scheme. Coincidentally service provider is using also same prefix range to address its network. Now with the VRF solution this is not a problem. It is required to have VRF instances being implemented all the way through service provider’s network with the dedicated interfaces for each VRF.
3.2.2.1 VRF implementation with the MPLS

The problem of having distributed VRF instances across whole service provider’s network can be easily overcome by deploying some sort of tunneling technique what can for example be MPLS. Having VRF instances implemented only on edge devices is absolutely enough as long as the actual traffic can be somehow encapsulated and delivered to the proper PE device. The most intuitive solution is to create full-mesh of tunnels between the PEs. Devices in the core don’t need to be aware of routing instances of the customers which makes separate interface per VRF no longer required for them. The customers’ traffic have already been classified and destined for the particular edge device where it will de-encapsulated and forwarded to the customer. This now allows for the core devices to be only able to forward the encapsulated payload to the edge. In other words they have to provide the ground for the tunnels being implemented across service provider’s network.

MPLS itself is a tunneling technique and its deployment is much easier than creating full-mesh topology of tunnels. For the MPLS to work is only required to have a fully reachable network running some IGP protocol. Then

![Diagram of VRF implementation for multiple customers](image3-2.png)

Figure 3-2 VRF implementation for multiple customers
it is just the matter of few commands to enable MPLS for the topology. Figure 3-3 demonstrates the MPLS tunneling technique helping to distribute customers’ traffic across the network.

Figure 3-3 VRF implementation for multiple customers with the MPLS technology

### 3.2.3 Distribution of routing information

It is now known how the routing information is kept inside the routers but so far it hasn’t been discussed the way of distributing and constraining all the routing information per VPN. There are two possible ways how to achieve secure distribution of routing information per customer or VPN. First way is to keep a routing information of each VPN within an instance of routing protocol. Clearly this solution is having issues with the scalability and management. For each VPN it is required to configure a new instance of a routing protocol. The management and processing all the information from numerous routing protocols just increase the complexity of this approach. Second possible and more convenient way is to deploy a MP-BGP.
3.2.3.1 Multi-protocol BGP

Following this approach we can distribute and manage all the routing information of VPNs within single instance. Multi-protocol BGP is an extension to the BGP that provides intelligence for distributing and constraining multiple address families. Into the same address family is put routing information we want the mutual communication for. On the other hand BGP is the only protocol which provides all the capabilities to support Layer 3 VPN model.

Here are provided some of them:

- Is a routing protocol operating at transport layer. Neighbor ship between devices is hold by the TCP session which allows them to don’t be directly connected. This fits the model where only PE devices need to communicate with each other.
- BGP is designed to be protocol operating across multiple Autonomous Systems (AS). Because of that it has a rich set of attributes which allows excellent control over the distribution of a routing information.
- BGP has built in filtering mechanisms allowing to provide required restrictions for VPNs.
- BGP is designed to be more suitable for distributing very large amount of routing information which only plays into the hands of supporting multiple customers.
- Multiple address families are supported within single instance of BGP.
- BGP is able to do label distribution for the MPLS.

3.2.3.2 Route Reflector (RR)

The architecture of BGP for the internal neighbor ship requires full-mesh of connections between all PEs. RR can be used in order to reduce this amount of iBGP connections. Following this approach RRs are fully meshed and rest of the BGP speakers within AS peer only with this RRs rather than with each other. Usually more the one RR exists within network topology to provide redundancy and resiliency. Therefore BGP speakers configured to be RR are essential to provide iBGP connections within AS.
3.2.3.3 Route distinguisher (RD)

For the environment where multiple routing domains co-exist there is a need to differentiate between routing instances of each of them. This is where the RD comes in. RD it is an 8 byte value prepended to the actual prefix within one VRF. This provides uniqueness for address space between the all routing instances. Therefore some procedure of allocating RDs to the routing instances has to be followed. It doesn’t really matter how RD is structured as long as allocating procedure ensures a unique number being prepended for each VRF instance. This is also because BGP is only able to distribute exclusive IP routes.

Figure 3-4 shows three different types of structure used for RD values. It is either the combination of arbitrary number and the AS number or the IP address. Usually the AS number of service provider network is chosen or the IP address of the PE router. Both of them are followed by the locally significant number. It just depends whether service provider wants the visibility of the overall VPN or they want the specific visibility as from which PE router was the specific prefix learnt.

<table>
<thead>
<tr>
<th>Type 0</th>
<th>AS number</th>
<th>arbitrary value</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 byte</td>
<td>2 byte</td>
<td>4 byte</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type 1</th>
<th>IP address</th>
<th>arbitrary value</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 byte</td>
<td>4 byte</td>
<td>2 byte</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type 2</th>
<th>AS number</th>
<th>arbitrary value</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 byte</td>
<td>4 byte</td>
<td>2 byte</td>
</tr>
</tbody>
</table>

Figure 3-4 RD different types of structure

Additionally to the RD there is a 4 byte IPv4 address as shown in Figure 3-5. This address has to be unique inside one VRF/VPN instance. The actual RD with the IPv4 address can be written 36500:400:10.0.3.0/24.
Variants of MPLS implementations

<table>
<thead>
<tr>
<th>RD value</th>
<th>IPv4 address</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 byte</td>
<td>4 byte</td>
</tr>
</tbody>
</table>

**Figure 3-5**  RD with IPv4 address

RDs are only present inside service provider’s network. They only serve to service provider to distinguish between multiple customers’ address spaces and thus there is no need to use them also with the customers.

### 3.2.3.4 Route Target (RT)

Whereas RDs provide the uniqueness for the address space across multiple VPNs the Route Targets (RT) are used to constrain the routing information. RTs are also known as BGP extended community. It is a 64 bit long field split in half where the first 32 bits are used for the AS number of service provider and rest 32 bits are for an arbitrary value. The whole combination of RD, IPv4 address and the RT provides a unique route information for the BGP and it can be written in a sequence where each value is separated by colon. For example 36500:400:10.0.3.0/24: 36500:999.

<table>
<thead>
<tr>
<th>RD value</th>
<th>IPv4 address</th>
<th>RT value</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 byte</td>
<td>4 byte</td>
<td>8 byte</td>
</tr>
</tbody>
</table>

**Figure 3-6**  Combination of RD, IPv4 address and RT

It is the primary task of VPN to provide secure and isolated environment. RTs can also be used to share paths among different VPNs. RTs allows service provider to create a complex routing policies in a flexible way such as having overlapping VPN connectivity and different kind of topologies among VPNs\(^\text{10}\). RTs basically control what enters and what exits the VRF table. RTs can be attached with an arbitrary granularity. For example one RT can be attached to more routes or more RTs can be attached to a single route. When routing policies are being configured the “export” and “import” statement are from the perspective of a local VRF table:

- Export RT – what routes will be exported from VRF table into BGP.
- Import RT – what routes will be imported from BGP into VRF table.

\(^{10}\) For example topologies like full-mesh, hub and spoke or central services VPNs where the service provider is hosting services (exchange servers, IP telephony and so).
Variants of MPLS implementations

The RT values with the import and export policies per VRF is where everything is tied together creating now complete distribution function for service providers’ network with multiple VPN customers.

The purpose of importing and exporting routes from the VRF table is shown in Figure 3-7 where the distribution of routing information is done between PEs within same VPN. Figure 3-8 demonstrates the logic of distributing routing information across VRF instances. Both VRF A and VRF B have a bidirectional import/export policy with the VRF C allowing them to communicate but yet still prohibiting the communication between VRF A and VRF B since no import/export policies were defined for them. In order to have a fully working model also logic from the Figure 3-7 has to be configured within same VRF instance on each PE.

Let’s recall the topology from the beginning of this section where there were configured two VRF instances with MPLS in the core. Figure 3-9 provides one of the examples how to configure constrained routing between two PEs in order to differentiate routes of all VPNs and ensure the uniqueness of each prefix. Both PEs are configured to be BGP speakers and maintain the iBGP session with each other. MPLS is in the core providing the logical
tunneling for the PEs. Each prefix from both VPNs is configured with RD and RT value. The blue color is used for RDs and green color for RTs. The actual prefix being carried in the BGP update message is in the middle. RD’s value is composed of loopback address of the PE (1.1.1.1 for PE1 and 2.2.2.2 for PE2) followed by the arbitrary number (100 for VPN A and 200 for VPN B). Now it can be determined which PE provides the connectivity for which VPN prefix. For the RT value the allocating method is pretty clear. Since the RT is the BGP community it is reasonable to configure it as the AS number of our BGP domain followed again by the arbitrary value (100 for VPN A and 200 for VPN B). Now each prefix exported into service provider’s network is uniquely identified by the RD and enabled for the constrained routing with the RT values. Finally it is up to each PE what will import into particular VRF instances based on the RT of given route.

Figure 3-9  RD and RT configuration for multi VPN environment
3.2.3.5 VPN tunnel

We already know how MP-BGP can be used for exchanging VPN routing information and how MPLS tunnels are used to provide PE to PE connectivity and thus path for the next-hop of VPN prefix. However so far it hasn’t been mentioned the way of identifying target VRF instance where the PE should do the IP lookup in order to deliver received MPLS packet to the correct VPN. To achieve binding between received packet and VRF instance is quite simple at the ingress PE just by applying VRF to the particular interface facing VPN site. Unfortunately same approach cannot be implemented for the egress PE. Fortunately the solution is again quite simple. Use another layer of MPLS which basically forms the VPN tunnel. In the BGP update message PE is also informed about the label to VPN association for each VPN prefix. This label never changes on the path and its only purpose is to carry the VPN information for the egress PE to which VRF instance the received packet belongs to. For the VPN label value allocation any policy will work which ensures separate label per VPN. This VPN tunnel is being carried inside regular MPLS transport tunnel where the regular forwarding process is being used.

Once packet reach the egress PE two forwarding techniques for delivering received packet to the VPN site can be implemented. VPN label is used to determine either:

- VRF table where the usual IP lookup is done.
- Outgoing interface where the packet is sent out.
Variants of MPLS implementations

The method of VPN label directly pointing to the outgoing interface seems to be more practical saving device one step to forward the packet. Incoming MPLS packet is stripped of all switching labels (MPLS headers) and immediately forwarded out of VPN site facing interface. However in some cases it is required to do forwarding decisions on the information found in IP header like DiffServ code in order to satisfy QoS requirements for the packet. In this scenario VPN label points to the appropriate VRF instance.

3.2.3.6 PE to CE communication

The communication between the PE and the local CE is essential for exchanging VPN routes. In order to achieve an exchange of routing information between CE and the VRF instance inside PE any dynamic routing protocol or static routes can be deployed. The purpose of this instance of routing protocol is to install routes learnt from VPN site into VRF table associated with interface facing this site. The real implementation is to have running separate context of routing protocol per VRF where the VPN routes are being redistributed back and forth with the customer’s VPN address family of MP-BGP.

Several aspects can be taken into account while deciding between routing protocols or static routes such as CE limitation, routing protocol on the CE, customer’s credibility and the required level of control over routes. Static routes are easy and every devices supports them but on the other hand they don’t provide any visibility or reachability information which might come useful or objective. If the high degree of control and scaling properties are required BGP is best fit. BGP was designed to support policy based routing and therefore it allows extensive handling of routing information [5].

3.2.3.7 VPN control and forwarding plane in action

For this VPN concept it is very important to realize the PE to PE logical connectivity must exists. This is because the VPN routes are advertised from the BGP speaker where this node acts as a next-hop for all his originated VPN prefixes. Therefore abstract path to the particular BGP speaker must exists. With the MPLS to this path is referred to as a tunnel. This tunnel is nothing more than a LSP between PEs. When this path becomes unavailable VPN traffic forwarding cannot continue until current path will be repaired or
Variants of MPLS implementations

alternate LSP will be established. In the chapter 5 thesis provides all different kind of techniques available in MPLS to mitigate the impact of network failures on customers’ traffic.

Figure 3-11 demonstrates in detail the forwarding plane of VPN packet and how is the control plane across service provider’s network maintained with the use of MPLS as transport layer and MP-BGP as control layer. Same topology scenario as before, two BGP speakers PE1 and PE2 interconnected across the MPLS core via LDP/RSVP tunnel. Topology now provides all information (including RDs and RTs from before) needed to demonstrate the forwarding process of the VPN packet.

Following section reveals the control plane and forwarding plane process in one direction. Control plane of VPN route advertisement from the egress PE to ingress PE with the intention of demonstrating the forwarding plane the of VPN packet for the VPN route learnt through the control plane process. Exact same process happens also for the opposite direction.

Control plane in action:

1) Egress PE receives an advertisement of the VPN prefix from the CE router inside the protocol used for PE to CE communication.
2) For a particular VPN prefix\(^\text{11}\) configured RD and RT are attached and VPN label is allocated.
3) MP-BGP update message is created containing VPN routing information with the label value to use in order to forward the packet out to the proper VPN site.
4) Egress PE sends the BGP update message out to MPLS transport tunnel.
5) Update is received at the ingress PE and processed by MP-BGP with the egress PE as the next-hop. Based on the RT VPN route is injected into proper VRF instance and RT is stripped from the VPN prefix. Also MPLS path for the particular next-hop has to be resolved in order forward VPN traffic to the egress PE.
6) VPN label is injected into MPLS forwarding table.
7) Ingress PE advertise the VPN route by the routing protocol used for PE to CE communication.

\(^{11}\) In this example VPN prefix represents the VPN route containing RD and RT. VPN route stands for just the actual IP route.
Variants of MPLS implementations

Forwarding plane in action:

1) VPN packet arrives at any VRF enabled interface facing the customer.
2) Based on the VRF association with the particular interface, the VRF table is determined for resolving the BGP next-hop of VPN prefix.
3) The result of lookup is the MPLS tunnel where another lookup in the MPLS forwarding table takes place.
4) The VPN label with the outgoing transport label are determined and packet is being forwarded out of MPLS interface with the transport label at the top and VPN label at the bottom with the bit indicating *bottom of the stack* set to 1.
5) Then MPLS packet is being forwarded across the core only based on the transport *label* value. VPN *label* never changes. Once packet reaches the egress PE it has only MPLS header with VPN *label* because transport *label* was popped at the preceding LSR.
6) Lookup based on the VPN label in the MPLS forwarding table happens to determine either the outgoing interface or VRF instance.
7) If configured, another lookup in the VRF table is done in order to properly handle the IP packet.
8) Once the packet handling is determined with the outgoing interface, the packet is stripped of the remaining MPLS header and forwarded as IP packet out of particular interface to the VPN site.
Variants of MPLS implementations


3.3 Layer 2 MPLS VPNs

The other type of VPNs available in MPLS are Layer 2 MPLS VPNs. The overall idea of layer 2 VPNs in MPLS is since we already have MPLS Layer 3 VPN infrastructure it can be used also for the transport of layer 2 payloads where they are directly tunneled across the existing IP MPLS network. The end result of direct encapsulation of layer 2 frame is that the CE devices appear from their perspective to be directly connected. Also the customers broadcast domain is now end to end. This is getting rid of the running CE to PE routing instance. This solution also provide huge flexibility in term of media we support transport for. It really doesn’t matter what type of layer 2 connection is being used with the customer. It can be either T-carrier (E-carrier), Frame Relay, ATM, PPP or Ethernet used altogether because the final result is they end up tunneled over MPLS network. For the connection between PE and CE in L2VPNs is referred as to an Attachment Circuit (AC). There are several factors leading to Layer 2 MPLS VPNs implementation:

- For L3VPNs some sort of communication between PE and CE is required. For L2VPNs only desired layer 2 technology for PE to CE communication is required.
- CE devices don’t have to be necessary IP aware. This allows for the customer to attach to the PE with just layer 2 capable devices.
- L2VPNs are transparent to the protocol used in network layer since it is transporting frames with the layer 3 payload.
- CE doesn’t support routing protocol offered by service provider.
- L2VPNs are more capable of providing connectivity when the customer requires service for IP unaware technologies such as ATM or Frame Relay PVCs.
- Interworking of different layer 2 technologies.

However the disadvantage is that the multiple logical or physical interfaces are required between the CE and PE one per target CE. With the L3VPNs only one interface was enough. Furthermore in order to support different kind of media coming from CEs service provider has to implement same layer 2 technology at the PE.
3.3.1 Layer 2 interworking

The great feature available in L2VPNs is the layer 2 interworking. Basically what it represents is the any-to-any layer 2 tunneling. Since it really doesn’t matter with what kind of media customer is attaching to the PE it allows to interconnect sites each using different kind of layer 2 technologies. The logic behind this is the IP packet is withdrawn from the layer 2 payload before entering service provider’s network. IP packet is then encapsulated into MPLS and forwarded across the core to the edge. Once the packet arrives to the egress PE it’s now being encapsulated into layer 2 transport technology used with that particular local CE. The example can be that one site is attached with Ethernet and the other with ATM which creates layer 2 interworking Ethernet vs. ATM. Essentially regardless how the customer is attached connectivity between sites can be formed.

Huge benefit of layer 2 interworking comes to the customers using all kind of layer 2 technologies. It may be they still supports different legacy solutions at different sites and still want to utilize them.

3.3.2 Layer 2 over MPLS transport principles

To a particular layer 2 connection between CEs across service provider’s network is referred as to a pseudowire. This is because the connection can be interpreted as a single wire while being formed by the MPLS. Multiple pseudowires are used to provide connectivity between customers’ CE nodes.

Figure 3-12 shows the L2VPN implementation where multiple CE sites are interconnected with each other by using so called pseudowires. The traffic is forwarded from the CE on the certain circuit which is being used to reach particular CE. Depending on the technology it can be a VLAN, PVC or DLCI. Separate circuit per destination CE site is required. Once packet reaches PE facing the target CE it has to change the circuit id for the id recognized by this destined CE as it belongs to a particular CE to CE pseudowire. For example the CE1 sends the packet for the CE3 on the circuit 200, the PE3 has to change this value to the 500 in order to tell the CE3 the packet belongs to the pseudowire between CE1 and CE3. It really doesn’t matter what technology is deployed for the link between PE and CE. In our case Ethernet VLAN would be translated into ATM PVC and vice versa.
3.3.3 Forwarding plane

The forwarding plane in the MPLS core practically does not differ between L3VPN and L2VPN. Core device doesn’t care to which VPN packet belongs. Even no additional configuration is required for them. Everything is configured on the PEs and it is them who are in control of each VPN network crossing the MPLS core. However the way of encapsulating customer’s payload isn’t the same.

To understand the main difference between L2VPN and L3VPN realization is the comprehension of layer 2 and layer 3 in their basics. Since the layer 3 connection is natively end-to-end the implementation of L3VPNs in term of providing true link is much easier. However the layer 2 connection by definition is only locally related to particular neighbors. Because of that there is a need for some element to go with the packet across the MPLS core in order to provide true end-to-end layer 2 connection.
Variants of MPLS implementations

Control Word (CW) is an additional 4 byte information being carried with the frame along the path. It is used for preserving original information from the layer 2 header which arrives at the ingress PE. This information is later being used by the egress PE in order to build layer 2 header for the link facing the CE. From the customer’s perspective now the connection appears as a true layer 2 link.

Forwarding plane in action:

1) Layer 2 frame is received at the ingress PE on a particular interface.
2) Ingress PE analyzes the layer 2 header and either:
   a. In case same layer 2 transport technology is used between sites it strips the unnecessary information from layer 2 header and prepares the whole frame for the MPLS encapsulation.
   b. When the sites are using different layer 2 technologies it takes out the entire payload encapsulated in the layer 2 frame and prepares for MPLS encapsulation.
3) If needed, CW is created and prepended right after the frame (or payload withdrawn from the frame).
4) Ingress PE determines the VPN and transport label and attach them to the rest respectively and forward the MPLS packet to the core. For the transport tunnel same LSP can be used as for L3VPN. Generally it doesn’t matter if transport tunnel is shared or not as long as it provides MPLS path between PEs.
5) MPLS packet is received at the egress PE. Depending whether PHP took place at node before it would need firstly to remove the transport label and then analyze the VPN label.

Figure 3-13  L2VPN forwarding plane with packet capture
6) Egress PE regenerates the layer 2 frame from the payload of MPLS packet and forwards it to the CE. If CW was present it would use its flags also to preserve the original information sent to the ingress PE.

### 3.3.4 Control plane

The signaling of individual *pseudowires* happens in the control plane of L2VPN. There exists two basis approaches. One is the use of original LDP signaling scheme and the other is the use of MP-BGP. Ultimately the signaling technique has no effect on the forwarding plane of L2VPN. The matter of L2VPN control plane is to provide:

- Signaling of VPN label egress PE expects.
- Providing end-to-end signaling for the *pseudowire* so remote sites can detect whether connection is working or not.
- Additional constraints such as MTU, type of the media and others.
- The impression the *pseudowire* is bidirectional connection. In case connection is broken in one way it has to make sure the whole connection goes down rather than having unidirectional path.

#### 3.3.4.1 LDP signaling

The first technique which was developed for signaling *pseudowires* involved LDP protocol. The principle is to manually configure targeted LDP sessions between PEs for each L2VPN in both directions. If the L2VPN is required between set of sites in the full-mesh fashion each two corresponding PEs have to be configured for emulating individual *pseudowire* between the sites. As it is obvious this scheme has its scalability limitations. Multiple *pseudowires* can exists between same PEs. Virtual Circuit (VC) id is used to differentiate between each signaled connection. Same VC id (VCID) is configured at both PEs for particular *pseudowire*.

Everything required for L2VPN connection to happen is carried inside this LDP session\(^\text{12}\) between PEs. Apparently the most important one is the VPN label. Further it is the VCID, bit indicating the use of CW, the type of

\(^\text{12}\) L2VPN targeted LDP session has nothing to do with the transport LDP session which can exists between PEs. It is just the use of LDP for signaling also information required for L2VPN.
the media customer connects with (PPP, ATPM, VLAN...) and additional parameters related to the particular layer 2 technology.

3.3.4.2 MP-BGP signaling

The MP-BGP signaling approach comes with the benefit of autodiscovery new sites and the possibility the MP-BGP is already being used in the network for maintaining L3VPN connections. The benefit of autodiscovery process is when a new site is added to the local PE, other PEs do not require additional configuration for this site to be a part of an existing L2VPN. This is because they learn about its existence through MP-BGP. Unambiguously BGP based signaling approach reduce this burden since pseudowires are now created automatically between each corresponding PEs.

Another advantage of MP-BGP signaling is the reuse of existing infrastructure present in service provider’s network between PEs. The problem of configuring L2VPN connection between remote sites reduce to only local site configuration at the PE. The way configuration works is assigning each local CE an identifier (CE id) which has to be unique within L2VPN. CE id is then used to create an association between AC and the CE. This way when a customer’s packet is received on the particular AC at the PE, PE knows to which CE packet belongs to and forward it in that direction. This is because PE is already aware of location of the remote CE through BGP session.

Everything required for a L2VPN connection to be operational is now carried inside the BGP update message. In addition to the BGP attributes (AS path, communities ...) update message includes the remaining of information related to L2VPN. It is the VPN label, bit indicating the use of CW, ids of all locally attached CEs and the rest of layer 2 parameters needed to create true separate end-to-end layer 2 connection.
### 3.4 Virtual Private LAN Services (VPLS)

The design of L2VPN solutions discussed so far were always point-to-point meaning each pseudowire represented a single broadcast domain. This made the whole concept of L2VPNs much easier to understand and configure. With the VPLS the whole machinery concept is more complex. It is the L2VPN which emulates a LAN services over the WAN. The main feature of VLPS is that the connections between PEs are multipoint. This allows multiple CEs to be a part of the same broadcast domain. The end result is that they attach to service provider network as to an actual switch.

![Layer 2 VPLS implementation with 3 different sites using different media](image)

**Figure 3-14** Layer 2 VPLS implementation with 3 different sites using different media

Since the VPLS network acts as a switch for individual CEs, the CAM table containing MAC addresses needs to be maintained at each node. This wasn’t required at point-to-point L2VPNs because the layer 2 address is not required to send a packet to the other end since there is only one recipient. PE has to inspect each incoming frame of particular site for the destination MAC address and forward it out corresponding pseudowire or local port depending whether remote site is locally attached or not. This means each PE has to have its own scheme for learning MAC addresses and associating them with egress PE. This is done by inspecting source MAC address of each incoming frame and binding it with the port on which frame arrived. Similar logic is with forwarding plane of a switch, where destination MAC address of incoming
frame is associated with outgoing port. In other words PE acts as a separate switch per VPLS instance.

### 3.4.1 Forwarding plane

VPLS data plane fully supports all standard MAC address operations like learning, flooding and aging in order to forward layer 2 traffic across MPLS network. The fundamental requirement for the VPLS topology is to have all PEs belonging to the same VPLS instance fully meshed. This requirement ensures loop free topology because each PE has now direct *pseudowire* to all PEs residing in same VPLS domain and thus can send traffic directly to egress PE. This approach eliminates the need for STP in order to maintain loop free network.

No matter which control plane mechanism is implemented, the final results is the same for either LDP based or MP-BGP based signaling. However the complexity of implementation is significant area where these two signaling techniques differ.

### 3.4.2 Control plane

#### 3.4.2.1 LDP signaling

The LDP signaling approach is very similar from the point-to-point L2VPNs. Since autodiscovery process is not implemented here manual binding between local site and egress PE, which creates a *pseudowire*, has to be configured. However for the VPLS it is required to configure full-mesh of *pseudowires* between all PEs within same VPLS domain. Process of provisioning new PE to the VPLS topology is quite complex because of missing autodiscovery process. All existing PEs have to be configured with the mapping of sites VCID to this new PE and vice versa new PE with the mapping of site VCID to the all PEs in VPLS domain. Figure 3-15 demonstrates the required amount of *pseudowires* which needs to be configured for LDP signaling in order to get new PE provisioned into VPLS instance.
3.4.2.2 MP-BGP signaling

The MP-BGP signaling approach is very similar from the point-to-point L2VPNs. It defines a means for a PE router to discover which remote PE routers are members of a given VPLS (autodiscovery), and for a PE router to know which *pseudowire* label a given remote PE router will use when sending the data to the local PE router (signaling). With the BGP-VPLS control plane, BGP carries enough information to provide the autodiscovery and signaling functions simultaneously [6]. Amount of configuration required to provision new PE into VPLS domain using MP-BGP based signaling is significantly less than for LDP based signaling. This is because of RR feature in BGP.
3.5 Conclusion

This chapter provided a knowledge of various implementations for MPLS technology. We have seen numerous of applications. MPLS is really the leader between transports technologies used in service providers’ environment. It is due to its architecture and interconnection with BGP. Chapter went through the most used implementations of MPLS as VPN service for the customers. Firstly with L3VPNs services, their architecture, functionality and basics components. Secondly it was L2VPNs similarly explained in order to get a full understanding of a problem. Finally very useful variant of L2VPNs which is VPLS was mentioned too to provide the full picture of MPLS VPN capabilities.
4  MPLS traffic engineering

4.1  Introduction

Having the control over the packet flow in the MPLS network is the matter of Traffic Engineering (TE) concept. Many reasons exist why is useful to influence packet flow in the network. The main reason is to have network resources wisely utilized avoiding congestions and underutilization of certain parts. Another reason is to ensure certain guarantees for LSP along the path it takes thorough the MPLS network. For example the LSP will use only low-latency links with and will get high priority which will make it more preferable to get new resources in case of their lack.

TE is not the necessity for the MPLS network. There are service providers which choose not to implement it and rather invest money into new connections. This solution works up to the certain point. The service guarantees can be only ensured by deploying additional individual network connections which provides the overall capacity much bigger than is really needed. This approach comes with a great cost and is practically unmanageable. TE deployment then offers service provider to increase revenue allowing them to save money spent on extra resources which are not really required. This is achieved with the extra work spent for building a MPLS network topology with embedded TE functionality. Now it allows service provider to have a control over each flow in the network. Which path it takes and how is served across the MPLS network.

4.2  Goals of TE functions

The goals of TE functions can be broken into 3 following sections:

- The need to forward specific traffic along predefined path.
- Improving the overall network resource utilization.
- Control over network resources in case of argument.
Forwarding traffic along predefined path

There are cases where it is required to forward certain traffic along specific path rather than leave the decision to IGP. Figure 4-1 provides an example of network topology with unequal link cost. There are low latency links but with low throughput and links with higher latency which have much higher throughput. Let’s have a request for a voice traffic that needs to be taken through our network. Usual scenario is to have IGP protocol such as OSPF running inside our service provider’s network to provide reachability between individual nodes. Then depending whether TE is implemented or not we can provide guarantees for this voice traffic. In case we haven’t deployed any control mechanism and LDP is used as distribution protocol for labels it is obvious path PE1-P3-PE2 will be chosen. This is because LDP follows IGP image of the network topology where OSPF shortest path from PE1 to PE2 is through giabit links no matter how high is the latency. This is unacceptable for the voice traffic.

We need to take control over this service by deploying TE. Shown network topology offers us still couple more available paths towards PE2 which will satisfy the need for the voice traffic. In order to do so explicit path has to be configured. It can be individually specified which hops to take through the network or which hops to avoid. The easiest solution would be to prohibit to use the link P3-PE2. Explicitly defining the path to follow PE1-P1-P2-PE2 would work also. Link P2-PE2 provides quite low capacity which isn’t the case for the voice while it guarantees certain delay and jitter.

![Network topology with unequal cost paths](image)

**Figure 4-1** Network topology with unequal cost paths
Another request might be for passing regular data flow of high throughput. This traffic can safely pass through the shortest path PE1-P3-PE2 hence for the regular data traffic delay isn’t the issue. Further in this chapter we will see more solutions how to ease this problem.

Utilization of network resources

Second area where TE finds its use is the utilization of network resources. This is essential to provide guaranteed service. As a service provider we cannot take a risk of a congestion in the network. It can lead to disruption of customers’ service delivery and violation of SLAs. Refer to the Figure 4-1 where certain available path of unequal cost between PE1 and PE2 exist. For such a network it is required to do bandwidth reservation to prevent congestion and underutilization. Suppose we need to create LSPs which require bandwidth guarantees of 500 Mbit/s, 400 Mbit/s, 300 Mbit/s and last one carrying only voice up to 100 simultaneous calls (approximately 6.4 Mbit/s of bandwidth). With the help of RSVP shown topology can utilize each traffic flow. 500 Mbit/s and 400 Mbit/s traffic flow along the path PE1-PE3-PE2, 300 Mbit/s along the path PE1-P4-P5-PE2 and voice traffic for example PE1-P3-P5-PE2.

Admission control of network resources

Next area of TE focus is to provide admission control of network resources. Let us have the same topology as in Figure 4-1. Suppose 2 LSP A and B are crossing the network on the path PE1-P3-PE2. LSP A is 300 Mbit/s and LSP B is 200 Mbit/s and requires strict service guarantees. In case of link failure P3-PE2 a detour is created following the path PE1-P3-P5-PE2 which doesn’t correspond of enough bandwidth to cover both LSP. Without admission control of resources congestion and packet loss would occur. In order to protect LSP B it has to have higher priority over LSP A. Now it will be just the LSP B which will get the alternate path towards PE2 and LSP A stays down. This way LSP A won’t interfere with LSP B which requires strict service guarantees.
4.3 Setting up TE paths

The process of setting up TE paths can be divided into two parts. Firstly the path computation according given constraints and RSVP-TE attributes. Secondly forwarding the traffic over such a computed LSP.

4.3.1 LSP priorities and preemption

In order to differentiate LSP with stricter guarantees, concept of LSP priorities was introduced into MPLS-TE. Each LSP can be configured with setup priority and hold priority. Priority value is moving from 0 to 7 where 0 is considered as the best and 7 as the worst priority. Setup priority is relevant when the LSP is being established and hold priority when it comes to the conflict with other LSP. The principle of two priority values is that it allows to confiscate resources of present LSP which holds priority is worse than the setup priority of new LSP. This action is called preemption. It is said that new LSP preempted existing one.

LSP priorities allows to prioritize more important LSP in two different scenarios. Firstly while setting up the path and secondly when failure occurs. This way this more important LSP is still given service over other less important LSP. Network resources can be still divided by any LSP.

4.3.2 Distribution of TE information

TE related information has to be somehow distributed to all nodes in service provider’s network. For this purpose extensions for existing IGP protocols were created allowing them to carry MPLS-TE information along with the link state. Both link-state protocols IS-IS and OSPF provide those extensions.

MPLS-TE Information:

- Bandwidth.
- Administrative attributes – link colors.
- TE metric.
- Maximum hop count.
- Setup priority of LSP.
MPLS traffic engineering

This way each node has all information related to MPLS-TE locally present and stored in TE Database (TED) [5].

4.3.3 Link coloring

The process of finding shortest path via CSPF can be constrained by the use only of links belonging to the specific administrative group or groups. Administrative groups are configured through RSVP and their purpose is to differentiate links one from another. It is like coloring the links with specific colors. Same color is used for the links belonging to the same administrative group. It is customary to configure color as the name of administrative group. Once administrative groups are created they are applied to RSVP interfaces. One or more color can be assigned to RSVP enabled interface. MPLS-TE allows to include, exclude or ignore links with specific color. Up to 32 different colors can be used in the topology.

![Diagram of administrative groups structure](image)

**Figure 4-2** Example of administrative groups structure

**Figure 4-2** provides an example of link coloring in the MPLS-TE network. Constraints for CSPF path computation can be as follows:

1) For primary LSP use default links (links with no color).
2) For local protection in the core use default links:
   a) with combination of BLUE links in case PEs are 3 hops away.
   b) with combination of RED links in case PEs are 4 hops away.
3) For *end-to-end protection* use any combination of GREEN, RED and BLUE links.
4.3.4 CSPF

Link-state IGP use Shortest Path First (SPF) algorithm to determine shortest path in the network. RSVP uses a modification of that algorithm called Constrained SPF (CSPF) which allows path computation to be influenced by additional constraints. All the constraints are maintained in TED which provides current MPLS-TE topology information.

While determining which path to select, CSPF follows these rules [7]:

1) LSPs are computed one at a time, beginning with the highest priority LSP then with the LSPs with the highest bandwidth requirements.
2) Prunes all links that:
   a) do not have sufficient reservable bandwidth.
   b) do not share any included colors.
   c) contain excluded colors. Links without color assignment are accepted.
3) Finds the shortest path toward the LSP's egress router, taking into account any ERO. For example, if the path must pass through Router A, two separate SPF algorithms are computed: one from the inbound router to Router A and one from Router A to the outbound router.
4) If several paths have equal cost:
   a) chooses the one with a last-hop address the same as the LSP's destination.
   b) selects the path with the least number of hops.
   c) applies CSPF load-balancing rules configured on the LSP.

4.3.4.1 Path reoptimization

Path at the time of computation may not be the most optimal one. CSPF computation relays on underling information from IGP. Depending whether network topology information was the most current one or not, suboptimal LSPs can exist. Another case is when the failure occurred and traffic have been transferred to alternate path. Now the failure is cleared and it is required to move the traffic back to the primary path. To avoid the case when the traffic is bouncing between primary and alternate LSP path reoptimization is done by default in the certain time frame. Path reoptimization countdown timer is same for all LSP on a given device. It would be inefficient to maintain one
timer per LSP. Remember that the forwarding plane for LSP has to be maintained regardless LSP is optimal or not. Switchover from suboptimal to optimal path has to happen without any traffic loss. This means when more optimal path is found, it has to be also established in forwarding plane. After switchover the suboptimal LSP is torn down. This switchover approach is known as make-before-break.

4.3.5 Selection of TE paths

The last portion of implementing MPLS-TE path is to make the device to choose them. From the forwarding point of view LSP is treated as an outgoing interface. Metric is also associated with it. It can be IGP metric of the underlying path or MPLS-TE metric. It depends on the configuration. The simplest way is to configure static routes to use LSP as outgoing interface for specific prefix. This isn’t very manageable approach since it requires manual interaction with paths.

Another option is the choice of BGP where destination prefixes are installed into routing table with the next hop address of egress LSR. If the LSP (considered now as interface) exists for particular next hop address of given prefix, it will be used to forward the traffic. This behavior is crucial for BGP/MPLS VPNs.

If it is allowed through the configuration LSP paths can be used together with IGP routing to determine shortest path to the destination. This allows MPLS-TE to be applied to a portion of the network and include LSP in the SPF calculation. In order to propose LSP as a candidate for SPF computation process it has to be allowed on the LSP head end. If we want to use LSP further in the network it has to be advertised through IGP.

4.4 MPLS DiffServ-TE

The problem of IntServ model is that it requires signaling protocol (e.g. RSVP) to let others know which flow requires special QoS treatment. DiffServ model solves this by directly implementing class type into packet header. It allows to configure 64 different types of classes through the 6 bits of DiffServ Code Point (DSCP). For the IP packet field Type of Service (ToS)
MPLS traffic engineering

is used to encode DSCP values. Packet handling is determined by each node separately based on IP packet header. This is known as Per Hop Behavior (PHB).

The problem with MPLS is that packet forwarding is done based on the label value in the header of the MPLS packet. This creates a potential issue with achieving PHB for the LSP. Fortunately MPLS header contains 3 EXP bits (see Figure 2-1) which can be used for carrying DiffServ information in the MPLS header. This now allows nodes on the path to take into account DSCP along with label value while determining PHB for the MPLS packet. However this solution has its caveat. 6 bits are used to address 64 different types of classes where MPLS header has available only 3 EXP bits. To address this problem there exists two approaches:

a) No more than 8 PHBs will be supported in the MPLS-TE – in this case nothing is done and DSCP values are directly mapped to EXP bits.
b) More than 8 PHBs will be supported in the MPLS-TE – label value with combination of EXP bits is used to determine PHB. This solution requires be conveyed at the time of signaling LSP.

Table 4-1 sums up the main differences between EXP signaled LSP (E-LSP) and Label-inferred LSP (L-LSP) [8].

<table>
<thead>
<tr>
<th>DiffServ with only EXP bits : E-LSP</th>
<th>DiffServ with label value and EXP bits : L-LSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHB is determined only from the EXP bits.</td>
<td>PHB is determined from the label or from the label along with EXP bits.</td>
</tr>
<tr>
<td>Can carry traffic with up to 8 distinct PHBs in a single LSP.</td>
<td>One or more PHBs per LSP with the same scheduling regimen but different drop priorities.</td>
</tr>
<tr>
<td>Label is used only for advertising path information.</td>
<td>More labels states since they are used also for encoding PHB.</td>
</tr>
<tr>
<td>PHB distributed without signaling.</td>
<td>PHB needs to be distributed with signaling.</td>
</tr>
<tr>
<td>Only up to 8 PHBs can be supported in the network for E-LSP. When more than 8 PHBs are required E-LSP can be used in conjunction with L-LSP.</td>
<td>Any number of PHBs can be supported in the network.</td>
</tr>
</tbody>
</table>

Table 4-1  Comparison of E-LSP and L-LSP
Failover functionality in MPLS

Path protection and restoration is a key element in MPLS networks. Providers sell their services to customers. This service delivery is concluded by a contract where certain SLAs are specified. In order to provide reliable service delivery we have to make sure our network is immune to different kind of network outages and instability. We also have to consider variety of application we provide service for. Application which require high quality service need to be treated differently than application which don’t. Voice and video are usually referred to a “fragile traffic” because of their real-time nature. Such a traffic requires immediate handling in case of outage. They can’t recover from the higher traffic loss using retransmissions as regular data flow would do. There are even more vulnerable services to the network instability and traffic loss than voice and video such as haptic applications. They deliver haptic feedback over the network and here even very small peak in a delay is not acceptable. SLAs for voice and video services are quite strict against usual data transport and even stricter for haptic applications. Therefore we need to protect such a path. Techniques which are used to do so will be described in this chapter.

5.1 Introduction

We already know that MPLS works on many layer 2 technologies which may or may not come with their own way of protection. We need to consider whether it is relevant to implement another layer of protection or we can safely relay on what was already implemented somewhere else. One of the most common failures is link failure. For example SONET or SDH provides a protection at physical layer using the Automatic Protection Switching (APS) where backup link is maintained and ready to take over as soon as failure is detected. Since the detection happen on the device itself and no further convergence is required, switchover can happen within 50 ms. Such a short disruption is almost unnoticeable for the application layer. The cost we have to pay for it is maintaining backup link and reserving bandwidth for it with utilizing addition hardware for the switchover [5].
Similar idea was adopted by MPLS. They named it Fast Reroute (FRR). Here we have MPLS tunnels instead of one physical link. However everything else remain the same. We need to have a backup tunnel established and maintained in hardware. This solution needs to be differentiated from the path protection where on the head end we have configured another LSP to provide the backup for the primary one. Remember the idea behind very fast switchover is in local decision and pre-computed path which is immediately ready to take over once failure is detected. In order to signal new end-to-end LSP we need to propagate failure to the head end of LSP, which takes some additional time. After that the particular device can disclaim broken LSP and switch to the backup one. Now it is clear that the main purpose of the FRR is to provide temporary LSP recovery till the new LSP will be ready to take over using different physical path than was the primary. While using FRR we no longer need to maintain backup LSP since now we have the time to calculate and build the new LSP.

Another concept which needs to be taken into account is how fast restoration we need. In many cases we do not really need to restore traffic within 50 ms. The diversity of application we provide service for reflects to different requirements for the paths. Therefore we need firstly to classify the traffic, e.g. if loss is tolerated or how much can be tolerated, the maximum accepted delay, whether bandwidth needs to be guaranteed and many others. For example for the voice loss of 300 ms and more is noticeable. Having loss over 1 - 2 seconds can disrupt control traffic and cause the drop of the call. If loss over 3 seconds occur it may influence the IGP protocol to re-converge the network topology [5].

Protection and restoration is very expensive but service providers wouldn’t survive consequences of not implementing one.

### 5.2 Path protection

Path protection also known as end-to-end protection is one of the essential protection service providers can offer. Primary LSP is backed up by another LSP between the same source and destination but using different physical path. It is one of the most common practice of providing resiliency. Under normal conditions only primary LSP is used for the traffic. While the primary
LSP is operating without any disruption secondary LSP is carrying nothing. Depending on the configuration secondary LSP can be already pre-signaled and maintained in hardware which greatly reduce restoration time but with the cost of reservation idle resources. Second option is new path can be signaled on the fly once head end is informed about the failure on the primary LSP. Let’s have a link failure to occur between P3 and D as shown in Figure 5-1. Notification is done by PLR, router P3, simply by propagating RSVP PathErr back to the source.

In order to provide appropriate end-to-end protection service provider has to consider various aspects of such a protection and its requirements:

- **Resource wasting** – if the path is pre-signaled, same amount of resources we are using for the primary LSP is staying idle for the backup LSP. When resources start to be questioned this wasting can be avoided by simply signaling the path in the time of failure. This solution comes with its price. The total restoration time may increase couple times. Another thing to consider is what will happen when new primary LSP path needs to be built and there isn’t enough resources to accomplish so due to idle reservations of backup LSPs in the network. To address this problem LSP preemption was created. It simply assigns priority to the LSP and in case of argument, LSP with lower priority is torn down with the purpose of releasing its resources for the LSP with higher priority.
Failover functionality in MPLS

- Traffic flow control – having path pre-signaled comes with great advantage of knowing where exactly the traffic will flow. With this knowledge we can guarantee there will be enough capacity and secondary path will meet all requirements of the primary. This goal can be also achieved by explicit path configuration.

- Path diversity requirement – it is essential that primary and secondary LSPs do not share any link or device between source and destination. If this cannot be obtained providers risk the case where both LSPs fail to form.

- Nondeterministic switchover delay – the RSVP error needs to be propagated from PLR to the head end of the path. Every device on the path back to the source is involved in this propagation. This is the matter of control plane and it is absolutely not guaranteed every device is ready to serve this request by immediate actions.

- Needless protection – whole path from the head end to the tail end is protected. End-to-end protection cannot by applied to certain sectors and therefore if path has others recovery mechanisms they can’t be mutually excluded.

It is not only in the networking environment when we try to protect certain paths. We may find many similarities every day. This end-to-end protection can be compared with the situation where we are about to go to our destination and it is announced our primary highway there is blocked somewhere in the middle. We haven’t yet entered the highway so we choose different one in order to avoid delay. Next section will describe cases what will happen to those who are using broken path [5].

5.3 Local protection

Many today’s applications are that sensitive it isn’t enough to only relay on path protection. Switchover time can be still unacceptable even if we have backup path pre-signaled and maintained in hardware. For this reason local protection was introduced. It is based on very simple idea. Provide a fast bypass path very near around the failure point. As it was mentioned in real life example in previous section what would happen to those who haven’t been warned about the broken path they are already using? Solution is
Failover functionality in MPLS

obvious. They will use shortest bypass path around the point where the road is blocked. Place where the main path is left is called Point of Local Repair (PLR) and the place where it back rejoins Merge Point (MP). In other words PLR represents device from which traffic is locally rerouted and MP is device where it all merge together and continue using main path.

![Figure 5-2 FRR with local protection](image)

Intuitively this can concluded to be a temporary solution where we don’t want to affect the traffic which is still present on primary LSP by dropping it and black holing next one for certain period. FRR was designed to provide temporary but very fast path recovery. Temporary till the point when the head end of the path is informed about the failure on the primary LSP and starts switching the traffic using the backup LSP.

Local protection using FRR comes with following advantages:

- It is resource related. It protects a single resource in the network which can be a single link or a whole device. This makes it easy to understand and have fast deployment.
- It can provide shortest possible bypass path around the failure point. This in the end makes path recovered very fast with the minimum devices involved for the recovery.
- Bypass path is pre-computed and maintained in hardware ready to be used immediately failure is detected.
- When FRR is present it isn’t necessary to maintain backup LSP. There is plenty of time to signal and build backup LSP.
Failover functionality in MPLS

FRR protection can work in two different modes. It can either protect every single LSP separately or it can protect the complete bundle of LSPs. To a single LSP protection is referred to as 1:1 protection and to the bundle as N:1 protection or facility protection. This with the link and node protection makes in total 4 variants of FRR.

The process of implementing local protection into existing MPLS network can be broken into 4 chronological sections as shown in Figure 5-3.

5.3.1 Pre-failure configuration

The advantage of local protection is freedom of choice for which LSP it will provide protection for against certain resource inaccessibility. Service provider can choose which services are going to receive this kind of protection and which aren’t. With this approach more vulnerable traffic, like voice, can get better thresholds for failover to happen than other less important traffic. Another benefit is that already protected resources can be excluded from the protection. In order to provide FRR, head end of protected LSP firstly needs to request a local protection. This information will be propagated along the LSP using the RSVP Path message with the flag “local protection desired” set on either in the Session Attribute Object (SAO). This way PLR is told which LSPs need to be protected. Further PLR is configured for which resources is going to provide FRR.

Figure 5-3  Chronological process of achieving local protection
5.3.1.1 Backup path computation

Once everything is configured a PLR computes a backup tunnel running CSPF computation with the next hop of protected resource as destination with simple restriction, which is to avoid using this protected resource. Head end can limit a backup path with certain criteria using FRR Object (FRO) in RSVP message. Limitation can apply e.g. for hop count, required bandwidth, hold priorities and others. These constraints assure the backup tunnel provides sufficient resources for the protected traffic in order to avoid sudden congestion making traffic to be dropped. It is also demanded to provide certain guarantees while traffic is being switched over protection tunnel.

In case of 1:1 protection it is not needed to specify which links color or bandwidth to use. These attributes are inherited from the protected LSP [5].

5.3.1.2 Forwarding state installation

Forwarding states for protection tunnel are installed just like they are for main LSP. MP is informed that is the tunnel endpoint so it will inform device upstream on the tunnel path what label it expects in order to successfully merge protection tunnel with protected LSP. Two different techniques exist to do so. MP will receive a MPLS packet with the same label as it would receive from PLR under normal conditions where the preceding device either popped the label or did the regular swap. This will be explained more in depth in each section of FRR protection variants further in this chapter.

5.3.2 Failure detection

While providing recovery the very first step is failure detection. The time period in which we are able to detect a network failure is essential and is given by two main factors.

One of them is the distance between the point where failure occurred and the object we need to report to about it. In case of local protection the detection happens on the closest device to the break point and it is only matter of milliseconds to detect it. If the failure needs to be reported somewhere further in the network, the overall detection time starts increasing approximately by 100s of milliseconds.
Second factor is whether the failure detection is done in hardware or needs to be processed by software. In many today’s scenarios the detection is built in hardware where the break in transmission at physical layer is detected within couple milliseconds. Having the detection in hardware has many advantages. However not a small price is paid for it. Here we have a dedicated hardware responsible just for the detection. Imagine if we would need to go to the software layer. There is still a chance some obstacles may arise. It is not guaranteed processor will be ready to serve the request at that time. It can be busy with processing and generating updates, handling different threads or simply preferring another task.

When the hardware doesn’t come with failure detection built in, this task can still be handled by the entity in the upper layer. Classic example is the convergence of the IGP protocols. Link connection is maintained by sending periodic hello packets between the neighbors. When hello packets stop arriving from the neighbor, failure on the link is assumed. Usually it takes three times the period of hello packet. The minimum values which can be configured for these timers are still unacceptable for fast detection. Even with the fact hello packets need to be handled by processor which isn’t simple task making the frequency of CPU to raise. Under the conditions where time plays key role those two factors need to be well evaluated.

5.3.2.1 BFD hello protocol

Bidirectional Forwarding Detection (BFD) is a network protocol used to detect faults between two forwarding engines connected by a link. It provides low-overhead detection of faults even on physical media that don't support failure detection of any kind, such as Ethernet, VCs, tunnels and MPLS LSPs [9]. It is a simple process running on the device which only role is to provide handling of the hello packet for some underlying protocol. BFD doesn’t come with any discovery mechanism. It is meant for the protocol to keep it simple and easy to deploy. That is why it needs to be explicitly configured for link protection on both endpoints.

One of the simplest solutions for providing fast recovery is to have a LDP protocol running in the core of MPLS and since LDP paths follows the IGP, ensure the recovery on that layer. With the BFD managing the hellos we can tune up the failure detection to be around 100s of milliseconds. It may appear
the given time for the detection is quite high in the comparison with SONET/SDH but yet still enough for numerous of the applications.

5.3.2.2 Failure detection mechanisms embedded in physical layer of a particular technology.

This is the case of local alarm being generated for a particular technology and thus either solving connectivity issue inside this technology or handing over this issue to upper layers. Perfect example of solving problem locally is SONET/SDH where APS takes care of restoring connection by switching over to a protected link or ring. Typical example of handing over problem away from technology itself is Ethernet. It does not at all provide any protection and traffic restoration resulting into additional intelligence has to be implemented in order to provide one.

5.3.2.3 Keepalives messages exchanged on point-to-point links.

Couple different encapsulation standards exist for maintaining point-to-point links. These links tend to maintain their connection state by exchanging periodic messages called keepalives. Once messages stop arriving at the end of point-to-point link, this is considered as a failure on the link has occurred.

5.3.2.4 RSVP hellos extension.

This is the extension to the RSVP protocol with the similar idea as keepalives for the point-to-point protocols. Here the RSVP protocol has a direct mechanism to provide node-to-node failure detection by exchanging periodic RSVP hellos between them. This solution is often used when RSVP is already implemented as label distribution protocol and link layer doesn’t provide sufficient values for failure detection. Failure detection values provided by RSVP hellos are lower than values provided by IGP protocols. They move around couple hundreds of milliseconds which is sufficient for most implementation of local protection.
5.3.3 Connectivity restoration

Once the PLR is aware of failure on the protected LSP it can immediately starts forwarding protected traffic over the backup tunnel or tunnels. This to happen takes almost no time since the backup tunnels are already maintained in hardware and ready to serve. There exists 4 different approaches how backup tunnel can be built with respect to a protected resource. All of them are deeply explained later.

5.3.4 Post-failure signaling

5.3.4.1 LSP teardown suppression and LSP head end notification

Very important task is to suppress LSP teardown which happens once failure is detected via IGP advertisements. This is done by suppressing any error generation at the both ends of LSP tunnel. Otherwise generated error would lead to the LSP teardown and making the local protection pointless. LSP head end needs to be notified about the failure on the path via RSVP even if it may eventually find out about it from IGP. The core network on the IGP level can be structured into different areas or autonomous systems which would lead to the case where head end will never find out about the failure.

Default behavior of RSVP-TE when failure occurs is to inform head end of each protected LSP by propagating path error message (PathErr) message using error code “routing problem” (24) with flag “no route available” (5) toward destination. By the following this default approach whole concept of local protection would be useless. Obviously different technique has to be applied for LSP head end notification. LSP teardown suppression is an approach where PLR notifies LSP head end for each protected LSP using error code “notify error” (25) in the RSVP-TE PathErr message and flag “path locally repaired” (3) set on in RRO. This then results the protected traffic wouldn’t be black-holed for a period of a time when new path is being calculated and established as it would happen by default. In case where at least one head end of protected LSPs couldn’t successfully form new path, local protection remains still present avoiding termination of particular LSP13.

---

13 Sometimes to these two different techniques is referred just by using error codes 24/5 and 25/3 respectively.
Failover functionality in MPLS

However LSP can be still terminated by the tail end of the LSP in case it is told LSP has been broken. This can happen in 3 different ways. It may receive IGP update, PathTear message or wouldn’t receive RSVP Path refresh messages for a certain period of time. It has become clear that also PathTear message needs to be suppressed. This can be simply achieved by fooling the device which eventually will become MP by transmitting path refresh messages belonging to the failed RSVP session over the backup tunnel. Device doesn’t check on which interface it received path refresh message for particular RSVP session with the neighbor and this way there wouldn’t even be a need to generate PathTear message. Having MP to continue receiving path refresh messages also prevents from stopping them being generated on the way further downstream [5].

Where to this process fits IGP notification about the failure? Well it depends whether FRR is configured for particular LSP or not. In case it is we already know RSVP-TE ignores any IGP notifications and protected LSP can be torn down only by using RSVP-TE PathErr message. When FRR isn’t implemented and head end receives IGP notification about network error on the certain LSP it will tear down that LSP and tries to form a new path for it.

5.3.4.2 New path establishment

As it was described in chapter 5.3 backup LSP can be established offline, before the path break occurs or as soon as LSP head end is informed about the failure. Either way path is computed in a same manner using make-before-break approach. Depending on the traffic engineering restriction made to the core, the backup path is computed. If there are no restrictions and primary LSP is locally protected, backup LSP can share resources with the primary LSP. This knowledge is indicated by shared explicit descriptor in the RSVP Resv message. Now if the network topology will look like as shown in Figure 5-4 it is possible that the backup LSP is able to share resources of the primary LSP for path computation. For this short time the bandwidth for certain traffic is utilized twice. When all the LSPs with the configured local protection have been moved to the backup path the main LSP can be torn down and free its resources.
5.3.5 Data plane

Data plane is pretty straightforward. Source device receives a packet which needs to forward. It will do a destination routing lookup and check if any label is associated with particular FEC and in case of match received packet is given an MPLS header with embedded found label and processed by LFIB as labeled packet.

Figure 5-4 LSPs flow under normal conditions

Figure 5-4 shows MPLS network topology with three LSPs established between source and three different destinations. For easier understanding suppose the metric is a hop count. Then under normal conditions all 3 LSPs follow shortest path to their destination. Path through devices S-P3-P4 is shared by all of them.

5.3.6 Control plane

Control plane is much more complicated than data plane. There are certain tasks which need to happen in order to create a stable forwarding plane. The backup path has to be computed and pre-signaled before the failure occurs which means the forwarding state must be installed at the all transit nodes including PLR and MP. To a LSP which is being protected may be referred as to a protected LSP. Figure 5-5 shows LSP process establishment between the source and destination with FRR protection. Device P4 is PLR because it
switches the traffic to the protection tunnel in case of failure and P5 is the MP where the traffic is joined back to the protected LSP.

5.3.7 Link protection

Link failures in the network are quite a common thing. They tend to happen much more often than node failures what makes link protection one of the most favorite fast recovery solutions. However there exists a common solution to strengthen connection between nodes where multiple links are aggregated into port-channels. This is mostly done for the connection between core devices in MPLS network where traffic load is huge. Having link aggregation in place mitigates the impact of link failure between nodes. It also avoids to trigger local protection mechanism since logical unit is the port-channel rather than single link.

![Figure 5-5 LSP establishment with protection tunnel for link P4-P5](image)

Link protection relies the next hop device on the LSP path is working properly and will try to establish bypass tunnel to it using CSPF calculation. In case of link protection whole next hop device would become unavailable both protected LSP and bypass tunnel won’t forward any traffic. The behavior of setting up bypass path varies whether it has been chosen protection of 1:1 or N:1 model. If we recall the example with our accident at the highway and the bypass route around the car crash point path to the destination can be completed in two ways. It can join main path immediately after the crash point or it can have new different path to the destination which does not necessary needs to merge with the main one.
5.3.7.1 Link protection using N:1 model

Data plane

Facility protection provides a bypass tunnel for all LSPs passing PLR which share same next-hop device. This tunnel is created by PLR very around the failure and ends on the MP device which was the original next hop.

Assume link between P3 and P4 goes down as shown in Figure 5-6. In this scenario device P3 has now a role of PLR creating bypass tunnel around the link failure. Note that PLR sees for all 3 LSPs device P4 as a next-hop. Given than only one bypass tunnel will be built using lowest metric of 2 hops to the MP. Note the fact that the given bypass path for LSP1 is not optimal using total hop count of 5 instead of shortest possible 4 through link P1-P2.

Control plane

For the facility protection PLR node has a backup tunnel built above any other LSP. This tunnel is used on the top of existing LSPs which are configured for protection adding another label. Tunnel is signaled just like any other LSP. Specifically in this case for the tunnel path MP advertise upstream device label 3, also called an implicit null label. Tunnel is terminated at PLR where first downstream node signals PLR label 101 for the protection tunnel usage.
Failover functionality in MPLS

From the scalability point of view, PLR just keeps one forwarding state for the protection tunnel use no matter for how many LSPs N:1 local protection will be provided. MP node does not need any new forwarding state updates done to the LFIB. When MPLS packet arrives to the PLR, forwarding plane of the device needs with addition to swap the label also push new one which is being switched by LSR providing the bypass tunnel. Node in front of MP does penultimate hop popping which results in same packet arrival to the MP as it would as it would arrive directly from PLR except on different port. Once packet arrives to the MP the following packet processing remains unchanged from the packet processing of protected LSP and the packet is delivered to the destination.

**Figure 5-7** demonstrates such a case through P4-P5 link failure. P4 has the role of PLR and MP of protected LSP and bypass tunnel is at P5. Node P2 does the penultimate hop popping. Note that the packets sent by P2 and P4 are identical.

### 5.3.7.2 Link protection using 1:1 model

**Data plane**

For the 1:1 protection model PLR provides each protected LSP individual tunnel to avoid link failure and continues with the shortest\(^\text{14}\) path to the destination. It is obvious using this approach we can have multiple MP for the same PLR. To these individual tunnels are commonly referred as to the detour

\(^{14}\) This does not necessarily means the shortest path will be used. It can join the protected LSP right away or follow completely different path to the destination.
paths. For the demonstration same topology will be used as in N:1 model in order to easily spot the differences.

Figure 5-8 shows again P3-P4 link failure. With the facility protection provided backup path for LSP1 wasn’t optimal. In this model LSP1 can use the shortest possible path to its destination calculated from the PLR’s perspective. This improvement comes with the cost of signaling separate detour path for each protected LSP and maintaining more forwarding states compared to N:1 model.

**Control plane**

For the 1:1 protection model the PLR node disrupts the protected LSP and builds a new tunnel path which is linked with LSP we are protecting. Same process is applied to every LSP which requires local protection for certain resource in case this resource becomes unavailable. This scenario does not scale very well in case of protection many LSPs. Every LSR involved in providing these detour paths has to maintain extra forwarding states for each protected LSP. When MPLS packet arrives to the PLR, forwarding plane of the device is going to interconnect existing LSP with its detour path. PLR in swap action simply chooses label advertised from the device downstream on

---

15 Each LSP needs to have own labeled path which makes number of new forwarding states results in function of count of passing nodes used for the detour path.
the detour path over the label signaled by MP. This certain detour path has its own specific label installed in LFIB at MP which will be swapped with label used for the rest of the existing path.

![Figure 5-9](image.png)

**Figure 5-9** Forwarding process of LSP with backup tunnel in case of 1:1 protection

In **Figure 5-9** MPLS packet is again being switched from P4 over protection tunnel in case of P4-P5 link failure and merging back to main LSP at P5. Formation of protection tunnel starts at P5 which sends a new label 502 to preceding LSR on already pre-calculated detour path. Tunnel is terminated at PLR, node P4, which in case of P4-P5 link failure will be using label advertised by P1 for forwarding the protected traffic. However for 1:1 protection model this bypass tunnel is logically being injected in the middle of LSP which results in no change in label stack for the MPLS packet.

### 5.3.8 Node protection

Sometimes it is vital to use node protection rather than link protection. Service provider’s network topology also requires to protect against node failures, especially in the core where is crucial to have redundancy. Node failure also beats the link aggregation since other end is not responding at all. The problem of link protection in previous section is overcome by configuring the local protection against the whole node instead of link. Here the CSPF calculation with try to establish bypass tunnel to the device on the LSP right after protected node. Node protection also comes in two different variants of providing the protection against node using 1:1 or N:1 model approach where the model logic remains exactly same as for link protection.
5.3.8.1 Node protection using N:1 model

Data plane

Facility protection against node in the topology provides a bypass tunnel created by PLR for each group of protected LSPs which have the same downstream next hop right after the node they are being protected against. Since it is no longer about the protection of one local link which have the same end for all protected LSPs the next next hop (device after protected node) can vary depending on the path protected LSP follows.

Figure 5-10 is showing an example of topology where next hop is different between LSPs in case of P4 failure. This is because there no longer exist available path from P5 to D1 or from P2 to D1 and D2. Node P3 has the role of PLR and the protected traffic is merged at two different MPs. For LSP1 at P2 and LSP2 and LSP3 at P5. Once traffic is merged it continues to its destination as usual. As it is known new path does not guarantee LSP will follow the shortest path to its destination. LSP3 reaches to P5 only because tunnel ends there and has to travel back to P8 in order to reach D3.
Failover functionality in MPLS

**Control plane**

Similarly to the control plane of link protection PLR builds a backup tunnel for each protected LSP group which have same MP above any other LSP in the topology. Those tunnels then give extra layer of labels for the bypass segment of protected LSP. The control plane operation remains practically same as in case of link protection except one obstacle. PLR has to know two additional pieces of information in order to build the backup tunnel and correctly forward traffic in case of failure [5].

1) The address of MP device. This information is needed to run CSPF computation at PLR using MP address\(^\text{16}\) as destination. Address\(^\text{17}\) of MP device can be obtained from the RRO where it is specified as a *loose hop* for reaching the MP.

2) The label which was used to reach the MP from the preceding device, failed node, for each of the protected LSPs. This is done similarly as the discovery of the downstream node where label of MP should be recorded in RRO. By default this information isn’t present in RRO and it has to be requested by setting the flag *label recording desired* in SAO.

![Figure 5-11 Forwarding process of LSP with backup tunnel in case of N:1 protection](image)

\(^{16}\) Tunnel path is calculated using MP address as destination with the restriction to avoid the resource we are protecting against.

\(^{17}\) This address has to something what can be interpreted by IGP as destination in order to run algorithm for path computation. It can be address of particular interface or router ID.
For the previous example in Figure 5-10 it is necessary to let PLR know about the forwarding labels being used by P4 for each protected LSP. Once PLR is aware it can firstly do a swap operation using these obtained labels of P4 and then push the label for the tunnel. Process of distributing the labels and forwarding actions is demonstrated in Figure 5-11. P4 has role of PLR but MP is now at P6. Device P2 is doing penultimate hop popping what results MP receives MPLS packet just like it would come from failed node P5. Since MP is one hop to the destination it does too penultimate hop popping and sends pure IP packet to the destination.

5.3.8.2 Node protection using 1:1 model

Data plane

It has already been mentioned that for using 1:1 model approach a separate detour path has to be created for each protected LSP. Detours created by separate PLRs can merge together if they protect the same LSP. This makes the overall number of detours per LSP to don’t be as high as it might look like at the beginning.

![Figure 5-12 LSPs flow for 1:1 protection in case of failure node P4](image)

18 For the explanation purpose IP protocol was chosen to be at layer 3. With MPLS it does not matter what layer 3 protocol is being encapsulated.
**Figure 5-12** shows an example of a network topology where PLR is providing 1:1 model protection for each LSP. In case of P4 node failure separate detour paths are put into use creating totally 3 different MP for each one of them. It is not required for the detour path to merge with main LSP. As it was mentioned early in this section LSPs created by different PLRs can merge into one detour path in case they protect same LSP. Assuming LSP2 is protected against link failures between nodes P3-P4 and P4-P5. Instead of having two separate detours P3-P1-P4 and P3-P6-P7-P8-P5 we can keep only the second one in order to provide protection against both link failures.

**Control plane**

For node protection control plane logic of 1:1 model protection model remains same as for link protection. PLR provides bypass tunnels by disrupting existing LSPs and linking them with detours which don’t necessary have to merge with main LSP. They can just follow shortest path to their destination from the PLR’s perspective.

**Figure 5-13** Forwarding process of LSP with backup tunnel in case of 1:1 protection

**Figure 5-13** demonstrates tunnel signaling and traffic forwarding in case of P5 node failure. Tunnel starts at P4 which in case of P5 node failure will be using label advertised by P1 for forwarding the protected traffic and terminates at P6 where MP is expecting label 602 in order to successfully merge detour path with the rest of main LSP.
5.4 Additional constraints for providing protection

5.4.1 Fate sharing

If MPLS network provides either an end-to-end or local protection it seems like a sufficient way to ensure resiliency to the existing topology. Well the problem of fate sharing hasn’t been discussed yet and it may seem from the resiliency point of view maximum was accomplished. What would happen if both protected LSP and alternate path in case of failure would share same fate by being susceptible to a particular event causing both of them to fail? Yes, they would be torn down or better to say protected LSP would be torn down with a backup tunnel and new end-to-end LSP would fail establishment if it wasn’t pre-signaled. Typical example is where none of the links are shared but they share one or more LSR. Another case would be they would share same physical path and single even like a soil erosion would cause both links to be torn apart. To the resources which share same “fate” are referred as to a single Shared Risk Link Group (SRLG) or sometimes called fate-sharing group.

Once resources has been classified to which SRLG group they belong to it is quite simple to implement logic ensuring none of the resources from the protected LSP and tunnel providing backup or new end-to-end LSP would share same group. From the computation point of view this is just another restriction to the CSPF computation process.

This approach may look very familiar to link coloring discussed in MPLS-TE section in chapter 4.3.3. However it is not the same concept. While the link coloring is completely static restriction such as to avoid all red links for path formation the fate-sharing group is a dynamic restriction because the restriction to avoid particular fate-sharing group for a backup path depends whether the protected path is currently using resources of same fate-sharing group or not. Another big difference is that restriction of fate-sharing group doesn’t completely prohibit formation of backup tunnel or new end-to-end LSP in case some of the resources are shared. Otherwise the resulting effect would be the alternate path will fail its formation as it would happen using the link coloring method. This way alternate path can be still formed but it will be less preferred since it shares one or more resources with the same “fate” [5].
Membership of a particular fate-sharing group can be subsequently configured for a certain resource or IGP can be used to distribute this information among the devices.

### 5.4.2 Bandwidth protection

While providing the connection over service provider’s network some applications may require additional guarantees to be applied onto leased connection. In some cases it is right for both parties to be bound by some rules in order to define certain boundaries between successfully providing service and case where they aren’t. The most used one is the ensuring certain level of bandwidth per connection. For a service provider is useful to know how much bandwidth allocation each LSP requires. This information helps to enable such a network where it is easy to deploy new connection while still guarantying service for the existing ones. Service provider can then estimate which links are lacking bandwidth and which part of the network requires upgrade of the connections.

Bandwidth protection is the ability to continue guarantying bandwidth for the protected traffic in case the original LSP would need to be rerouted. Does this implicate the whole concept of local protection is useless without bandwidth guarantees? The answer is no. There are several reasons why.

- Local protection is temporary solution for protecting traffic which usually provide reroute just for couple seconds till new end-to-end LSP is established.
- Many service providers ignores the fact of bandwidth protection for bypass tunnels because of additional effort to accomplish it and harder troubleshooting.
- Average utilization of the links is up to 50% of bandwidth. When this limit is exceeded usually the link capacity is increased. This provides plenty of space for FRR without necessity to also protect a bandwidth.
- For the most of the cases this is satisfactory. There is low probability of some peak in data transport for such a short period of time when local protection is in use.

Bandwidth protection is achieved by ingress node at the time of forming LSP and signaling it with “bandwidth protection” flag in SAO. In case there is a space to provide bandwidth protection PLR responds with flags “local
Failover functionality in MPLS

Failover functionality in use” and “bandwidth protection available” set up in RRO. This way head end is able to determine whether desired protection for certain LSP was acquired or not.

Alternate path keeps same amount of bandwidth as protected LSP. For 1:1 protection it is straightforward to provide a backup tunnel with the direct knowledge of required bandwidth. However for a facility backup using same approach would contradict with its purpose. Rather than following the idea of 1:1 protection an approach with the exactly opposite logic is followed. A backup tunnel or several backup tunnels are established around the protected resources with specific bandwidth and admission control takes care which LSP will be provided which backup tunnel based on its bandwidth requirements. Following this reversed approach has become very attractive because the estimation of overall bandwidth required for backup tunnels cannot cross the available bandwidth of the protected link. Another advantage is that multiple backup tunnels can be built to satisfy the bandwidth need in case certain resources become unavailable and each protected LSP will be given one of the available backup tunnels as seen in Figure 5-14. It is important to realize that the traffic of protected LSP cannot be split across multiple backup tunnels. Doing so would cause packets to arrive out of order at the egress LSR. This entire process of creating backup tunnels is automated by some platforms where:

- Bypass tunnels are automatically created with sufficient bandwidth needed for all protected LSP or maximum possible.
- Optimization of LSP division into tunnels with the best fit.
- Freeing allocated bandwidth with disposal of bypass tunnels once they are not needed.

Figure 5-14 provides a good example how important it is to optimize the allocation of bypass tunnels to protected LSPs. There are 3 different LSPs with different bandwidth requirements. In case of failure shown LSR two backup tunnels were created to provide reroute and satisfy bandwidth need for all of them. Tunnel1 provides bandwidth of 200 Mb and tunnel2 of 60 Mb. If tunnel assignment would be done without any optimization it could happen some LSPs would be rejected to get into backup tunnels because of the lack of the bandwidth. When for example LSP1 and LSP3 would get tunnel1 there is no enough bandwidth in tunnel2 to accommodate LSP2.
Failover functionality in MPLS

However by doing optimization best fit can be found and thus protecting all LSPs. As demonstrated tunnel1 for LSP1 and LSP2 and tunnel2 for LSP3.

![Diagram](image)

**Figure 5-14** Bandwidth protection and spreading protected LSPs across multiple backup tunnels

### 5.4.3 Scalability

When deploying any kind of protection to the network important factor to consider is scalability. Each protection scales differently according to actual network topology. It is then mandatory to evaluate given network topology prior implementing the protection to network. Couple various factors like different kind of technology present in network, average degree of links per node, structure of LSP connection across network and potential network growth have to be taken into account [5].

In order to provide appropriate picture I will assume different kind of network topologies consisting of various numbers of nodes with certain scope of link degree. This will then serve as source for evaluating all 3 kinds of protection available in MPLS networks.
Failover functionality in MPLS

Figure 5-15: Example of service provider network for Australia and New Zealand.
Network topology shown in Figure 5-15 provides a typical example of moderate service provider network. In order to give a topology physical ground I choose it to be a service provider network for the part of Australia and New Zealand providing connectivity for 4 different regions in Australia and 2 different regions in New Zealand. All this geographical areas are interconnected by the MPLS core network consisting of P nodes. In each region there are PE routers providing connectivity to customers. Roughly all core devices have the degree of local links to be in a range between 3 and 5. Green area consists only of core MPLS devices doing just packet switching. They aren’t involved in traffic engineering at all. Purple area are devices which do run TE for the MPLS core and provide and maintain redundancy within this area between each other. Usually network is configured to have logical full-mesh connection on this level. Finally the edge area is the section where devices usually maintain logical connection between customers and provide TE for individual services for the customers. This structure isn’t somehow generally defined or instructed to follow. It is simply a good practice which most of the good service providers try to follow. Having network structured and organized it provides lot of benefits and advantages for the network operators to configure, maintain and troubleshoot the problems.

The scalability of path protection is simple matter. For the end-to-end protection of primary LSPs same amount of backup LSPs is needed. From the point of scalability this is the worst case and is rarely being implemented. It is much more convenient to implement local protection because it scales far better. There isn’t a formula which would find exact result for any kind of topology. However it is possible to find good approximation which provide results quite close to the reality. Following approximation formulas for protection comparison are taken from this book [1].

\[
D = \text{average degree of local links per node.}
\]

\[
R_n = \text{number of nodes involved in mesh topology.}
\]

\[
\text{Number of primary LSPs in full mesh} = R_n \times (R_n - 1)
\]

\[
\text{Number of LSPs needed for Link protection} = R_n \times D
\]

\[
\text{Number of LSPs needed for Node protection} = R_n \times D^2
\]
With the intention to simplify calculation for network example provided in Figure 5-15 suppose $R_n$ is all devices with green and red glare which have full-mesh of LSPs between each other. Then $R_n = 25$. $D$ moves between 3 to 5, rarely it is 6 or 2. Then it is safe to assume $D = 4$.

*Number of primary LSPs in full mesh* = $25 \times (25 - 1) = 600$

*Number of LSPs needed for Link protection* = $25 \times 4 = 100$

*Number of LSPs needed for Node protection* = $25 \times 4^2 = 400$

*Number of LSPs needed for Path protection* = 600

Table 5-1 provides an overview of an average number of LSPs for each protection depending on a number of nodes involved in the mesh topology. For a small number of primary LSPs the difference between scalability isn’t that big. However with the growth of $R_n$ exponentially grows the amount of LSPs and with it LSPs needed for path protection.

<table>
<thead>
<tr>
<th># Nodes</th>
<th># Primary LSPs</th>
<th># LSPs for Path protection</th>
<th># LSPs for Link protection</th>
<th># LSPs for Node protection</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>380</td>
<td>380</td>
<td>80</td>
<td>320</td>
</tr>
<tr>
<td>40</td>
<td>1560</td>
<td>1560</td>
<td>160</td>
<td>640</td>
</tr>
<tr>
<td>60</td>
<td>3540</td>
<td>3540</td>
<td>240</td>
<td>960</td>
</tr>
<tr>
<td>80</td>
<td>6320</td>
<td>6320</td>
<td>320</td>
<td>1280</td>
</tr>
<tr>
<td>100</td>
<td>9900</td>
<td>9900</td>
<td>400</td>
<td>1600</td>
</tr>
<tr>
<td>120</td>
<td>14280</td>
<td>14280</td>
<td>480</td>
<td>1920</td>
</tr>
<tr>
<td>140</td>
<td>19440</td>
<td>19440</td>
<td>560</td>
<td>2240</td>
</tr>
<tr>
<td>160</td>
<td>25440</td>
<td>25440</td>
<td>640</td>
<td>2560</td>
</tr>
<tr>
<td>180</td>
<td>32220</td>
<td>32220</td>
<td>720</td>
<td>2880</td>
</tr>
<tr>
<td>200</td>
<td>39800</td>
<td>39800</td>
<td>800</td>
<td>3200</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th># Nodes</th>
<th># Primary LSPs</th>
<th># LSPs for Path protection</th>
<th># LSPs for Link protection</th>
<th># LSPs for Node protection</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>9900</td>
<td>9900</td>
<td>400</td>
<td>1600</td>
</tr>
<tr>
<td>200</td>
<td>39800</td>
<td>39800</td>
<td>800</td>
<td>3200</td>
</tr>
<tr>
<td>300</td>
<td>89700</td>
<td>89700</td>
<td>1200</td>
<td>4800</td>
</tr>
<tr>
<td>400</td>
<td>159600</td>
<td>159600</td>
<td>1600</td>
<td>6400</td>
</tr>
<tr>
<td>500</td>
<td>249500</td>
<td>249500</td>
<td>2000</td>
<td>8000</td>
</tr>
<tr>
<td>600</td>
<td>359400</td>
<td>359400</td>
<td>2400</td>
<td>9600</td>
</tr>
<tr>
<td>700</td>
<td>489300</td>
<td>489300</td>
<td>2800</td>
<td>11200</td>
</tr>
<tr>
<td>800</td>
<td>639200</td>
<td>639200</td>
<td>3200</td>
<td>12800</td>
</tr>
<tr>
<td>900</td>
<td>809100</td>
<td>809100</td>
<td>3600</td>
<td>14400</td>
</tr>
<tr>
<td>1000</td>
<td>999000</td>
<td>999000</td>
<td>4000</td>
<td>16000</td>
</tr>
</tbody>
</table>

Table 5-1 Path, link and node protection scalability numbers for moderate and large amount of nodes
Failover functionality in MPLS

Following graphs in Figure 5-16 show dependence between number of primary LSPs and amount of links required for certain protection.

Figure 5-16  Path, link and node protection scalability graphs
6 Implementation of MPLS network

This section of thesis is going to provide implementation of MPLS network under two different vendors. Since the network is going to be used only for study purposes and measurements it is convenient to create a LAB. Mostly emulation techniques will be used to build a network. This is because it is easier to deploy such a network and maintain it. Emulation has both positive and negative impact on LAB environment. Huge benefit of emulation as opposed to the real implementation is the price. Having devices emulated and keeping whole network topology in one processing environment is almost at no cost. However there are certain limitations. VPLS isn’t possible to implement.

6.1 Conclusion

I have successfully configured service provider MPLS network under vendors Cisco and Juniper. Network environment was for both vendors virtualized. I also attended a LAB with physical devices at Karel English College but unfortunately couldn’t get MPLS running on devices which were available in the LAB because the IOS required a license in order to allow MPLS to be even configured. However I still was able to implement almost everything except VPLS in the virtualized environment of GNS3.

Network scenarios between Cisco and Juniper slightly differ in order to provide network implementation based on different kind of techniques and protocols.

For the testing purposes of failover functionality available in MPLS I choose Cisco LAB environment. I preferred Cisco over Juniper mainly because Cisco LAB in GNS3 is able to support whole failover functionality concept available in MPLS.

Configuration of both LABs and results from the testing of failover functionality are properly documented on the following pages. I tried to be as diverse as possible while measuring different kind of recovery mechanisms implementation. Results are provided through the graphs.
6.2 Cisco LAB

Network environment for Cisco LAB is implemented in GNS3 application. It’s a well know emulation environment used for implementing different kind of LAB scenarios without any kind of real hardware as routers, switches, firewalls and so.

Lab provides implementation of the L3VPN and point-to-point L2VPN service with interworking. Unfortunately GNS3 does not support VPLS so I wasn’t able to implement it in the topology.

List of the protocols running MPLS-TE VPN network:

- OSPF – for MPLS core reachability.
- LDP – for full-mesh of connectivity across MPLS core.
- MPLS-TE – TE tunnels to provide TE functionality to the network.
- RSVP – to signal LSP which requires protection.
- RSVP hello protocol – for fast detection of link failure.
- VRF – in order to isolate individual customers and implement L3VPN service.
- MP-BGP – protocol running the whole L3VPN concept. Providing the control plane for all routing information inside the network.
  - iBGP – for PE to PE sessions.
  - eBGP – for CE to PE sessions (CUSTOMER2).
- EIGRP – protocol for communication with the CE (CUSTOMER1).
- LDP L2VPN signaling – communication between PEs to provide L2VPN service for particular Site to Site connection.
Figure 6-1: Cisco MPLS LAB topology for demonstrating failover functionalities.
6.2.1 Configuration

6.2.1.1 PE

Configuration for both PEs is almost the same. Thus whole configuration only for PE1 will be explained here. PE2 has the mirrored version.

PE1

# Hostname configuration
hostname PE1

# Loopback interface configuration
# Loopback0 is the interface used for local identification of a router in the network topology. It is used by OSPF, BGP, MPLS (LDP). LSP are configured from 1 loopback to another loopback of a router. Loopback 1 simulating network being advertised into BGP topology.
interface Loopback0
  ip address 1.1.1.1 255.255.255.255
  ip ospf network point-to-point
  ip ospf 1 area 0

interface Loopback1
  ip address 50.0.0.1 255.0.0.0

# Physical interface configuration facing MPLS core
# Configuration is almost identical for each physical interface in the MPLS network. It may vary depending on the actual requirements of policy and MPLS-TE (metric, policy, access-lists, ...).
interface GigabitEthernet1/0
  ip address 10.0.7.1 255.255.255.0

interface GigabitEthernet2/0
  ip address 10.0.1.1 255.255.255.0

interface GigabitEthernet3/0
  ip address 10.0.10.1 255.255.255.0

# Physical interface configuration facing L3VPN customer
# Configuration needs to specify VRF for the customer.
interface GigabitEthernet5/0
  vrf forwarding CUSTOMER2
  ip address 20.0.3.1 255.255.255.0

interface GigabitEthernet6/0
  vrf forwarding CUSTOMER1

Required for BGP reachability between PEs.

Applying VRF CUSTOMER2 for the CE router.
Implementation of MPLS network

ip address 20.0.1.1 255.255.255.0
!
# Configuration of IGP protocol for core reachability
# OSPF is being configured to provide reachability through the core and to
# support carrying MPLS-TE extensions.
#
router ospf 1
  mpls traffic-eng router-id Loopback0
  mpls traffic-eng area 0
  router-id 1.1.1.1
  log-adjacency-changes
  auto-cost reference-bandwidth 100000
  network 10.0.0.0 0.255.255.255 area 0
!
# MPLS configuration
# Contains global settings for MPLS and configuration for the interfaces to
# enable MPLS on them. It has also enabled RSVP hello protocol on interfaces for
# the fast detection of failure.
#
ip cef
mpls traffic-eng tunnels
mpls traffic-eng reoptimize timers frequency 30
mpls label protocol ldp
mpls ldp router-id Loopback0
  ip rsvp signalling hello
  !
interface GigabitEthernet1/0
  mpls traffic-eng tunnels
  mpls ip
  ip rsvp signalling hello
  !
interface GigabitEthernet2/0
  mpls traffic-eng tunnels
  mpls ip
  ip rsvp signalling hello
  !
interface GigabitEthernet3/0
  mpls traffic-eng tunnels
  mpls ip
  ip rsvp signalling hello
  !
#
# VRF configuration
# Configuration of VRF for the customer. Specify RD and RTs for export and
# import.
#
vrf definition CUSTOMER1
  rd 1.1.1.1:100
  route-target export 36500:100
  route-target import 36500:100
  !
  address-family ipv4
  exit-address-family
Implementation of MPLS network

vrf definition CUSTOMER2  
rd 1.1.1.1:200  
route-target export 36500:200  
route-target import 36500:200  
!  
address-family ipv4  
ext-address-family  
!  
# BGP configuration  
# Configuration of MP-BGP for peering between PEs. Since topology is quite small RR aren’t necessary. BGP speakers peer directly with each other. Logically MP-BGP configuration is divided into 3 sections. First one is for backbone peering between PEs and regular IPv4 reachability. Second section is for VPN support. Enables peering of PEs providing VPN connectivity and configure to send extended community to the specific neighbor. Last section is for VPN configuration. It is used for redistributing routes from the customer into BGP in order to get them to the egress PE which will redistribute them back to the instance of routing protocol running with the VPN customer.  
###

router bgp 36500  
no synchronization  
bgp log-neighbor-changes  
network 50.0.0.0  
neighbor 2.2.2.2 remote-as 36500  
neighbor 2.2.2.2 update-source Loopback0  
no auto-summary  
!  
address-family vpnv4  
neighbor 2.2.2.2 activate  
neighbor 2.2.2.2 send-community both  
ext-address-family  
!  
address-family ipv4 vrf CUSTOMER1  
redistribute eigrp 100  
no synchronization  
ext-address-family  
!  
address-family ipv4 vrf CUSTOMER2  
neighbor 20.0.3.103 remote-as 65100  
neighbor 20.0.3.103 activate  
no synchronization  
ext-address-family  
!  
# Configuration of IGP protocol running with the customer  
#  
# This IGP protocol is used to speak with customer CE and provide reachability to his prefixes.  
###

router eigrp 1  
auto-summary  
!  
address-family ipv4 vrf CUSTOMER1  
redistribute bgp 36500 metric 1500 4000 200 10 1500  
network 20.0.0.0  
no auto-summary  
autonomous-system 100  
ext-address-family  
!
Implementation of MPLS network

# LSP configuration
#
# This is the configuration of the LSP tunnel. In cisco LSP is being created through the tunnel interface.

```conf
interface Tunnel1
  ip unnumbered Loopback0
  ip ospf interface-retry 0
  tunnel mode mpls traffic-eng
  tunnel mpls traffic-eng autorumote announce
  tunnel mpls traffic-eng path-option 10 explicit name TUNNEL1_PRI
  tunnel mpls traffic-eng path-option 15 explicit name TUNNEL1_BCK
  tunnel mpls traffic-eng path-option 20 dynamic
  tunnel mpls traffic-eng record-route
  tunnel mpls traffic-eng path-selection metric igp
  tunnel mpls traffic-eng fast-reroute
  no routing dynamic
```

# Explicit path configuration
#
# Configuration of explicit paths for the primary path and secondary path.

```conf
ip explicit-path name TUNNEL1_PRI enable
  next-address 3.3.3.3
  next-address 5.5.5.5
  next-address 6.6.6.6
  next-address 2.2.2.2

ip explicit-path name TUNNEL1_BCK enable
  next-address 7.7.7.7
  next-address 2.2.2.2
```

# L2VPN configuration
#
# Contains configuration of individual pseudowires and logical interfaces where bidirectional mapping is set up for sites between PEs.

```conf
pseudowire-class SITE1-TO-SITE2
  encapsulation mpls

pseudowire-class SITE1-TO-SITE3
  encapsulation mpls
  interworking ethernet

interface GigabitEthernet5/0.12
  encapsulation dot1Q 12
  xconnect 2.2.2.2 12 pw-class SITE1-TO-SITE2

interface GigabitEthernet5/0.13
  encapsulation dot1Q 13
  xconnect 2.2.2.2 13 pw-class SITE1-TO-SITE3
```
Implementation of MPLS network

PE2

interface Tunnel0
  ip unnumbered Loopback0
  ip ospf interface-retry 0
  tunnel destination 1.1.1.1
  tunnel mode mpls traffic-eng
  tunnel mpls traffic-eng autoroute announce
  tunnel mpls traffic-eng path-option 3 explicit name EXCEPT-P4-P5
  tunnel mpls traffic-eng path-option 5 dynamic
  no routing dynamic
  !
  ip explicit-path name EXCEPT-P4-P6 enable
  exclude-address 4.4.4.4
  exclude-address 6.6.6.6
  !

6.2.1.2 P

Configuration for all Ps is almost the same. Thus whole configuration only for P3 will be explained here. Remaining Ps have the very similar version of configuration. Additionally P which acts as PLR have configured backup path for local protection.

P4

hostname P4
  !
  ip cef
  mpls traffic-eng tunnels
  mpls traffic-eng reoptimize timers frequency 30
  mpls label protocol ldp
  mpls ldp router-id Loopback0
  ip rsvp signalling hello
  !
  interface Loopback0
    ip address 4.4.4.4 255.255.255.255
    ip ospf network point-to-point
    ip ospf 1 area 0
    !
  interface GigabitEthernet1/0
    ip address 10.0.2.4 255.255.255.0
    mpls traffic-eng tunnels
    mpls ip
    ip rsvp signalling hello
    !
  interface GigabitEthernet2/0
    ip address 10.0.3.4 255.255.255.0
    mpls traffic-eng tunnels
    mpls ip
    ip rsvp signalling hello
    !
  interface GigabitEthernet3/0
    ip address 10.0.6.4 255.255.255.0
    mpls traffic-eng tunnels
    mpls ip
    ip rsvp signalling hello
    !
  router ospf 1
Implementation of MPLS network

mpls traffic-eng router-id Loopback0
mpls traffic-eng area 0
router-id 4.4.4.4
log-adjacency-changes
auto-cost reference-bandwidth 100000
network 10.0.0.0 0.255.255.255 area 0

P3 (PLR)

interface Tunnel52
  ip unnumbered Loopback0
  tunnel destination 6.6.6.6
tunnel mode mpls traffic-eng
tunnel mpls traffic-eng path-option 10 explicit name TUNNEL1_BCK(Node5)
tunnel mpls traffic-eng path-selection metric igp
  no routing dynamic
  !
  ip explicit-path name TUNNEL1_BCK(Node5) enable
  next-address 4.4.4.4
  next-address 6.6.6.6
  !
interface GigabitEthernet3/0
  ip address 10.0.4.3 255.255.255.0
mpls traffic-eng tunnels
mpls traffic-eng backup-path Tunnel52
mpls ip
  ip rsvp signalling hello
  !

P5 (PLR)

interface Tunnel51
  ip unnumbered Loopback0
  tunnel destination 6.6.6.6
tunnel mode mpls traffic-eng
tunnel mpls traffic-eng path-option 10 explicit name TUNNEL1_BCK(Link5)
tunnel mpls traffic-eng path-selection metric igp
  no routing dynamic
  !
  ip explicit-path name TUNNEL1_BCK(Link5) enable
  next-address 3.3.3.3
  next-address 4.4.4.4
  next-address 6.6.6.6
  !
interface GigabitEthernet1/0
  ip address 10.0.5.5 255.255.255.0
  negotiation auto
mpls traffic-eng tunnels
mpls traffic-eng backup-path Tunnel51
mpls ip
  ip rsvp signalling hello
  !

Configuring physical interface to use Tunnel S2 as a backup path in case of failure node P5.

Configuring physical interface to use Tunnel S1 as a backup path in case of link failure P5-P6.
Implementation of MPLS network

6.2.1.3 CE

Configuration of CE is very simple. It is configured just to peer with particular PE through EIGRP and thus advertise and learn prefixes through it. It is service provider’s job to do the routing for L3VPN.

HQ

hostname HQ
!
interface Loopback0
  ip address 101.101.101.101 255.255.255.255
  ip ospf network point-to-point
!
interface Loopback1
  ip address 151.0.0.1 255.0.0.0
!
interface FastEthernet0/0
  ip address 172.21.0.1 255.255.255.0
duplex auto
speed auto
!
interface GigabitEthernet1/0
  ip address 20.0.1.101 255.255.255.0
negotiation auto
!
router eigrp 100
network 20.0.0.0
network 101.0.0.0
network 151.0.0.0 0.255.255.255
network 172.21.0.0
no auto-summary
!

CE1

hostname CE1
!
interface Loopback0
  ip address 103.103.103.103 255.255.255.255
!
interface Loopback1
  ip address 153.0.0.1 255.0.0.0
!
interface GigabitEthernet1/0
  ip address 20.0.3.103 255.255.255.0
negotiation auto
!
router bgp 65100
no synchronization
bgp log-neighbor-changes
network 20.0.3.0 mask 255.255.255.0
network 103.103.103.103 mask 255.255.255.255
network 153.0.0.0 mask 255.0.0.0
neighbor 20.0.3.1 remote-as 36500
neighbor 20.0.3.1 allowas-in
no auto-summary
!
Implementation of MPLS network

Site1

hostname Site1
!
interface GigabitEthernet1/0.12
  encapsulation dot1Q 12
  ip address 3.0.0.1 255.255.255.0
!
interface GigabitEthernet1/0.13
  encapsulation dot1Q 13
  ip address 4.0.0.1 255.255.255.0
!

Site2

interface FastEthernet0/0.12
  encapsulation dot1Q 12
  ip address 3.0.0.2 255.255.255.0
!

Site3

interface FastEthernet0/0
  ip address 4.0.0.2 255.255.255.0
!

6.2.2 Reachability documentation

HQ

HQ# show ip interface brief

<table>
<thead>
<tr>
<th>Interface</th>
<th>IP-Address</th>
<th>OK? Method</th>
<th>Status</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>FastEthernet0/0</td>
<td>172.21.0.1</td>
<td>YES NVRAM</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td>FastEthernet1/1</td>
<td>unassigned</td>
<td>YES NVRAM</td>
<td>administratively down</td>
<td>down</td>
</tr>
<tr>
<td>GigabitEthernet1/0</td>
<td>20.0.1.101</td>
<td>YES NVRAM</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td>GigabitEthernet2/0</td>
<td>unassigned</td>
<td>YES NVRAM</td>
<td>administratively down</td>
<td>down</td>
</tr>
<tr>
<td>GigabitEthernet3/0</td>
<td>unassigned</td>
<td>YES NVRAM</td>
<td>administratively down</td>
<td>down</td>
</tr>
<tr>
<td>GigabitEthernet4/0</td>
<td>unassigned</td>
<td>YES NVRAM</td>
<td>administratively down</td>
<td>down</td>
</tr>
<tr>
<td>GigabitEthernet5/0</td>
<td>unassigned</td>
<td>YES NVRAM</td>
<td>administratively down</td>
<td>down</td>
</tr>
<tr>
<td>Loopback0</td>
<td>101.101.101</td>
<td>YES NVRAM</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td>Loopback1</td>
<td>151.0.0.1</td>
<td>YES NVRAM</td>
<td>up</td>
<td>up</td>
</tr>
</tbody>
</table>

HQ# show ip route

Codes: 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
* - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

102.0.0.0/32 is subnetted, 1 subnets
D  102.102.102.102 [90/31072] via 20.0.1.1, 05:38:32, GigabitEthernet1/0
101.0.0.0/32 is subnetted, 1 subnets
C  101.101.101.101 is directly connected, Loopback0
20.0.0.0/24 is subnetted, 2 subnets
C  20.0.1.0 is directly connected, GigabitEthernet1/0
20.0.0.0/24 is subnetted, 1 subnets
D  172.21.0.0/24 is subnetted, 1 subnets
Implementation of MPLS network

C 172.21.0.0 is directly connected, FastEthernet0/0
   172.22.0.0/24 is subnetted, 1 subnets
D 172.22.0.0 [90/28672] via 20.0.1.1, 05:38:32, GigabitEthernet1/0
D 152.0.0.0/8 [90/131072] via 20.0.1.1, 05:38:32, GigabitEthernet1/0
C 151.0.0.0/8 is directly connected, Loopback1
HQ# ping 102.102.102.102
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 102.102.102.102, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 48/66/88 ms
HQ# traceroute 102.102.102.102
Type escape sequence to abort.
Tracing the route to 102.102.102.102
   1 20.0.1.1 32 msec 20 msec 8 msec
   2 10.0.1.3 [MPLS: Labels 31/32 Exp 0] 68 msec 36 msec 48 msec
   3 10.0.4.5 [MPLS: Labels 28/32 Exp 0] 56 msec 44 msec 48 msec
   4 10.0.5.6 [MPLS: Labels 31/32 Exp 0] 44 msec 52 msec 48 msec
   5 20.0.2.2 [MPLS: Label 32 Exp 0] 56 msec 44 msec 40 msec
   6 20.0.2.102 40 msec 48 msec 52 msec
HQ#

Branch

Branch# show ip interface brief
Interface            IP-Address      OK? Method Status              Protocol
FastEthernet0/0      172.22.0.1    YES NVRAM  up                    up
FastEthernet0/1      unassigned    YES NVRAM  administratively down down
GigabitEthernet1/0   20.0.2.102    YES NVRAM  up                    up
GigabitEthernet3/0   unassigned    YES NVRAM  administratively down down
GigabitEthernet5/0   unassigned    YES NVRAM  administratively down down
GigabitEthernet6/0   unassigned    YES NVRAM  administratively down down
Loopback0            102.102.102.102 YES NVRAM  up                    up
Branch#

Branch# show ip route
Codes: 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
       * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

102.0.0.0/32 is subnetted, 1 subnets
C 102.102.102.102 is directly connected, Loopback0
101.0.0.0/32 is subnetted, 1 subnets
D 101.101.101.101 [90/131072] via 20.0.2.2, 00:07:33, GigabitEthernet1/0
20.0.0.0/24 is subnetted, 2 subnets
D 20.0.1.0 [90/3072] via 20.0.2.2, 00:07:33, GigabitEthernet1/0
C 20.0.2.0 is directly connected, GigabitEthernet1/0
172.21.0.0/24 is subnetted, 1 subnets
D 172.21.0.0 [90/28672] via 20.0.2.2, 00:07:33, GigabitEthernet1/0
172.22.0.0/24 is subnetted, 1 subnets
C 172.22.0.0 is directly connected, FastEthernet0/0
Branch#

Branch# ping 101.101.101.101
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 101.101.101.101, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 52/61/84 ms
Branch#

Branch# traceroute 101.101.101.101
Type escape sequence to abort.
Tracing the route to 101.101.101.101
Implementation of MPLS network

1 20.0.2.2 24 msec 24 msec 8 msec
2 10.0.8.7 [MPLS: Labels 30/32 Exp 0] 36 msec 64 msec 44 msec
3 20.0.1.1 [MPLS: Label 32 Exp 0] 48 msec 52 msec 40 msec
4 20.0.1.101 40 msec 52 msec 56 msec

Branch#

CE1

CE1#show ip interface brief
Interface          IP-Address      OK? Method Status                Protocol
FastEthernet0/0    unassigned      YES NVRAM  administratively down down
FastEthernet0/1    unassigned      YES NVRAM  administratively down down
GigabitEthernet1/0  20.0.3.103      YES NVRAM up                    up
Loopback0          103.103.103.103 YES NVRAM up                    up
Loopback1          153.0.0.1       YES NVRAM up                    up

CE1#show ip route
Codes: 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
Gateway of last resort is not set

103.0.0.0/32 is subnetted, 1 subnets
C 103.103.103.103 is directly connected, Loopback0
20.0.0.0/24 is subnetted, 2 subnets
B 20.0.4.0 [20/0] via 20.0.3.1, 01:14:08
C 20.0.3.0 is directly connected, GigabitEthernet1/0
104.0.0.0/32 is subnetted, 1 subnets
B 104.104.104.104 [20/0] via 20.0.3.1, 01:14:08
C 153.0.0.0/8 is directly connected, Loopback1
B 154.0.0.0/8 [20/0] via 20.0.3.1, 01:14:08

CE1#ping 104.104.104.104
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 104.104.104.104, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/64/96 ms

CE1#traceroute 104.104.104.104
Type escape sequence to abort.
Tracing the route to 104.104.104.104
1 20.0.3.1 20 msec 24 msec 12 msec
2 10.0.1.1 [MPLS: Labels 31/36 Exp 0] 68 msec 72 msec 68 msec
3 10.0.4.5 [MPLS: Labels 28/36 Exp 0] 80 msec 68 msec 72 msec
4 20.0.4.2 [AS 65100] [MPLS: Label 36 Exp 0] 68 msec 72 msec 36 msec
5 20.0.4.104 [AS 65100] 72 msec 56 msec 44 msec
6 20.0.4.104 [AS 65100] 72 msec 40 msec 56 msec

CE2

CE2#show ip interface brief
Interface          IP-Address      OK? Method Status                Protocol
FastEthernet0/0    unassigned      YES NVRAM  administratively down down
FastEthernet0/1    unassigned      YES NVRAM  administratively down down
GigabitEthernet1/0  20.0.4.104      YES NVRAM up                    up
Loopback0          104.104.104.104 YES NVRAM up                    up
Loopback1          154.0.0.1       YES NVRAM up                    up

CE2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
Implementation of MPLS network

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

Gateway of last resort is not set

103.0.0.0/32 is subnetted, 1 subnets
 B 103.103.103.103 [20/0] via 20.0.4.2, 01:15:56
20.0.0.0/24 is subnetted, 2 subnets
 C 20.0.4.0 is directly connected, GigabitEthernet1/0
 B 20.0.3.0 [20/0] via 20.0.4.2, 01:15:56
104.0.0.0/32 is subnetted, 1 subnets
 C 104.104.104.104 is directly connected, Loopback0
 B 153.0.0.0/8 [20/0] via 20.0.4.2, 01:15:56
 C 154.0.0.0/8 is directly connected, Loopback1

CE2# ping 103.103.103.103
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 103.103.103.103, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 44/56/72 ms

CE2# traceroute 103.103.103.103
Type escape sequence to abort.
Tracing the route to 103.103.103.103
1 20.0.4.2 20 msec 24 msec 16 msec
2 10.0.8.7 [MPLS: Labels 20/36 Exp 0] 80 msec 44 msec 56 msec
3 20.0.3.1 [AS 65100] [MPLS: Label 36 Exp 0] 56 msec 56 msec 52 msec
4 153.0.0.0/8 [AS 65100] 44 msec 76 msec 60 msec

Site1 (Site2 and Site3)

Site1# show ip interface brief
Interface          IP-Address      OK? Method Status                Protocol
FastEthernet0/0    unassigned      YES NVRAM  administratively down down
GigabitEthernet1/0  unassigned      YES NVRAM  up                    up
GigabitEthernet1/0.12 3.0.0.1      YES NVRAM  up                    up
GigabitEthernet1/0.13 4.0.0.1      YES NVRAM  up                    up
GigabitEthernet2/0  unassigned      YES NVRAM  administratively down down
GigabitEthernet3/0  unassigned      YES NVRAM  administratively down down
GigabitEthernet4/0  unassigned      YES NVRAM  administratively down down
GigabitEthernet6/0  unassigned      YES NVRAM  administratively down down
Site1# ping 3.0.0.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 3.0.0.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 48/60/92 ms

Site1# ping 4.0.0.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.0.0.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 44/56/76 ms

Site1# traceroute 3.0.0.2
Type escape sequence to abort.
Tracing the route to 3.0.0.2
1 3.0.0.2 80 msec 56 msec 96 msec
Site1# traceroute 4.0.0.2
Type escape sequence to abort.
Implementation of MPLS network

Tracing the route to 4.0.0.2

1 4.0.0.2 68 msec 72 msec 56 msec

Site1#

PEI (head end & tail end)

PEI# show ip interface brief

<table>
<thead>
<tr>
<th>Interface</th>
<th>IP-Address</th>
<th>OK? Method</th>
<th>Status</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>FastEthernet0/0</td>
<td>unassigned</td>
<td>YES NVRAM</td>
<td>administratively down</td>
<td>down</td>
</tr>
<tr>
<td>FastEthernet0/1</td>
<td>unassigned</td>
<td>YES NVRAM</td>
<td>administratively down</td>
<td>down</td>
</tr>
<tr>
<td>GigabitEthernet2/0</td>
<td>10.0.1.1</td>
<td>YES NVRAM</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td>GigabitEthernet3/0</td>
<td>10.0.10.1</td>
<td>YES NVRAM</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td>GigabitEthernet4/0.12</td>
<td>unassigned</td>
<td>YES unset</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td>GigabitEthernet4/0.13</td>
<td>unassigned</td>
<td>YES unset</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td>GigabitEthernet5/0</td>
<td>20.0.3.1</td>
<td>YES NVRAM</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td>GigabitEthernet6/0</td>
<td>20.0.1.1</td>
<td>YES NVRAM</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td>Loopback0</td>
<td>1.1.1.1</td>
<td>YES NVRAM</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td>Loopback1</td>
<td>50.0.0.1</td>
<td>YES NVRAM</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td>Tunnel1</td>
<td>1.1.1.1</td>
<td>YES TFTP</td>
<td>up</td>
<td>up</td>
</tr>
</tbody>
</table>

PEI#

PEI# show mpls interfaces

<table>
<thead>
<tr>
<th>Interface</th>
<th>IP</th>
<th>Tunnel</th>
<th>BGP Static Operational</th>
</tr>
</thead>
<tbody>
<tr>
<td>GigabitEthernet1/0</td>
<td>Yes (ldp)</td>
<td>Yes</td>
<td>No No Yes</td>
</tr>
<tr>
<td>GigabitEthernet2/0</td>
<td>Yes (ldp)</td>
<td>Yes</td>
<td>No No Yes</td>
</tr>
<tr>
<td>GigabitEthernet3/0</td>
<td>Yes (ldp)</td>
<td>Yes</td>
<td>No No Yes</td>
</tr>
<tr>
<td>Tunnel1</td>
<td>No</td>
<td>No</td>
<td>No No Yes</td>
</tr>
</tbody>
</table>

PEI#

PEI# show ip route

Gateway of last resort is not set

1.0.0.0/32 is subnetted, 1 subnets
C 1.1.1.1 is directly connected, Loopback0
C 50.0.0.0/8 is directly connected, Loopback1
2.0.0.0/32 is subnetted, 1 subnets
O 2.2.2.2 [10/251] via 2.2.2.2, 00:02:20, Tunnel1
3.0.0.0/32 is subnetted, 1 subnets
O 3.3.3.3 [10/51] via 10.0.1.3, 00:02:20, GigabitEthernet2/0
4.0.0.0/32 is subnetted, 1 subnets
O 4.4.4.4 [10/151] via 10.0.1.3, 00:02:20, GigabitEthernet2/0
5.0.0.0/32 is subnetted, 1 subnets
O 5.5.5.5 [10/101] via 10.0.10.5, 00:02:20, GigabitEthernet3/0
6.0.0.0/32 is subnetted, 1 subnets
O 6.6.6.6 [10/201] via 10.0.10.5, 00:02:20, GigabitEthernet3/0
7.0.0.0/32 is subnetted, 1 subnets
O 7.7.7.7 [10/101] via 10.0.7.7, 00:02:20, GigabitEthernet1/0
10.0.0.0/24 is subnetted, 10 subnets
C 10.0.0.0 is directly connected, GigabitEthernet3/0
O 10.0.8.0 [10/2100] via 10.0.7.7, 00:02:20, GigabitEthernet1/0
O 10.0.9.0 [10/350] via 2.2.2.2, 00:02:20, Tunnel1
O 10.0.2.0 [10/150] via 10.0.1.3, 00:02:20, GigabitEthernet2/0
O 10.0.3.0 [10/250] via 10.0.1.3, 00:02:20, GigabitEthernet2/0
C 10.0.1.0 is directly connected, GigabitEthernet2/0
O 10.0.6.0 [10/250] via 10.0.1.3, 00:02:20, GigabitEthernet2/0
C 10.0.7.0 is directly connected, GigabitEthernet1/0
O 10.0.4.0 [10/150] via 10.0.1.3, 00:02:20, GigabitEthernet2/0
O 10.0.5.0 [10/200] via 10.0.10.5, 00:02:20, GigabitEthernet3/0

PEI#

PEI# traceroute 2.2.2.2

Type escape sequence to abort.
Tracing the route to 2.2.2.2

1 10.0.1.3 [MPLS: Label 31 Exp 0] 24 msec 72 msec 16 msec

101
Implementation of MPLS network

```
2 10.0.4.5 [MPLS: Label 28 Exp 0] 28 msec 32 msec 24 msec
3 10.0.5.6 [MPLS: Label 31 Exp 0] 28 msec 32 msec 24 msec
4 10.0.9.2 36 msec 32 msec 36 msec
```

PE1# show ip route vrf CUSTOMER1

Routing Table: CUSTOMER1
Codes: 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

Gateway of last resort is not set

```
102.0.0.0/32 is subnetted, 1 subnets
B  102.102.102.102 [200/130816] via 2.2.2.2, 05:40:41
101.0.0.0/32 is subnetted, 1 subnets
20.0.0.0/24 is subnetted, 2 subnets
B  20.0.1.0 is directly connected, GigabitEthernet6/0
B  20.0.2.0 [200/0] via 2.2.2.2, 05:40:41
172.21.0.0/24 is subnetted, 1 subnets
D  172.21.0.0 [90/28416] via 20.0.1.101, 05:42:41, GigabitEthernet6/0
172.22.0.0/24 is subnetted, 1 subnets
B  172.22.0.0 [200/28416] via 2.2.2.2, 05:40:41
B  152.0.0.0/8 [200/130816] via 2.2.2.2, 05:40:41
D  151.0.0.0/8 [90/130816] via 20.0.1.101, 05:42:41, GigabitEthernet6/0
```

PE1# show mpls forwarding-table

```
Local   Outgoing   Prefix     Bytes Label   Outgoing   Next Hop
Label   Label or VC Switched Interface
16      No Label   12ckt(13)  248356 G14/0.13 point2point
17      No Label   12ckt(12)  2386 G14/0.12 point2point
18      Pop Label  7.7.7.7/32 0 G11/0 10.0.7.7
19      Pop Label  10.0.8.0/24 0 G11/0 10.0.7.7
20      16         6.6.6.6/32 0 G13/0 10.0.10.5
21      Pop Label  5.5.5.5/32 0 G13/0 10.0.10.5
22      16         4.4.4.4/32 0 G12/0 10.0.1.3
23      Pop Label  3.3.3.3/32 0 G12/0 10.0.1.3
24      Pop Label  [T] 2.2.2.2/32 54424 Tu1 point2point
25      No Label   [T] 10.0.9.0/24 0 Tu1 point2point
26      17         10.0.6.0/24 0 G12/0 10.0.1.3
27      18         10.0.3.0/24 0 G12/0 10.0.1.3
28      Pop Label  10.0.5.0/24 0 G13/0 10.0.10.5
29      Pop Label  10.0.4.0/24 0 G12/0 10.0.1.3
30      Pop Label  10.0.2.0/24 0 G12/0 10.0.1.3
31      No Label   20.0.1.0/24[V] 14628 aggregate/CUSTOMER1
32      No Label   101.101.101.101/32[V] 0 G16/0 20.0.1.101
33      No Label   151.0.0.0/8[V] 3306 G16/0 20.0.1.101
```

102
### Implementation of MPLS network

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>34</td>
<td>No Label</td>
<td>172.21.0.0/24 [V]</td>
<td>0</td>
<td>G16/0</td>
</tr>
<tr>
<td>35</td>
<td>No Label</td>
<td>153.0.0.0/8 [V]</td>
<td>684</td>
<td>G15/0</td>
</tr>
<tr>
<td>36</td>
<td>No Label</td>
<td>103.103.103.103/32 [V]</td>
<td>\</td>
<td>G15/0</td>
</tr>
<tr>
<td>37</td>
<td>No Label</td>
<td>20.0.3.0/24 [V]</td>
<td>0</td>
<td>aggregate/CUSTOMER2</td>
</tr>
</tbody>
</table>

[T] Forwarding through a LSP tunnel.

View additional labelling info with the 'detail' option

PE1#

PE1# show mpls traffic-eng tunnels

Name: PE1_t1  (Tunnel1)  Destination: 2.2.2.2

Status:

- Admin: up
- Oper: up
- Path: valid
- Signalling: connected

path option 10, type explicit TUNNEL1 PRI (Basis for Setup, path weight 1250)
- path option 15, type explicit TUNNEL1_BCK
- path option 20, type dynamic

Config Parameters:

- Bandwidth: 0 kbps (Global)
- Priority: 7
- Affinity: 0x0/0xFFFF
- AutoRoute: enabled
- LockDown: disabled
- Loadshare: 0
- bw-based auto-bw: disabled

Active Path Option Parameters:

- State: explicit path option 10 is active
- InLabel Override: disabled
- OutLabel: GigabitEthernet2/0, 31

RSVP Signalling Info:

- Src 1.1.1.1, Dat 2.2.2.2, Tun_Id 1, Tun_Instance 74
- RSVP Path Info:
  - My Address: 10.0.1.1
  - Explicit Route: 10.0.1.3 10.0.4.3 10.0.4.5 10.0.5.5 10.0.5.6 10.0.9.6 10.0.9.2 2.2.2.2
- Record Route: TaSpec: ave rate=0 kbits, burst=1000 bytes, peak rate=0 kbits

RSVP Resv Info:

- Record Route: TaSpec: ave rate=0 kbits, burst=1000 bytes, peak rate=0 kbits
- Source: 1.1.1.1
- Destination: 2.2.2.2
- Tun_Id: 0
- Tun_Instance: 17
- Explicit Route: None
- Record Route: None
- FSpec: ave rate=0 kbits, burst=1000 bytes, peak rate=0 kbits

History:

- Tunnel:
  - Time since created: 5 hours, 42 minutes
  - Time since path change: 2 minutes, 37 seconds
  - Number of LSP IDs (Tun_Instances) used: 74
- Current LSP:
  - Uptime: 2 minutes, 40 seconds
  - Selection: reoptimization
- Prior LSP:
  - ID: path option 15 [73]
  - Removal Trigger: reoptimization completed

LSP Tunnel PE2_t0 is signalled, connection is up

InLabel : -
OutLabel : GigabitEthernet1/0, implicit-null

RSVP Signalling Info:

- Src 2.2.2.2, Dat 1.1.1.1, Tun_Id 0, Tun_Instance 17
- RSVP Path Info:
  - My Address: 1.1.1.1
  - Explicit Route: None
- Record Route: None
- TaSpec: ave rate=0 kbits, burst=1000 bytes, peak rate=0 kbits
- RSVP Resv Info:
  - Record Route: None
  - FSpec: ave rate=0 kbits, burst=1000 bytes, peak rate=0 kbits

PE1#

PE1# show mpls l2transport vc

<table>
<thead>
<tr>
<th>Local intf</th>
<th>Local circuit</th>
<th>Dest address</th>
<th>VC ID</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>G14/0.12</td>
<td>Eth VLAN 12</td>
<td></td>
<td>2.2.2.2</td>
<td>12</td>
</tr>
<tr>
<td>G14/0.13</td>
<td>Eth VLAN 13</td>
<td></td>
<td>2.2.2.2</td>
<td>13</td>
</tr>
</tbody>
</table>

103
Implementation of MPLS network

PE1#show ip bgp summary
BGP router identifier 50.0.0.1, local AS number 36500
BGP table version is 2, main routing table version 2
1 network entries using 132 bytes of memory
1 path entries using 52 bytes of memory
11/1 BGP path/bestpath attribute entries using 1848 bytes of memory
4 BGP extended community entries using 204 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
Bitfield cache entries: current 2 (at peak 3) using 64 bytes of memory
BGP using 2324 total bytes of memory
BGP activity 64/42 prefixes, 97/75 paths, scan interval 60 secs

Neighbor V AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State/PfxRcd
2.2.2.2 4 36500 362 354 2 0 0 05:41:13 0
PE1#

PE1#show ip bgp vpnv4 all
BGP table version is 182, local router ID is 50.0.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

Network Next Hop Metric LocPrf Weight Path
 Route Distinguisher: 1.1.1.1:100 (default for vrf CUSTOMER1)
* 20.0.1.0/24 0.0.0.0 0 32768 7
**120.0.2.0/24 2.2.2.2 0 100 0 7
* 101.101.101.101/32 20.0.1.101 130816 32768 7
**1102.102.102.102/32 2.2.2.2 130816 100 0 7
* 151.0.0.0/8 20.0.1.101 130816 32768 7
**1152.0.0.0/8 2.2.2.2 130816 100 0 7
* 172.21.0.0/24 20.0.1.101 28416 32768 7
**1172.22.0.0/24 2.2.2.2 28416 100 0 7
Route Distinguisher: 1.1.1.1:200 (default for vrf CUSTOMER2)
 r 20.0.3.0/24 20.0.3.103 0 0 65100 i
**120.0.4.0/24 2.2.2.2 0 100 0 65100 i
* 103.103.103.103/32 20.0.3.103 0 0 65100 i
**1104.104.104.104/32 2.2.2.2 0 100 0 65100 i
* 153.0.0.0/8 20.0.3.103 0 0 65100 i
**1154.0.0.0/8 2.2.2.2 0 100 0 65100 i
Route Distinguisher: 2.2.2.2:100
**120.0.2.0/24 2.2.2.2 0 100 0 7
**1102.102.102.102/32 2.2.2.2 130816 100 0 7
**1152.0.0.0/8 2.2.2.2 130816 100 0 7
**1172.22.0.0/24 2.2.2.2 28416 100 0 7
Route Distinguisher: 2.2.2.2:200
**120.0.4.0/24 2.2.2.2 0 100 0 65100 i
**1104.104.104.104/32 2.2.2.2 0 100 0 65100 i
**1154.0.0.0/8 2.2.2.2 0 100 0 65100 i
PE1#

PE2 (head end & tail end)

PE2#show interface brief
Interface      IP-Address      OK? Method Status            Protocol
FastEthernet0/0 unassigned YES NVRAM up           up
FastEthernet0/1 unassigned YES NVRAM administratively down down
GigabitEthernet1/0 10.0.8.2 YES NVRAM up           up
GigabitEthernet2/0 10.0.3.2 YES NVRAM up           up
GigabitEthernet3/0 10.0.9.2 YES NVRAM up           up
GigabitEthernet4/0 unassigned YES NVRAM up           up
GigabitEthernet4/0.12 unassigned YES unset        up
GigabitEthernet5/0 20.0.4.2 YES NVRAM up           up
GigabitEthernet6/0 20.0.2.2 YES NVRAM up           up
Loopback0          1.1.1.1 YES NVRAM up           up
Tunnel0           2.2.2.2 YES TFTP up              up
PE2#

PE2#show mpls interfaces

104
Implementation of MPLS network

<table>
<thead>
<tr>
<th>Interface</th>
<th>IP</th>
<th>Tunnel</th>
<th>BGP Static</th>
<th>Operational</th>
</tr>
</thead>
<tbody>
<tr>
<td>GigabitEthernet1/0</td>
<td>Yes (ldp)</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>GigabitEthernet2/0</td>
<td>Yes (ldp)</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>GigabitEthernet3/0</td>
<td>Yes (ldp)</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Tunnel0</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

PE2# show ip route

Codes: 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, F - periodic downloaded static route

Gateway of last resort is not set

1.0.0.0/32 is subnetted, 1 subnets
 O 1.1.1.1 [110/301] via 1.1.1.1, 05:42:45, Tunnel0
B 50.0.0.0/8 [200/0] via 1.1.1.1, 05:42:39
2.0.0.0/32 is subnetted, 1 subnets
 C 2.2.2.2 is directly connected, Loopback0
O 3.0.0.0/32 is subnetted, 1 subnets
 O 3.3.3.3 [110/201] via 10.0.3.4, 05:42:45, GigabitEthernet2/0
4.0.0.0/32 is subnetted, 1 subnets
 O 4.4.4.4 [110/101] via 10.0.3.4, 05:42:45, GigabitEthernet2/0
5.0.0.0/32 is subnetted, 1 subnets
 O 5.5.5.5 [110/201] via 10.0.9.6, 05:32:22, GigabitEthernet3/0
6.0.0.0/32 is subnetted, 1 subnets
 O 6.6.6.6 [110/101] via 10.0.9.6, 05:42:45, GigabitEthernet3/0
7.0.0.0/32 is subnetted, 1 subnets
 O 7.7.7.7 [110/401] via 1.1.1.1, 05:42:45, Tunnel0
10.0.0.0/24 is subnetted, 10 subnets
 O 10.0.10.0 [110/300] via 10.0.9.6, 05:32:22, GigabitEthernet3/0
10.0.8.0 is directly connected, GigabitEthernet1/0
C 10.0.9.0 is directly connected, GigabitEthernet2/0
O 10.0.1.0 [110/201] via 10.0.3.4, 05:42:45, GigabitEthernet2/0
C 10.0.2.0 is directly connected, GigabitEthernet2/0
O 10.0.3.0 [110/201] via 10.0.3.4, 05:42:45, GigabitEthernet2/0
O 10.0.6.0 [110/201] via 10.0.3.4, 05:42:45, GigabitEthernet2/0
O 10.0.7.0 [110/300] via 1.1.1.1, 05:42:45, Tunnel0
O 10.0.8.0 [110/300] via 10.0.9.6, 05:32:23, GigabitEthernet3/0
10.0.9.0 is directly connected, GigabitEthernet2/0
10.0.5.0 [110/201] via 10.0.9.6, 05:42:45, GigabitEthernet3/0

PE2# traceroute 1.1.1.1

Type escape sequence to abort.
Tracing the route to 1.1.1.1
1 10.0.8.7 [MPLS: Label 20 Exp 0] 48 msec 16 msec 44 msec
2 10.0.7.1 60 msec 44 msec 12 msec

PE2# show ip route vrf CUSTOMER1

Routing Table: CUSTOMER1

Codes: 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, F - periodic downloaded static route

Gateway of last resort is not set

102.0.0.0/32 is subnetted, 1 subnets
 O 102.102.102.102 [90/130816] via 20.0.2.102, 05:43:39, GigabitEthernet6/0
101.0.0.0/32 is subnetted, 1 subnets
20.0.0.0/24 is subnetted, 2 subnets
 B 20.0.1.1 [200/0] via 1.1.1.1, 05:42:18
C 20.0.2.0 is directly connected, GigabitEthernet6/0
172.21.0.0/24 is subnetted, 1 subnets
 B 172.21.0.0 [200/20416] via 1.1.1.1, 05:42:18

105
Implementation of MPLS network

172.22.0.0/24 is subnetted, 1 subnets
D   172.22.0.0 [90/28416] via 20.0.2.102, 05:43:39, GigabitEthernet6/0
D   152.0.0.0/8 [90/130816] via 20.0.2.102, 05:43:39, GigabitEthernet6/0
B   151.0.0.0/8 [200/130816] via 1.1.1.1, 05:42:18

PE2# show ip route vrf CUSTOMER2

Routing Table: CUSTOMER2
Codes: 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

Gateway of last resort is not set

103.0.0.0/32 is subnetted, 1 subnets
B   103.103.103.103 [200/0] via 1.1.1.1, 02:02:42
20.0.0.0/24 is subnetted, 2 subnets
C   20.0.4.0 is directly connected, GigabitEthernet5/0
B   20.0.3.0 [200/0] via 1.1.1.1, 02:02:42
104.0.0.0/32 is subnetted, 1 subnets
B   104.104.104.104 [20/0] via 20.0.4.104, 01:09:03
B   153.0.0.0/8 [200/0] via 1.1.1.1, 02:02:42
B   154.0.0.0/8 [20/0] via 20.0.4.104, 01:09:03

PE2# show mpls forwarding-table

Local Outgoing Prefix Bytes Label Outgoing Next Hop
Label Label or VC Switched or Tunnel Id Switched interface
Label or Tunnel Id
16   No Label 12ckt(13) 2316 Fa0/0 point2point
17   No Label 12ckt(12) 2416 Gi4/0.12 point2point
18   No Label [T] 7.7.7.7/32 0 Tu0 point2point
19   Pop label 6.6.6.6/32 0 G13/0 10.0.9.6
20   16  5.5.5.5/32 0 G13/0 10.0.9.6
21   Pop label 4.4.4.4/32 0 G12/0 10.0.3.4
22   16  3.3.3.3/32 0 G12/0 10.0.3.4
23   Pop label [T] 1.1.1.1/32 0 Tu0 point2point
24   Pop label 10.0.6.0/24 0 G12/0 10.0.3.4
25   Pop label 10.0.5.0/24 0 G13/0 10.0.9.6
26   17  10.0.4.0/24 0 G13/0 10.0.9.6
27   Pop label 10.0.2.0/24 0 G13/0 10.0.9.6
28   23  10.0.10.0/24 0 G13/0 10.0.9.6
29   18  10.0.1.0/24 0 G12/0 10.0.3.4
30   No Label [T] 10.0.7.0/24 0 Tu0 point2point
31   No Label 20.0.2.0/24 [V] 5826 aggregate/CUSTOMER1
32   No Label 102.102.102.102/32 [V] \ 5994 G16/0 20.0.2.102
33   No Label 152.0.0.0/8 [V] 4854 G16/0 20.0.2.102
34   No Label 172.22.0.0/24 [V] 0 G16/0 20.0.2.102
35   No Label 20.0.4.0/24 [V] 1314 aggregate/CUSTOMER2
36   No Label 104.104.104.104/32 [V] \ 0 G15/0 20.0.4.104
37   No Label 154.0.0.0/8 [V] 0 G15/0 20.0.4.104

[T] Forwarding through a LSP tunnel.
View additional labelling info with the 'detail' option

PE2# show mpls traffic-eng tunnels

Name: PE2_t0
   (Tunnel0) Destination: 1.1.1.1
Status:
   Admin: up   Oper: up   Path: Valid   Signalling: Connected
   path option 3, type explicit EXCEPT-P4-P5 (Basis for Setup, path weight 2100)
   path option 5, type dynamic
Config Parameters:
   Bandwidth: 0  kbps (Global) Priority: 7 7 Affinity: 0x0/0xFFFF
   Metric Type: TE (default)
   AutoRoute: enabled  LockDown: disabled  Loadshare: 0  bw-based
   auto-bw: disabled
Active Path Option Parameters:
   State: explicit path option 3 is active
   BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled
InLabel: -
OutLabel: GigabitEthernet1/0, 20
RSVP Signalling Info:
Src 2.2.2.2, Dst 1.1.1.1, Tun_Id 0, Tun_Instance 17
RSVP Path Info:
My Address: 10.0.8.2
Explicit Route: 10.0.8.7 10.0.7.7 10.0.7.1 1.1.1.1
Record Route: NONE
Tspec: ave rate=0 kbits, burst=1000 bytes, peak rate=0 kbits
RSVP Resv Info:
Record Route: NONE
Fspec: ave rate=0 kbits, burst=1000 bytes, peak rate=0 kbits
History:
Tunnel:
Time since created: 5 hours, 43 minutes
Time since path change: 5 hours, 43 minutes
Number of LSP IDs (Tun_Instances) used: 17
Current LSP:
Uptime: 5 hours, 43 minutes
LSP Tunnel PE1_t1 is signalled, connection is up
InLabel: GigabitEthernet3/0, implicit-null
OutLabel: -
RSVP Signalling Info:
Src 1.1.1.1, Dst 2.2.2.2, Tun_Id 1, Tun_Instance 74
RSVP Path Info:
My Address: 2.2.2.2
Explicit Route: NONE
Record Route: 10.0.9.6 10.0.5.5 10.0.4.3 10.0.1.1
Tspec: ave rate=0 kbits, burst=1000 bytes, peak rate=0 kbits
RSVP Resv Info:
Record Route: NONE
Fspec: ave rate=0 kbits, burst=1000 bytes, peak rate=0 kbits

PE2#
PE2#show mpls l2transport vc
Local intf  Local circuit     Dest address  VC ID  Status
----------  -------------------  ------------- ----  ----
Gi4/0.12    Eth VLAN 12       1.1.1.1       12    UP
Fa0/0       Ethernet           1.1.1.1       13    UP

PE2#
PE2#show ip bgp summary
BGP router identifier 2.2.2.2, local AS number 36500
BGP table version is 2, main routing table version 2
1 network entries using 132 bytes of memory
1 path entries using 52 bytes of memory
11/1 BGP path/bestpath attribute entries using 148 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
4 BGP extended community entries using 204 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP hash cache entries: current 2 (at peak 3) using 1848 bytes of memory
BGP using 2324 total bytes of memory
BGP activity 46/24 prefixes, 82/60 paths, scan interval 60 secs

PE2#
PE2#show ip bgp vpnv4 all
BGP table version is 157, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

Network      Next Hop     Metric LocPrf Weight Path
-----------  -----------  ------ ------- -----  -------
*120.0.1.0/24 1.1.1.1     0     100      0   0
*120.101.0.101.0/32 1.1.1.1 130816   100  0   0
*151.0.0.0/8 1.1.1.1 130816   100  0   0
*172.21.0.0/24 1.1.1.1 28416   100  0   0
Route Distinguisher: 1.1.1.1:200
*120.0.3.0/24 1.1.1.1 0     100      0   65100 1
*1103.103.103.103/32 1.1.1.1 0     100      0   65100 1

107
Implementation of MPLS network

*1153.0.0.0/8    1.1.1.1    0   100    0   65100 i
Route Distinguisher: 2.2.2.2:100 (default for vrf CUSTOMER1)
*120.0.1.0/24    1.1.1.1    0   100    0   7
*20.0.2.0/24     0.0.0.0     0   32768   ?
*1101.101.101.101/32
  1.1.1.1     130816    100    0   7
  102.102.102.102/32
  1.1.1.1     20.0.2.102 130816    32768   ?
*151.0.0.0/8     1.1.1.1    130816    100    0   7
  152.0.0.0/8     20.0.2.102 130816    32768   ?
*172.21.0.0/24   1.1.1.1    28416    100    0   7
  172.22.0.0/24   20.0.2.102 28416    32768   ?
Route Distinguisher: 2.2.2.2:200 (default for vrf CUSTOMER2)
*120.0.3.0/24    1.1.1.1    0   100    0   65100 i
> 20.0.4.0/24     20.0.4.104 0   0   65100 i
*1103.103.103.103/32
  1.1.1.1    104.104.104.104/32
  20.0.4.104 0   0   65100 i
*153.0.0.0/8     1.1.1.1    0   100    0   65100 i
  154.0.0.0/8     20.0.4.104 0   0   65100 i
PE2#

P3 (PLR for node P5)
P3#show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            unassigned      YES NVRAM  administratively down down
FastEthernet0/1            unassigned      YES NVRAM  administratively down down
GigabitEthernet1/0         10.0.1.3        YES NVRAM  up                    up
GigabitEthernet2/0         10.0.2.3        YES NVRAM  up                    up
GigabitEthernet3/0         10.0.4.3        YES NVRAM  up                    up
GigabitEthernet4/0         unassigned      YES NVRAM  up                    up
GigabitEthernet5/0         unassigned      YES NVRAM  administratively down down
GigabitEthernet6/0         unassigned      YES NVRAM  administratively down down
Loopback0                  3.3.3.3         YES NVRAM  up                    up
Tunnel52                   3.3.3.3         YES TFTP   up                    up
P3#
P3#show mpls interfaces
Interface              IP            Tunnel   BGP Static Operational
GigabitEthernet1/0     Yes (ldp)     Yes      No  No     Yes
GigabitEthernet2/0     Yes (ldp)     Yes      No  No     Yes
GigabitEthernet3/0     Yes (ldp)     Yes      No  No     Yes
Tunnel52               No            No       No  No     Yes
P3#
P3#show ip route
Codes: 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
* - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
1.0.0.0/32 is subnetted, 1 subnets
  O 1.1.1.1 [110/101] via 10.0.1.1, 05:56:56, GigabitEthernet1/0
  O 2.0.0.0/32 is subnetted, 1 subnets
  O 2.0.0.0/32 is subnetted, 1 subnets
  O 2.0.0.0/32 is subnetted, 1 subnets
  O 3.3.3.3 is directly connected, Loopback0
  O 4.0.0.0/32 is subnetted, 1 subnets
  O 4.4.4.4 [110/101] via 10.0.2.4, 05:56:46, GigabitEthernet2/0
  O 4.0.0.0/32 is subnetted, 1 subnets
  O 5.5.5.5 [110/101] via 10.0.4.5, 05:57:06, GigabitEthernet3/0
  O 5.0.0.0/32 is subnetted, 1 subnets
  O 6.0.0.0/32 is subnetted, 1 subnets
  O 6.6.6.6 [110/101] via 10.0.4.5, 05:46:22, GigabitEthernet3/0
  O 7.0.0.0/32 is subnetted, 1 subnets
  O 1.0.0.0/24 is subnetted, 10 subnets
  O 1.0.0.0/24 is subnetted, 10 subnets
  O 10.0.10.0 [110/200] via 10.0.4.5, 05:57:06, GigabitEthernet3/0
  O 110/200 via 10.0.1.1, 05:56:56, GigabitEthernet1/0
  O 10.0.0.0/24 is subnetted, 10 subnets
  O 10.0.0.0/24 is subnetted, 10 subnets
Implementation of MPLS network

P3#

P3# show mpls forwarding-table

Local  Outgoing      Prefix            Bytes Label   Outgoing   Next Hop
Label  Label or VC   or Tunnel Id      Switched      interface
16     Pop Label     4.4.4.4/32        712           Gi2/0      10.0.2.4
17     Pop Label     10.0.6.0/24       0             Gi2/0      10.0.2.4
18     Pop Label     10.0.3.0/24       0             Gi2/0      10.0.2.4
19     Pop Label     5.5.5.5/51 [20]  0             Gi2/0      10.0.2.4
20     Pop Label     6.6.6.6/32        1038          Gi2/0      10.0.2.4
21     Pop Label     6.6.6.6/32        0             Gi3/0      10.0.4.5
22     Pop Label     10.0.9.0/24       0             Gi2/0      10.0.2.4
23     Pop Label     10.0.5.0/24       0             Gi3/0      10.0.4.5
24     Pop Label     10.0.10.0/24      0             Gi1/0      10.0.1.1
25     Pop Label     5.5.5.5/32        0             Gi3/0      10.0.4.5
26     Pop Label     2.2.2.2/32        186           Gi2/0      10.0.2.4
27     Pop Label     1.1.1.1/32        0             Gi1/0      10.0.1.1
28     Pop Label     10.0.8.0/24       0             Gi1/0      10.0.1.1
29     Pop Label     10.0.7.0/24       0             Gi1/0      10.0.1.1
31     Pop Label     1.1.1.1 1 [74]    26850         Gi3/0      10.0.4.5

P3#

P3# show mpls traffic-eng tunnels summary

Signalling Summary:
  LSP Tunnels Process:            running
  Passive LSP Listener:          running
  RSVP Process:                   running
  Forwarding:                     enabled
  Head: 1 interfaces, 1 active signalling attempts, 1 established
  1 activations, 0 deactivations
    0 SSO recovery attempts, 0 SSO recovered
  Midpoints: 2, Tails: 0
  Periodic reoptimization:        every 30 seconds, next in 20 seconds
  Periodic FRR Promotion:         Not Running
  Periodic auto-bw collection:    every 300 seconds, next in 113 seconds

P3#

P3# show mpls traffic-eng fast-reroute database

Headend frr information:

Protected tunnel              In-label Out intf/label   FRR intf/label   Status
LSP midpoint frr information:

LSP identifier                In-label Out intf/label   FRR intf/label   Status
1.1.1.1 1 [74]                31       Gi3/0:28         Tu52:31          ready

P3#

P5 (PLR for link P5-P6)

P5# show ip interface brief

Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            unassigned      YES NVRAM  administratively down down
FastEthernet0/1            unassigned      YES NVRAM  administratively down down
GigabitEthernet1/0         10.0.5.5        YES NVRAM up                    up
GigabitEthernet2/0         10.0.4.5        YES NVRAM up                    up
GigabitEthernet3/0         10.0.10.5       YES NVRAM up                    up
GigabitEthernet4/0         unassigned      YES NVRAM administratively down down
GigabitEthernet5/0         unassigned      YES NVRAM administratively down down
Serial1/6                  unassigned      YES NVRAM administratively down down
Serial1/2                  unassigned      YES NVRAM administratively down down
Serial1/3                  unassigned      YES NVRAM administratively down down
Loopback0                  5.5.5.5        YES NVRAM up                    up
Tunnel151                  5.5.5.5        YES TFTP up                    up

P5#
Implementation of MPLS network

P5#show mpls interfaces
Interface            IP            Tunnel   BGP Static Operational
GigabitEthernet1/0   Yes (ldp)     Yes      No  No     Yes
GigabitEthernet2/0   Yes (ldp)     Yes      No  No     Yes
GigabitEthernet3/0   Yes (ldp)     Yes      No  No     Yes
Tunnel51             No            No       No  No     Yes
P5#

P5#show ip route
Codes: 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
Gateway of last resort is not set

1.0.0.0/32 is subnetted, 1 subnets
O       1.1.1.1 [110/101] via 10.0.10.1, 05:49:37, GigabitEthernet3/0

2.0.0.0/32 is subnetted, 1 subnets
O       2.2.2.2 [110/301] via 10.0.5.6, 05:48:10, GigabitEthernet1/0

3.0.0.0/32 is subnetted, 1 subnets
O       3.3.3.3 [110/101] via 10.0.4.3, 05:49:37, GigabitEthernet2/0

4.0.0.0/32 is subnetted, 1 subnets
O       4.4.4.4 [110/201] via 10.0.5.6, 05:48:10, GigabitEthernet1/0

5.0.0.0/32 is subnetted, 1 subnets
O       5.5.5.5 is directly connected, Loopback0

6.0.0.0/32 is subnetted, 1 subnets
O       6.6.6.6 [110/101] via 10.0.5.6, 05:48:10, GigabitEthernet1/0

7.0.0.0/32 is subnetted, 1 subnets
O       7.7.7.7 [110/201] via 10.0.10.1, 05:49:37, GigabitEthernet3/0

10.0.0.0/24 is subnetted, 10 subnets
C       10.0.10.0 is directly connected, GigabitEthernet3/0
O       10.0.8.0 [110/220] via 10.0.10.1, 05:49:37, GigabitEthernet3/0
O       10.0.9.0 [110/220] via 10.0.10.1, 05:49:37, GigabitEthernet3/0
O       10.0.10.0 [110/220] via 10.0.10.1, 05:49:37, GigabitEthernet3/0
O       10.0.11.0 [110/220] via 10.0.10.1, 05:49:37, GigabitEthernet3/0
O       10.0.12.0 [110/220] via 10.0.10.1, 05:49:37, GigabitEthernet3/0
O       10.0.13.0 [110/220] via 10.0.10.1, 05:49:37, GigabitEthernet3/0
O       10.0.14.0 [110/220] via 10.0.10.1, 05:49:37, GigabitEthernet3/0
O       10.0.15.0 [110/220] via 10.0.10.1, 05:49:37, GigabitEthernet3/0
O       10.0.16.0 [110/220] via 10.0.10.1, 05:49:37, GigabitEthernet3/0
O       10.0.17.0 [110/220] via 10.0.10.1, 05:49:37, GigabitEthernet3/0
O       10.0.18.0 [110/220] via 10.0.10.1, 05:49:37, GigabitEthernet3/0
O       10.0.19.0 [110/220] via 10.0.10.1, 05:49:37, GigabitEthernet3/0
O       10.0.20.0 [110/220] via 10.0.10.1, 05:49:37, GigabitEthernet3/0
O       10.0.21.0 [110/220] via 10.0.10.1, 05:49:37, GigabitEthernet3/0
O       10.0.22.0 [110/220] via 10.0.10.1, 05:49:37, GigabitEthernet3/0
O       10.0.23.0 [110/220] via 10.0.10.1, 05:49:37, GigabitEthernet3/0
O       10.0.24.0 [110/220] via 10.0.10.1, 05:49:37, GigabitEthernet3/0
O       10.0.25.0 [110/220] via 10.0.10.1, 05:49:37, GigabitEthernet3/0
O       10.0.26.0 [110/220] via 10.0.10.1, 05:49:37, GigabitEthernet3/0
O       10.0.27.0 [110/220] via 10.0.10.1, 05:49:37, GigabitEthernet3/0
O       10.0.28.0 [110/220] via 10.0.10.1, 05:49:37, GigabitEthernet3/0
O       10.0.29.0 [110/220] via 10.0.10.1, 05:49:37, GigabitEthernet3/0
O       10.0.30.0 [110/220] via 10.0.10.1, 05:49:37, GigabitEthernet3/0
O       10.0.31.0 [110/220] via 10.0.10.1, 05:49:37, GigabitEthernet3/0
O       10.0.32.0 [110/220] via 10.0.10.1, 05:49:37, GigabitEthernet3/0

P5#show mpls forwarding-table
Local Outgoing Prefix            Bytes Label   Outgoing   Next Hop
Label or VC   Prefix or Tunnel Id Switched      interface
16    Pop Label  6.6.6.6/32 0  G1/0/2 10.0.5.6
17    16       4.4.4.4/32 0  G2/0/2 10.0.4.3
17    16       4.4.4.4/32 0  G1/0/2 10.0.5.6
18    Pop Label  3.3.3.3/32 0  G2/0/2 10.0.4.3
19    22       10.0.9.0/24 0  G2/0/2 10.0.4.3
19    22       10.0.9.0/24 0  G1/0/2 10.0.5.6
20    18       10.0.3.0/24 0  G2/0/2 10.0.4.3
22    19       10.0.3.0/24 0  G1/0/2 10.0.5.6
21    18       10.0.1.0/24 0  G3/0/2 10.0.10.1
22    18       10.0.1.0/24 0  G2/0/2 10.0.4.3
23    18       10.0.6.0/24 0  G2/0/2 10.0.4.3
24    18       10.0.6.0/24 0  G1/0/2 10.0.5.6
25    Pop Label  7.7.7.7/32 0  G3/0/2 10.0.10.1
26    25       2.2.2.2/32 0  G1/0/2 10.0.5.6
27    Pop Label  2.2.2.2/32 0  G1/0/2 10.0.5.6
28    27       1.1.1.1 1 30576 G1/0/2 10.0.5.6
29    28       1.1.1.1 1 30576 G1/0/2 10.0.5.6
30    29       1.1.1.1 1 30576 G1/0/2 10.0.5.6

P5#show mpls traffic-eng tunnels summary
Signalling Summary:
LSP Tunnels Process: running
Passive LSP Listener: running
RSVP Process: running
Forwarding: enabled
Head: 1 interfaces, 1 active signalling attempts, 1 established
Implementation of MPLS network

4 activations, 3 deactivations
0 SSO recovery attempts, 0 SSO recovered
Midpoints: 1, Tails: 0
Periodic reoptimization: every 30 seconds, next in 9 seconds
Periodic FRR Promotion: Not Running
Periodic auto-bw collection: every 300 seconds, next in 13 seconds

P5#

P5#show mpls traffic-eng fast-reroute database
Readend frr information:
Protected tunnel In-label Out intf/label FRR intf/label Status
LSP identifier In-label Out intf/label FRR intf/label Status
1.1.1.1 1 [74] 28 G11/0:31 Tu51:31 ready

P7

P7#show ip interface brief
Interface IP-Address OK? Method Status Protocol
FastEthernet0/0 unassigned YES NVRAM administratively down down
FastEthernet0/1 unassigned YES NVRAM administratively down down
GigabitEthernet1/0 10.0.7.7 YES NVRAM up up
GigabitEthernet2/0 10.0.8.7 YES NVRAM up up
GigabitEthernet3/0 unassigned YES NVRAM administratively down up
GigabitEthernet4/0 unassigned YES NVRAM administratively down up
GigabitEthernet5/0 unassigned YES NVRAM administratively down up
GigabitEthernet6/0 unassigned YES NVRAM administratively down up
Loopback0 7.7.7.7 YES NVRAM up up

P7#

P7#show mpls interfaces
Interface IP Tunnel BGP Static Operational
GigabitEthernet1/0 Yes (ldp) Yes No No Yes
GigabitEthernet2/0 Yes (ldp) Yes No No Yes

P7# show ip route
Codes: 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
* - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route
Gateway of last resort is not set

1.0.0.0/32 is subnetted, 10 subnets
O 1.1.1.1 [110/101] via 10.0.7.1, 06:04:43, GigabitEthernet1/0
2.0.0.0/32 is subnetted, 1 subnets
O 2.2.2.2 [110/351] via 10.0.7.1, 06:04:33, GigabitEthernet1/0
3.0.0.0/32 is subnetted, 1 subnets
O 3.3.3.3 [110/151] via 10.0.7.1, 06:04:33, GigabitEthernet1/0
4.0.0.0/32 is subnetted, 1 subnets
O 4.4.4.4 [110/251] via 10.0.7.1, 06:04:33, GigabitEthernet1/0
5.0.0.0/32 is subnetted, 1 subnets
O 5.5.5.5 [110/201] via 10.0.7.1, 06:04:33, GigabitEthernet1/0
6.0.0.0/32 is subnetted, 1 subnets
O 6.6.6.6 [110/301] via 10.0.7.1, 06:04:33, GigabitEthernet1/0
7.0.0.0/32 is subnetted, 1 subnets
C 7.7.7.7 is directly connected, Loopback0
10.0.0.0/24 is subnetted, 10 subnets
O 10.0.10.0 [110/200] via 10.0.7.1, 06:04:33, GigabitEthernet1/0
C 10.0.8.0 is directly connected, GigabitEthernet2/0
O 10.0.9.0 [110/450] via 10.0.7.1, 06:04:33, GigabitEthernet1/0
O 10.0.2.0 [110/250] via 10.0.7.1, 06:04:33, GigabitEthernet1/0
O 10.0.3.0 [110/350] via 10.0.7.1, 06:04:33, GigabitEthernet1/0
O 10.0.1.0 [110/150] via 10.0.7.1, 06:04:33, GigabitEthernet1/0
O 10.0.6.0 [110/350] via 10.0.7.1, 06:04:33, GigabitEthernet1/0
C 10.0.7.0 is directly connected, GigabitEthernet1/0
O 10.0.4.0 [110/250] via 10.0.7.1, 06:04:33, GigabitEthernet1/0
O 10.0.5.0 [110/300] via 10.0.7.1, 05:54:02, GigabitEthernet1/0

P7# show mpls forwarding-table
Local Outgoing Prefix Bytes Label Outgoing Next Hop
### Implementation of MPLS network

<table>
<thead>
<tr>
<th>Label</th>
<th>Label or VC</th>
<th>or Tunnel Id</th>
<th>Switched</th>
<th>interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>Pop Label</td>
<td>1.1.1.1/32</td>
<td>0</td>
<td>Gi1/0</td>
</tr>
<tr>
<td>17</td>
<td>Pop Label</td>
<td>10.0.10.0/24</td>
<td>0</td>
<td>Gi1/0</td>
</tr>
<tr>
<td>18</td>
<td>Pop Label</td>
<td>10.0.1.0/24</td>
<td>0</td>
<td>Gi1/0</td>
</tr>
<tr>
<td>20</td>
<td>Pop Label</td>
<td>2.2.2.2 0 [17]</td>
<td>746392</td>
<td>Gi1/0</td>
</tr>
<tr>
<td>21</td>
<td>20</td>
<td>6.6.6.6/32</td>
<td>0</td>
<td>Gi1/0</td>
</tr>
<tr>
<td>22</td>
<td>21</td>
<td>5.5.5.5/32</td>
<td>0</td>
<td>Gi1/0</td>
</tr>
<tr>
<td>23</td>
<td>22</td>
<td>4.4.4.4/32</td>
<td>0</td>
<td>Gi1/0</td>
</tr>
<tr>
<td>24</td>
<td>23</td>
<td>3.3.3.3/32</td>
<td>0</td>
<td>Gi1/0</td>
</tr>
<tr>
<td>25</td>
<td>24</td>
<td>2.2.2.2/32</td>
<td>0</td>
<td>Gi1/0</td>
</tr>
<tr>
<td>26</td>
<td>25</td>
<td>10.0.9.0/24</td>
<td>0</td>
<td>Gi1/0</td>
</tr>
<tr>
<td>27</td>
<td>26</td>
<td>10.0.6.0/24</td>
<td>0</td>
<td>Gi1/0</td>
</tr>
<tr>
<td>28</td>
<td>27</td>
<td>10.0.3.0/24</td>
<td>0</td>
<td>Gi1/0</td>
</tr>
<tr>
<td>29</td>
<td>28</td>
<td>10.0.5.0/24</td>
<td>0</td>
<td>Gi1/0</td>
</tr>
<tr>
<td>30</td>
<td>29</td>
<td>10.0.4.0/24</td>
<td>0</td>
<td>Gi1/0</td>
</tr>
<tr>
<td>31</td>
<td>30</td>
<td>10.0.2.0/24</td>
<td>0</td>
<td>Gi1/0</td>
</tr>
</tbody>
</table>

F7#
6.3 Juniper LAB

Network environment for Juniper LAB is implemented in VMware Workstation application. It’s a well know emulation environment used for virtualization. It is possible to virtualize Juniper router with the proper image file of JunOS. For this LAB environment VSRX platform is virtualized in VMware. It is more challenging to configure network topology since everything has to be prepared manually from the importing the image file up to manually creating link connections between routers.

Lab provides implementation of the L3VPN and point-to-point L2VPN service with interworking. Unfortunately VPLS isn’t supported even through VMware virtualization.

List of the protocols running MPLS-TE VPN network:

- OSPF – for MPLS core reachability.
- LDP – for full-mesh of connectivity across MPLS core.
- MPLS-TE – TE tunnels to provide TE functionality to the network.
- RSVP – to signal LSP which requires protection.
- VRF – in order to isolate individual customers and implement L3VPN service.
- MP-BGP – protocol running the whole L3VPN concept. Providing the control plane for all routing information inside the network.
  - iBGP – for PE to PE sessions.
  - eBGP – for CE to PE sessions
- Link coloring
Figure 6-2  Juniper MPLS LAB network topology
6.3.1 Configuration

6.3.1.1 PE

Configuration for both PEs is almost the same. Thus whole configuration only for PE1 will be explained here. PE2 has the mirrored version.

PE1

system {
    host-name JR01;
}
interfaces {
    ge-0/0/0 {
        unit 0 {
            family inet {
                address 172.24.0.201/24;
            }
        }
    } ge-0/0/1 {
        unit 0 {
            family inet {
                address 10.0.7.1/24;
            }
            family mpls;
        }
    } ge-0/0/2 {
        unit 0 {
            family inet {
                address 10.0.1.1/24;
            }
            family mpls;
        }
    } ge-0/0/3 {
        unit 0 {
            family inet {
                address 10.0.10.1/24;
            }
            family mpls;
        }
    } ge-0/0/6 {
        unit 0 {
            family inet {
                address 20.0.1.1/24;
            }
        }
    } lo0 {
        unit 0 {
            family inet {
                address 1.1.1.32/32;
            }
        }
    }
} routing-options {
Implementation of MPLS network

router-id 1.1.1.1;
    autonomous-system 36500;
}
protocols {
    rsvp {
        interface ge-0/0/1.0;
        interface ge-0/0/2.0;
    }
    mpls {
        admin-groups {
            RED 1;
            BLUE 2;
        }
        icmp-tunneling;
        optimize-timer 15;
        label-switched-path PATH_1-3-5-6-2 {
            from 1.1.1.1;
            to 2.2.2.2;
            link-protection;
            fast-reroute;
            primary PRIMARY_PATH;
            secondary SECONDARY_PATH {
                admin-group include-all RED;
            }
        }
        path PRIMARY_PATH {
            3.3.3.3 strict;
            5.5.5.5 strict;
            6.6.6.6 strict;
        }
        path SECONDARY_PATH;
        interface lo0.0;
        interface ge-0/0/1.0 {
            admin-group RED;
        }
        interface all;
    }
    bgp {
        group internal-peers {
            type internal;
            local-address 1.1.1.1;
            family inet-vpn {
                unicast;
            }
            export next-hop-self;
            neighbor 2.2.2.2;
        }
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface ge-0/0/1.0 {
                interface-type p2p;
            }
            interface ge-0/0/2.0 {
                interface-type p2p;
            }
            interface ge-0/0/3.0 {
                interface-type p2p;
            }
            interface lo0.0;
        }
    }
    ldp {
        interface ge-0/0/1.0;
        interface ge-0/0/2.0;
        interface ge-0/0/3.0;
Implementation of MPLS network

interface lo0.0;
}
}
policy-options {
  policy-statement next-hop-self {
    term T1 {
      from {
        protocol bgp;
        external;
      }
      then {
        next-hop self;
      }
    }
  }
}

security {
  forwarding-options {
    family {
      mpls {
        mode packet-based;
      }
    }
  }
}

routing-instances {
  L3VPN1 {
    description "BETWEEN PE1 AND PE2";
    instance-type vrf;
    interface ge-0/0/6.0;
    route-distinguisher 1.1.1.1:100;
    vrf-target target:36500:100;
    vrf-table-label;
    protocols {
      bgp {
        group external-peers {
          type external;
          peer-as 65100;
          neighbor 20.0.1.101;
        }
      }
    }
  }
}

6.3.1.2 P

Configuration for all Ps is almost the same. Thus whole configuration only for P3 will be explained here. Remaining Ps have the very similar version of configuration.

P3

routing-options {
  router-id 3.3.3.3;
}
protocols {
  rsvp {
    interface all;
  }
  mpls {
Implementation of MPLS network

```plaintext
icmp-tunneling;
interface lo0.0;
interface all;
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface ge-0/0/1.0 {
            interface-type p2p;
        }
        interface ge-0/0/2.0 {
            interface-type p2p;
        }
        interface ge-0/0/3.0 {
            interface-type p2p;
        }
        interface lo0.0;
    }
}
ldp {
    interface ge-0/0/1.0;
    interface ge-0/0/2.0;
    interface ge-0/0/3.0;
    interface lo0.0;
}
}

6.3.1.3 CE

Configuration of CE is very simple. It is configured just to peer with particular PE through BGP and thus advertise and learn prefixes through it.

CE1

```
6.3.2 Reachability documentation

CE1

lab@CE1> show route table inet.0
inet.0: 9 destinations, 9 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

20.0.1.0/24  *[Direct/0] 01:09:02
> via ge-0/0/1.0
20.0.1.101/32 *[Local/0] 01:09:03
Local via ge-0/0/1.0
20.0.2.0/24  *[BGP/170] 01:08:51, localpref 100
AS path: 36500 I
> to 20.0.1.1 via ge-0/0/1.0
> via lo0.0
102.102.102.102/32 *[BGP/170] 01:08:51, localpref 100
AS path: 36500 65101 I
> to 20.0.1.1 via ge-0/0/1.0
172.21.0.0/24  *[Direct/0] 01:09:23, localpref 100
AS path: 36500 I
> to 20.0.2.2 via ge-0/0/1.0
172.21.0.1/32 *[Local/0] 01:09:23
Local via ge-0/0/1.0
172.24.0.0/24  *[Direct/0] 01:09:04
> via ge-0/0/0.0
172.24.0.208/32 *[Local/0] 01:09:04
Local via ge-0/0/0.0

lab@CE1>

CE2

lab@CE2> show route table inet.0
inet.0: 9 destinations, 9 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

20.0.1.0/24  *[BGP/170] 01:09:23, localpref 100
AS path: 36500 I
> to 20.0.2.2 via ge-0/0/1.0
20.0.2.0/24  *[Direct/0] 01:10:06
> via ge-0/0/0.0
20.0.2.102/32 *[Local/0] 01:10:07
Local via ge-0/0/0/1.0
AS path: 36500 65100 I
> to 20.0.2.2 via ge-0/0/1.0
102.102.102.102/32 *[Direct/0] 01:10:25
> via lo0.0
172.22.0.0/24  *[Direct/0] 01:10:06
> via ge-0/0/0.0
172.22.0.1/32 *[Local/0] 01:10:07
Local via ge-0/0/0/6.0
172.24.0.0/24  *[Direct/0] 01:10:07
> via ge-0/0/0/0.0
172.24.0.209/32 *[Local/0] 01:10:08
Local via ge-0/0/0/0.0

lab@CE2>
Implementation of MPLS network

PE1

lab@JR01> show route table inet.0
inet.0: 23 destinations, 23 routes (23 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1.1.1.1/32         *[Direct/0] 01:06:07
> via lo0.0
2.2.2.2/32         *[OSPF/10] 01:05:38, metric 2
> to 10.0.7.7 via ge-0/0/1.0
3.3.3.3/32         *[OSPF/10] 01:05:28, metric 1
> to 10.0.1.3 via ge-0/0/2.0
4.4.4.4/32         *[OSPF/10] 01:05:18, metric 2
> to 10.0.1.3 via ge-0/0/2.0
5.5.5.5/32         *[OSPF/10] 01:05:28, metric 1
> to 10.0.10.5 via ge-0/0/3.0
6.6.6.6/32         *[OSPF/10] 01:05:18, metric 3
> to 10.0.7.7 via ge-0/0/1.0
> to 10.0.1.3 via ge-0/0/2.0
7.7.7.7/32         *[OSPF/10] 01:05:38, metric 1
> to 10.0.7.7 via ge-0/0/1.0
10.0.1.0/24        *[Direct/0] 01:05:44
> via ge-0/0/2.0
10.0.1.1/32        *[Local/0] 01:05:45
Local via ge-0/0/2.0
10.0.2.0/24        *[OSPF/10] 01:05:28, metric 2
> to 10.0.1.3 via ge-0/0/2.0
10.0.3.0/24        *[OSPF/10] 01:05:18, metric 3
to 10.0.7.7 via ge-0/0/1.0
> to 10.0.1.3 via ge-0/0/2.0
10.0.4.0/24        *[OSPF/10] 01:05:28, metric 2
to 10.0.1.3 via ge-0/0/2.0
> to 10.0.10.5 via ge-0/0/3.0
10.0.5.0/24        *[OSPF/10] 01:05:18, metric 3
to 10.0.7.7 via ge-0/0/1.0
to 10.0.1.3 via ge-0/0/2.0
10.0.6.0/24        *[OSPF/10] 01:05:18, metric 3
to 10.0.1.3 via ge-0/0/2.0
10.0.7.0/24        *[Direct/0] 01:05:44
> via ge-0/0/1.0
10.0.7.1/32        *[Local/0] 01:05:45
Local via ge-0/0/1.0
10.0.8.0/24        *[OSPF/10] 01:05:38, metric 2
> to 10.0.1.3 via ge-0/0/2.0
10.0.9.0/24        *[OSPF/10] 01:05:38, metric 3
> to 10.0.7.7 via ge-0/0/1.0
10.0.10.0/24       *[Direct/0] 01:05:44
> via ge-0/0/3.0
10.0.10.1/32       *[Local/0] 01:05:45
Local via ge-0/0/3.0
172.24.0.0/24      *[Direct/0] 01:05:45
> via ge-0/0/0.0
172.24.0.201/32    *[Local/0] 01:05:50
Local via ge-0/0/0.0
224.0.0.5/32       *[OSPF/10] 01:06:08, metric 1
MultiRecv

lab@JR01> show route table inet.3
inet.3: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

2.2.2.2/32         *[LDP/9] 01:05:33, metric 1
> to 10.0.7.7 via ge-0/0/1.0, Push 299776
3.3.3.3/32         *[LDP/9] 01:05:28, metric 1
> to 10.0.1.3 via ge-0/0/2.0
4.4.4.4/32         *[LDP/9] 01:05:19, metric 1
> to 10.0.1.3 via ge-0/0/2.0, Push 299856
5.5.5.5/32         *[LDP/9] 01:05:25, metric 1
> to 10.0.10.5 via ge-0/0/3.0
6.6.6.6/32         *[LDP/9] 01:05:19, metric 1
to 10.0.7.7 via ge-0/0/3.0, Push 299808
> to 10.0.1.3 via ge-0/0/2.0, Push 299840
7.7.7.7/32         *[LDP/9] 01:05:33, metric 1
Implementation of MPLS network

lab@JR01> show mpls lsp detail
Ingress LSP: 1 sessions
2.2.2.2
From: 1.1.1.1, State: Dn, ActiveRoute: 0, LSPname: PATH_1-3-5-6-2
ActivePath: (none)
FastReroute desired
Link protection desired
LSPtype: Static Configured
LoadBalance: Random
Encoding type: Packet, Switching type: Packet, GPID: IPv4
Primary PRIMARY_PATH State: Dn
Priorities: 7 0
OptimizeTimer: 15
SmartOptimizeTimer: 180
Will be enqueued for recomputation in 15 second(s).
2 May 23 06:17:57.514 CSPF failed: no route toward 6.6.6.6
Secondary SECONDARY_PATH State: Dn
Priorities: 7 0
OptimizeTimer: 15
SmartOptimizeTimer: 180
Include All: RED
Will be enqueued for recomputation in 15 second(s).
1 May 23 06:17:57.514 CSPF failed: no route toward 2.2.2.2
Total 1 displayed, Up 0, Down 1
Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
lab@JR01> show bgp summary
Groups: 2 Peers: 2 Down peers: 0
Table Tot Paths Act Paths Suppressed History Damp State Pending
bgp.l3vpn.0 2 2 0 0 0 0
Peer AS InPkt OutPkt OutQ Flaps Last Up/Dwn
2.2.2.2 36500 148 150 0 0 1:05:31 Establ
bgp.l3vpn.0: 2/2/2/0
L3VPN1.inet.0: 2/2/2/0
20.0.1.101 65100 144 150 0 0 1:05:40 Establ
L3VPN1.inet.0: 1/1/1/0
lab@JR01> show route table bgp.l3vpn.0
bgp.l3vpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2.2.2.2:100:200.1.2.0/24
+ [bgp/170] 01:05:31, localpref 100, from 2.2.2.2
AS path: I
> to 10.0.7.7 via ge-0/0/1.0, Push 16, Push 299776 (top)
2.2.2.2:100:120.120.120.120/32
+ [bgp/170] 01:05:31, localpref 100, from 2.2.2.2
AS path: 65101 1
> to 10.0.7.7 via ge-0/0/1.0, Push 16, Push 299776 (top)
lab@JR01>

PE2

lab@JR02> show route table inet.0
inet.0: 23 destinations, 23 routes (23 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
1.1.1.1/32 * [OSPF/10] 01:07:38, metric 2
> to 10.0.8.7 via ge-0/0/1.0
2.2.2.2/32 * [Direct/0] 01:09:10
> via lo0.0
3.3.3.3/32 * [OSPF/10] 01:07:28, metric 2
Implementation of MPLS network

4.4.4.4/32  *[OSPF/10] 01:07:33, metric 1
> to 10.0.3.4 via ge-0/0/2.0
5.5.5.5/32  *[OSPF/10] 01:07:28, metric 3
> to 10.0.8.7 via ge-0/0/1.0
> to 10.0.3.4 via ge-0/0/2.0
6.6.6.6/32  *[OSPF/10] 01:07:38, metric 1
> to 10.0.9.6 via ge-0/0/3.0
7.7.7.7/32  *[OSPF/10] 01:08:26, metric 1
> to 10.0.8.7 via ge-0/0/1.0
10.0.1.0/24 *[OSPF/10] 01:07:28, metric 3
> to 10.0.3.4 via ge-0/0/2.0
10.0.2.0/24 *[OSPF/10] 01:07:33, metric 2
> to 10.0.3.4 via ge-0/0/2.0
10.0.3.0/24 *[Direct/0] 01:08:49
> via ge-0/0/2.0
10.0.3.2/32 *[Local/0] 01:08:51
Local via ge-0/0/2.0
10.0.4.0/24 *[OSPF/10] 01:07:28, metric 3
> to 10.0.3.4 via ge-0/0/2.0
10.0.5.0/24 *[OSPF/10] 01:07:38, metric 2
> to 10.0.9.6 via ge-0/0/3.0
> to 10.0.3.4 via ge-0/0/2.0
10.0.6.0/24 *[OSPF/10] 01:07:33, metric 2
> to 10.0.8.7 via ge-0/0/1.0
> to 10.0.3.4 via ge-0/0/2.0
10.0.7.0/24 *[OSPF/10] 01:08:26, metric 2
> to 10.0.8.7 via ge-0/0/1.0
10.0.8.0/24 *[Direct/0] 01:08:49
> via ge-0/0/1.0
10.0.8.2/32 *[Local/0] 01:08:51
Local via ge-0/0/1.0
10.0.9.0/24 *[Direct/0] 01:08:49
> via ge-0/0/3.0
10.0.9.2/32 *[Local/0] 01:08:51
Local via ge-0/0/3.0
10.0.10.0/24 *[OSPF/10] 01:07:38, metric 3
> to 10.0.8.7 via ge-0/0/1.0
172.24.0.0/24 *[Direct/0] 01:08:50
> via ge-0/0/0.0
172.24.0.202/32 *[Local/0] 01:08:51
Local via ge-0/0/0.0
224.0.0.5/32 *[OSPF/10] 01:09:10, metric 1
MultiRecv

lab@JR02> show route table inet.3
inet.3: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
1.1.1.1/32  *[LDP/9] 01:07:38, metric 1
> to 10.0.8.7 via ge-0/0/1.0, Push 299792
3.3.3.3/32  *[LDP/9] 01:07:25, metric 1
> to 10.0.3.4 via ge-0/0/2.0, Push 299840
4.4.4.4/32  *[LDP/9] 01:07:25, metric 1
> to 10.0.3.4 via ge-0/0/2.0
5.5.5.5/32  *[LDP/9] 01:07:25, metric 1
> to 10.0.8.7 via ge-0/0/1.0, Push 299840
to 10.0.3.4 via ge-0/0/2.0, Push 299856
6.6.6.6/32  *[LDP/9] 01:07:16, metric 1
> to 10.0.9.6 via ge-0/0/3.0
7.7.7.7/32  *[LDP/9] 01:08:24, metric 1
> to 10.0.8.7 via ge-0/0/1.0

lab@JR02> show mpls lsp detail
Ingress LSP: 1 sessions
1.1.1.1
From: 2.2.2.2, State: Dn, ActiveRoute: 0, LSName: PATH_2-6-5-3-1
ActivePath: (none)
LSPType: Static Configured
LoadBalance: Random
Encoding type: Packet, Switching type: Packet, GPID: IPv4
Primary PRIMARY_PATH State: Dn
Priorities: 7 0
SmartOptimizeTimer: 180
Will be enqueued for recomputation in 4 second(s).
Implementation of MPLS network

2 May 23 06:19:51.220 CSPF failed: no route toward 5.5.5.5[139 times]
Secondary SECONDARY_PATH State: Do
Priorities: 7 0
SmartOptimizeTimer: 180
Include All: RED
Will be enqueued for recomputation in 4 second(s).
1 May 23 06:19:51.220 CSPF failed: no route toward 1.1.1.1[139 times]
Total 1 displayed, Up 0, Down 1

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

lab@JR02> show bgp summary
Groups: 2 Peers: 2 Down peers: 0
Table Tot Paths Act Paths Suppressed History Damp State Pending
bgp.l3vpn.0 2 2 0 0 0

Peer AS InPkt OutPkt OutQ Flaps Last Up/Dwn
1.1.1.1 36500 154 154 0 0 1:07:36 Establ

lab@JR02> show route table bgp.l3vpn.0
bgp.l3vpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
1.1.1.1:100:20.0.1.0/24
* [OSPF/10] 01:08:13, metric 1
  > to 10.0.1.1 via ge-0/0/1.0
1.1.1.1:100:101.101.101.101/32
* [BGP/170] 01:07:36, localpref 100, from 1.1.1.1
  AS path: I
  > to 10.0.8.7 via ge-0/0/1.0, Push 16, Push 299792 (top)

lab@JR03> show route table inet.0
inet.0: 23 destinations, 23 routes (23 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
1.1.1.1/32
* [OSPF/10] 01:08:13, metric 1
  > to 10.0.1.1 via ge-0/0/1.0
2.2.2.2/32
* [OSPF/10] 01:08:08, metric 2
  > to 10.0.2.4 via ge-0/0/2.0
3.3.3.3/32
* [Direct/0] 01:08:53
  > via lo0.0
4.4.4.4/32
* [OSPF/10] 01:08:08, metric 1
  > to 10.0.2.4 via ge-0/0/2.0
5.5.5.5/32
* [OSPF/10] 01:08:23, metric 1
  > to 10.0.4.5 via ge-0/0/3.0
6.6.6.6/32
* [OSPF/10] 01:08:08, metric 2
  > to 10.0.2.4 via ge-0/0/2.0
7.7.7.7/32
* [OSPF/10] 01:08:13, metric 2
  > to 10.0.1.1 via ge-0/0/1.0
10.0.1.0/24
* [Direct/0] 01:08:28
  > via ge-0/0/1.0
10.0.1.3/32
* [Local/0] 01:08:13
  Local via ge-0/0/1.0
10.0.2.0/24
* [Direct/0] 01:08:28
  > via ge-0/0/2.0
10.0.2.3/32
* [Local/0] 01:08:29
  Local via ge-0/0/2.0
10.0.3.0/24
* [OSPF/10] 01:08:08, metric 2
  > to 10.0.2.4 via ge-0/0/2.0

123
Implementation of MPLS network

10.0.4.0/24       *[Direct/0] 01:08:28
   > via ge-0/0/3.0
10.0.4.3/32       *[Local/0] 01:08:29
   Local via ge-0/0/3.0
10.0.5.0/24       *[OSPF/10] 01:08:08, metric 3
   > to 10.0.2.4 via ge-0/0/2.0
10.0.6.0/24       *[OSPF/10] 01:08:08, metric 2
   > to 10.0.2.4 via ge-0/0/2.0
10.0.7.0/24       *[OSPF/10] 01:08:13, metric 2
   > to 10.0.1.1 via ge-0/0/1.0
10.0.8.0/24       *[OSPF/10] 01:08:08, metric 3
   > to 10.0.2.4 via ge-0/0/2.0
10.0.9.0/24       *[OSPF/10] 01:08:08, metric 3
   > to 10.0.2.4 via ge-0/0/2.0
10.0.10.0/24      *[OSPF/10] 01:08:13, metric 2
   > to 10.0.1.1 via ge-0/0/1.0
   > to 10.0.4.5 via ge-0/0/3.0
172.24.0.0/24     *[Direct/0] 01:08:29
   > via ge-0/0/0.0
172.24.0.203/32   *[Local/0] 01:08:30
   Local via ge-0/0/0.0
224.0.0.5/32      *[OSPF/10] 01:08:54, metric 1
   MultiRecv

lab@JR03> show route table inet.3
inet.3: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
1.1.1.1/32        *[LDP/9] 01:08:14, metric 1
   > to 10.0.1.1 via ge-0/0/1.0
2.2.2.2/32        *[LDP/9] 01:08:06, metric 1
   > to 10.0.2.4 via ge-0/0/2.0, Push 299792
4.4.4.4/32        *[LDP/9] 01:08:06, metric 1
   > to 10.0.2.4 via ge-0/0/2.0
5.5.5.5/32        *[LDP/9] 01:08:21, metric 1
   > to 10.0.4.5 via ge-0/0/3.0
6.6.6.6/32        *[LDP/9] 01:08:06, metric 1
   > to 10.0.2.4 via ge-0/0/2.0, Push 299776
7.7.7.7/32        *[LDP/9] 01:08:14, metric 1
   > to 10.0.1.1 via ge-0/0/1.0, Push 299776

lab@JR03>


6.4 Failover functionality

One of the main goals of this thesis is to provide results how the failover functionality available in MPLS preserve the connectivity across the MPLS network. For the testing purposes I created an Excel GUI which is using program Nping in order to test how fast the tested network connection can recover. It will run the program with defined parameters through the GUI and once finished it will parse the RAW data from the Nping into separate Excel spreadsheet with the graph representing measured values.

For the testing purposes of failover functionality I choose Cisco LAB implementation. The main reason is the failover functionality is working really close to real world implementation. Unfortunately with emulating Juniper LAB in VMware I faced the limitations of this environment which couldn’t be overcome.

MPLS network was tested to the 3 following LSP protection scenarios:

- End-to-end protection no pre-signaled.
- Pre-signaled end-to-end protection.
- Local protection against link failure.
- Local protection against node failure.

6.4.1 Running test through Excel GUI

Firstly we need to update routing table of PC on which we are emulating the LAB. This has to be done to make Nping program sends ICMP messages through the particular Windows loopback interface which is being represented as a host in the topology. The destination is the loopback interface of a particular CE device. Totally 3 different static routes needs to be added. One per Host1 and Host2 and one for the CE loopback reachability.

Once the routing table of a local PC is updated of all needed routes we can run a test. From the testing parameters timeout never changed, it was always 200 ms. Delay varied depending whether tuned RSVP hello was present. Tuned settings for RSVP hello for the interface:

```
ip rsvp signalling hello refresh interval 50
ip rsvp signalling hello dscp 30
```
Implementation of MPLS network

![Image of routing table and stress test Excel GUI]

**Figure 6-3** Excel GUI stressTest
6.4.2 Traceroutes

6.4.2.1 Bypass tunnel for P5-P6 link protection

PE1#traceroute 2.2.2.2
Type escape sequence to abort.
Tracing the route to 2.2.2.2

1 10.0.1.3 [MPLS: Label 16 Exp 0] 36 msec 28 msec 24 msec
2 10.0.4.5 [MPLS: Label 31 Exp 0] 16 msec 20 msec 24 msec
3 10.0.5.6 [MPLS: Label 26 Exp 0] 48 msec 24 msec 36 msec
4 10.0.9.2 48 msec 32 msec 12 msec
PE1#

PE1#traceroute 2.2.2.2
Type escape sequence to abort.
Tracing the route to 2.2.2.2

1 10.0.1.3 [MPLS: Label 16 Exp 0] 96 msec 52 msec 72 msec
2 10.0.4.5 [MPLS: Label 31 Exp 0] 68 msec 56 msec 60 msec
3 10.0.4.3 [MPLS Labels 31/26 Exp 0] 48 msec 48 msec 52 msec
4 10.0.2.4 [MPLS: Labels 31/26 Exp 0] 44 msec 44 msec 60 msec
5 10.0.6.6 [MPLS: Label 26 Exp 0] 48 msec 48 msec 48 msec
6 10.0.9.2 44 msec 52 msec 48 msec
PE1#

PE1#traceroute 2.2.2.2
Type escape sequence to abort.
Tracing the route to 2.2.2.2

1 10.0.7.7 [MPLS: Label 32 Exp 0] 48 msec 28 msec 28 msec
2 10.0.8.2 24 msec 44 msec 48 msec
PE1#

6.4.2.2 Bypass tunnel for P5 node protection

PE1#traceroute 2.2.2.2
Type escape sequence to abort.
Tracing the route to 2.2.2.2

1 10.0.1.3 [MPLS: Label 32 Exp 0] 52 msec 20 msec 48 msec
2 10.0.4.5 [MPLS: Label 32 Exp 0] 56 msec 56 msec 64 msec
3 10.0.5.6 [MPLS: Label 30 Exp 0] 48 msec 44 msec 48 msec
4 10.0.9.2 48 msec 48 msec 48 msec
PE1#

PE1#traceroute 2.2.2.2
Type escape sequence to abort.
Tracing the route to 2.2.2.2

1 10.0.1.3 [MPLS: Label 32 Exp 0] 68 msec 28 msec 44 msec
2 10.0.2.4 [MPLS: Labels 16/30 Exp 0] 52 msec 64 msec 52 msec
3 10.0.6.6 [MPLS: Label 30 Exp 0] 48 msec 48 msec 44 msec
4 10.0.9.2 48 msec 60 msec 24 msec
PE1#

PE1#traceroute 2.2.2.2
Type escape sequence to abort.
Tracing the route to 2.2.2.2

1 10.0.7.7 [MPLS: Label 31 Exp 0] 32 msec 32 msec 28 msec
2 10.0.8.2 24 msec 28 msec 28 msec
PE1#
6.4.3 Results of testing the failover functionality in MPLS

6.4.3.1 End-to-end protection no pre-signaled

Link error detected locally

![Figure 6-4](image.png)  
**Figure 6-4** End-to-end protection no pre-signaled, local link error

Link error detected through RSVP hello (default hello settings)

![Figure 6-5](image.png)  
**Figure 6-5** End-to-end protection no pre-signaled, remote link error and default hello
Implementation of MPLS network

Link error detected through RSVP hello (tuned hello settings)

6.4.3.2 Pre-signaled end-to-end protection

Link error detected locally

Figure 6-6  End-to-end protection no pre-signaled, remote link error and tuned hello

Figure 6-7  End-to-end protection pre-signaled, local link error
Implementation of MPLS network

Link error detected through RSVP hello (default hello settings)

![Graph](image1)

*Figure 6-8*  End-to-end protection pre-signaled, remote link error and default hello

Link error detected through RSVP hello (tuned hello settings)

![Graph](image2)

*Figure 6-9*  End-to-end protection pre-signaled, remote link error and tuned hello
6.4.3.3 Local protection against link failure

Link error detected locally

Figure 6-10 Local protection against link, local link error

Link error detected through RSVP hello (default hello settings)

Figure 6-11 Local protection against link, remote link error and default hello
Implementation of MPLS network

Link error detected through RSVP hello (tuned hello settings)

6.4.3.4 Local protection against node failure

Link error detected locally

Figure 6-12  Local protection against link, remote link error and tuned hello

Figure 6-13  Local protection against node, local link error
Implementation of MPLS network

Link error detected through RSVP hello (default hello settings)

Figure 6-14  Local protection against node, remote link error and default hello

Link error detected through RSVP hello (tuned hello settings)

Figure 6-15  Local protection against node, remote link error and tuned hello
6.5 Wireshark captures

Some of the Wireshark captures will be shown here providing the real evidence of how is the information being carried inside the protocol. Full RAW Wireshark captures are in the attachments as well as all individual ones.

**RSVP Path message**

[Figure 6-16 RSVP Path message]

**RSVP Resv message**

[Figure 6-17 RSVP Resv message]
Implementation of MPLS network

RSVP PathErr message

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000000000</td>
<td>10.0.1.1</td>
<td>10.0.1.1</td>
<td>RSVP</td>
<td>168 PATH ERROR MESSAGE. SESSION: IPv4-LSP, DESTINATION</td>
<td></td>
</tr>
</tbody>
</table>

![Figure 6-18 RSVP PathErr message](image)

RSVP PathTear message

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000000000</td>
<td>1.1.1.1</td>
<td>2.2.2.2</td>
<td>RSVP</td>
<td>168 PATH TEAR MESSAGE. SESSION: IPv4-LSP, DESTINATION</td>
<td></td>
</tr>
</tbody>
</table>

![Figure 6-19 RSVP PathTear message](image)

Traceroute (L3VPN)

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000000000</td>
<td>UDP</td>
<td>68 Source port: 49161 Destination port: 33441</td>
<td></td>
</tr>
</tbody>
</table>

![Figure 6-20 Traceroute L3VPN](image)
Implementation of MPLS network

Traceroute (L2VPN)

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000000000</td>
<td>4.0.0.1</td>
<td>4.0.0.2</td>
<td>UDP</td>
<td>8</td>
<td>Source port: 49158 Destination port: traceroute</td>
</tr>
</tbody>
</table>

![Figure 6-21 Traceroute L2VPN](image)

BGP Update message

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000000000</td>
<td>1.1.1.1</td>
<td>2.2.2.2</td>
<td>BGP</td>
<td>587</td>
<td>UPDATE Message, UPDATE Message, UPDATE Message</td>
</tr>
</tbody>
</table>

![Figure 6-22 BGP Update message](image)
I Bibliography


## Used Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AD</td>
<td>Administrative Distance</td>
</tr>
<tr>
<td>APS</td>
<td>Automatic Protection Switching</td>
</tr>
<tr>
<td>AS</td>
<td>Autonomous system</td>
</tr>
<tr>
<td>ASA</td>
<td>Adaptive Security Appliance</td>
</tr>
<tr>
<td>ASIC</td>
<td>Application Specific Integrated Circuit</td>
</tr>
<tr>
<td>ATM</td>
<td>Asynchronous Transfer Mode</td>
</tr>
<tr>
<td>BFD</td>
<td>Bidirectional Forwarding Detection</td>
</tr>
<tr>
<td>BGP</td>
<td>Border Gateway Protocol</td>
</tr>
<tr>
<td>CAM</td>
<td>Content Addressable Memory</td>
</tr>
<tr>
<td>CE</td>
<td>Customer Edge</td>
</tr>
<tr>
<td>CEF</td>
<td>Cisco Express Forwarding</td>
</tr>
<tr>
<td>Cisco IOS</td>
<td>Cisco Internetwork Operating System</td>
</tr>
<tr>
<td>Cisco PIX</td>
<td>Cisco Private Internet eXchange</td>
</tr>
<tr>
<td>CLR</td>
<td>Conservative Label Retention mode</td>
</tr>
<tr>
<td>CoS</td>
<td>Class of Service</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>CRC</td>
<td>Cyclic Redundancy Check</td>
</tr>
<tr>
<td>CSPF</td>
<td>Constrained SPF</td>
</tr>
<tr>
<td>CW</td>
<td>Control Word</td>
</tr>
<tr>
<td>DiffServ</td>
<td>Differentiated Services</td>
</tr>
<tr>
<td>DLCI</td>
<td>Data Link Connection Identifier</td>
</tr>
<tr>
<td>DSCP</td>
<td>DiffServ Code Point</td>
</tr>
<tr>
<td>eBGP</td>
<td>External Border Gateway Protocol</td>
</tr>
<tr>
<td>EC</td>
<td>Extended Community</td>
</tr>
<tr>
<td>EGP</td>
<td>Exterior Gateway Protocol</td>
</tr>
<tr>
<td>EIGRP</td>
<td>Enhanced Interior Gateway Routing Protocol</td>
</tr>
<tr>
<td>E-LDP</td>
<td>EXP based LSP</td>
</tr>
<tr>
<td>ERO</td>
<td>Explicit Route Object</td>
</tr>
<tr>
<td>EXP</td>
<td>Experimental</td>
</tr>
<tr>
<td>FEC</td>
<td>Forwarding Equivalency Class</td>
</tr>
<tr>
<td>FIB</td>
<td>Forwarding Information Base</td>
</tr>
<tr>
<td>FRO</td>
<td>FRR Object</td>
</tr>
<tr>
<td>FRR</td>
<td>Fast Reroute</td>
</tr>
<tr>
<td>GNS3</td>
<td>Graphical Network Simulator</td>
</tr>
<tr>
<td>GRE</td>
<td>Generic Routing Encapsulation</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>IGP</td>
<td>Internal Gateway Protocol</td>
</tr>
<tr>
<td>IntServ</td>
<td>Integrated Services</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>IPS</td>
<td>Intrusion Prevention System</td>
</tr>
<tr>
<td>IPsec</td>
<td>IP Security</td>
</tr>
<tr>
<td>IPv4</td>
<td>Internet Protocol version 4</td>
</tr>
<tr>
<td>IPv6</td>
<td>Internet Protocol version 6</td>
</tr>
<tr>
<td>IS-IS</td>
<td>Intermediate System to Intermediate System</td>
</tr>
<tr>
<td>ISP</td>
<td>Internet Service Provider</td>
</tr>
<tr>
<td>JunOS</td>
<td>Juniper Operating System</td>
</tr>
<tr>
<td>L2VPN</td>
<td>Layer 2 VPN</td>
</tr>
</tbody>
</table>
### Used Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>L3VPN</td>
<td>Layer 3 VPN</td>
</tr>
<tr>
<td>LAB</td>
<td>Laboratory</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>LDP</td>
<td>Label Distribution Protocol</td>
</tr>
<tr>
<td>LFIB</td>
<td>Label Forwarding Information Base</td>
</tr>
<tr>
<td>LLR</td>
<td>Liberal Label Retention mode</td>
</tr>
<tr>
<td>L-LSP</td>
<td>Label-inferred based LSP</td>
</tr>
<tr>
<td>LSP</td>
<td>Label Switch Path</td>
</tr>
<tr>
<td>LSR</td>
<td>Label Switch Router</td>
</tr>
<tr>
<td>MAC</td>
<td>Media Access Control</td>
</tr>
<tr>
<td>MP</td>
<td>Merge Point</td>
</tr>
<tr>
<td>MP-BGP</td>
<td>Multiprotocol BGP</td>
</tr>
<tr>
<td>MPLS</td>
<td>Multiprotocol Label Switching</td>
</tr>
<tr>
<td>NAT</td>
<td>Network Address Translate</td>
</tr>
<tr>
<td>NLRI</td>
<td>Network Layer Reachability Information</td>
</tr>
<tr>
<td>OSPF</td>
<td>Open Shortest Path First</td>
</tr>
<tr>
<td>P</td>
<td>Provider</td>
</tr>
<tr>
<td>PA</td>
<td>Path Attribute</td>
</tr>
<tr>
<td>Path</td>
<td>Path message</td>
</tr>
<tr>
<td>PathErr</td>
<td>Path Error message</td>
</tr>
<tr>
<td>PathTear</td>
<td>Path Teardown message</td>
</tr>
<tr>
<td>PE</td>
<td>Provider Edge</td>
</tr>
<tr>
<td>PHB</td>
<td>Per Hop Behavior</td>
</tr>
<tr>
<td>PHP</td>
<td>Penultimate Hop Popping</td>
</tr>
<tr>
<td>PLR</td>
<td>Point of Local Repair</td>
</tr>
<tr>
<td>PPP</td>
<td>Point-to-point protocol</td>
</tr>
<tr>
<td>PSTN</td>
<td>Public Switched Telephone Network</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RAW</td>
<td>Reading And Writing</td>
</tr>
<tr>
<td>RD</td>
<td>Route Distinguisher</td>
</tr>
<tr>
<td>Resv</td>
<td>Reservation message</td>
</tr>
<tr>
<td>ResvErr</td>
<td>Reservation Error message</td>
</tr>
<tr>
<td>ResvTear</td>
<td>Reservation Teardown message</td>
</tr>
<tr>
<td>RFC</td>
<td>Request For Comments</td>
</tr>
<tr>
<td>RIB</td>
<td>Routing Information Base</td>
</tr>
<tr>
<td>RR</td>
<td>Route Reflector</td>
</tr>
<tr>
<td>RRO</td>
<td>Record Route Object</td>
</tr>
<tr>
<td>RRO-TE</td>
<td>Resource Reservation Protocol - Traffic Engineering</td>
</tr>
<tr>
<td>RT</td>
<td>Route Target</td>
</tr>
<tr>
<td>SAO</td>
<td>Session Attribute Object</td>
</tr>
<tr>
<td>SDH</td>
<td>Synchronous Digital Hierarchy</td>
</tr>
<tr>
<td>SONET</td>
<td>Synchronous Optical Networking</td>
</tr>
<tr>
<td>SPF</td>
<td>Shortest Path First</td>
</tr>
<tr>
<td>SRLG</td>
<td>Shared Risk Link Group</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>TDM</td>
<td>Time Division Multiplex</td>
</tr>
<tr>
<td>TED</td>
<td>Traffic Engineering Database</td>
</tr>
<tr>
<td>ToS</td>
<td>Type of Service</td>
</tr>
<tr>
<td>TTL</td>
<td>Time To Live</td>
</tr>
<tr>
<td>VC</td>
<td>Virtual Circuit</td>
</tr>
<tr>
<td>VCI</td>
<td>VC identifier</td>
</tr>
<tr>
<td>VPI/VCI</td>
<td>Virtual Path Identifier / Virtual Channel Identifier</td>
</tr>
<tr>
<td>VPLS</td>
<td>Virtual Private LAN Services</td>
</tr>
<tr>
<td>VRF</td>
<td>Virtual Routing and Forwarding</td>
</tr>
<tr>
<td>VPN</td>
<td>Virtual Private Network</td>
</tr>
<tr>
<td>WAN</td>
<td>Wide Area Network</td>
</tr>
</tbody>
</table>
III  Content of DVD

- Captures_RAW – directory contains complete RAW captures from the MPLS network.
- Captures_Specific – directory contains specific captures from the MPLS network.
- Cisco_config – directory contains configuration files from Cisco LAB.
- Juniper_config – directory contains configuration files from Juniper LAB.
- NetStress – directory of an Excel GUI for Nping.
- NetStress/bin – directory of binary file Nping.
- NetStress/Nping GUI.xlsm – excel file containing GUI and all the RAW data from tests and graphs.