Universidade de Aveiro 2010

Departamento de Electrónica, Telecomunicações e Informática

João Pedro Morais Damms Machado

Design and Implementation of MPLS Laboratorial Scenarios

Concepção e Implementação de Experiências Laboratoriais sobre MPLS

Universidade de Aveiro 2010

Departamento de Electrónica,
Telecomunicações e Informática

João Pedro Morais Damms Machado

Design and Implementation of MPLS Laboratorial Scenarios

Concepção e implementação de experiências laboratoriais sobre MPLS

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica do Doutor António Manuel Duarte Nogueira, Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

o júri

Presidente	Doutor Rui Luís Andrade Aguiar Professor Associado da Universidade de Aveiro
Vogais	
	Doutor António Manuel de Jesus Pereira Professor Adjunto do Instituto Politécnico de Leiria
	Doutor António Manuel Duarte Nogueira Professor Auxiliar da Universidade de Aveiro

agradecimentos

A dissertação, cujo objectivo académico lhe confere um carácter solitário, jamais seria realizável na ausência de contributos e estímulos externos, não fosse o Homem um ser eminentemente social. Por essa razão, desejo expressar os meus sinceros agradecimentos:

Ao Professor António Nogueira por todo o apoio prestado, na sua incansável disponibilidade, na prontidão da sua resposta ás inúmeras questões, no valioso contributo tanto nas sugestões bem como nas ideias despertadas. De realçar a extraordinária paciência e boa disposição que sempre me prestou, bem como a liberdade de movimento, uma vez que uma boa porção deste processo se desenvolveu com 2000 Km de distância métrica, mas na proximidade iminente da ajuda prestada.

À minha esposa por toda a paciência e ajuda paralela prestada, uma vez que a vida se estende para além do teclado.

À minha filha, surgida no decorrer deste processo, por me ter mostrado que a vida é uma luta constante e que cada pequeno passo é significativo no longo caminho.

palavras-chave MPLS, VPN, QoS, Engenharia de Tráfego, IPv6, Ethernet, Label-switching, LSR, LDP, RSVP BGP, OSPF, Proteção, AToM

resumo

O Multiprotocol Label Switching (MPLS) é um mecanismo de transporte de dados, sob a forma de um protocolo agnóstico, com grande potencial de crescimento e adequação. Opera na "Camada 2.5" do modelo OSI e constitui um mecanismo de alto desempenho utilizado nas redes de núcleo para transportar dados de um nó da rede para outro. O sucesso do MPLS resulta do facto de permitir que a rede transporte todos os tipos de dados, desde tráfego IP a tráfego da camada de ligação de dados, devido ao encapsulamento dos pacotes dos diversos protocolos, permitindo a criação de "links virtuais" entre nós distantes.

O MPLS pertence à família das "redes de comutação de pacotes", sendo os pacotes de dados associados a "etiquetas" que determinam o seu encaminhamento, sem necessidade de examinar o conteúdo dos próprios pacotes. Isto permite a criação de circuitos "extremo-a-extremo" através de qualquer tipo de rede de transporte e independentemente do protocolo de encaminhamento que é utilizado.

O projecto do MPLS considera múltiplas tecnologias no sentido de prestar um serviço único de transporte de dados, tentando simultaneamente proporcionar capacidades de engenharia de tráfego e controlo "out-of-band", uma característica muito atraente para uma implementação em grande escala. No fundo, o MPLS é uma forma de consolidar muitas redes IP dentro de uma única rede.

Dada a importância desta tecnologia, é urgente desenvolver ferramentas que permitam entender melhor a sua complexidade. O MPLS corre normalmente nas redes de núcleo dos ISPs. No sentido de tornar o seu estudo viável, recorreu-se nesta dissertação à emulação para implementar cenários de complexidade adequada. Existem actualmente boas ferramentas disponíveis que permitem a recriação em laboratório de cenários bastante complicados.

Contudo, a exigência computacional da emulação é proporcional à complexidade do projecto em questão, tornando-se rapidamente impossível de realizar numa única máquina. A computação distribuída ou a "Cloud Computing" são actualmente as abordagens mais adequadas e inovadoras apara a resolução deste problema.

Esta dissertação tem como objectivo criar algumas experiências em laboratório que evidenciam aspectos relevantes da tecnologia MPLS, usando para esse efeito um emulador computacional, o Dynamips, impulsionado por generosas fontes computacionais disponibilizadas pela Amazon ec2. A utilização destas ferramentas de emulação permite testar cenários de rede e serviços reais em ambiente controlado, efectuando o *debugging* das suas configurações e optimizando o seu desempenho, antes de os colocar em funcionamento nas redes em operação.

keywords MPLS, VPN, QoS, Traffic engineering, IPv6, Ethernet, Label-switching, LSR, LDP, RSVP BGP, OSPF, Protection, AToM

abstract

The Multiprotocol Label Switching (MPLS) is a highly scalable and agnostic protocol to carry network data.

Operating at "Layer 2.5" of the OSI model, MPLS is an high-performance mechanism that is used at the network backbone for conveying data from one network node to the next.

The success of MPLS results from the fact that it enables the network to carry all kinds of traffic, ranging from IP to layer 2 traffic, since it encapsulates the packets of the diverse network protocols, allowing the creation of "virtual links" between distant nodes.

MPLS belongs to the family of packet switched networks, where labels are assigned to data packets that are forwarded based on decisions that rely only on the label contents, without the need to examine the packets contents. This allows the creation of end-to-end circuits across any type of transport medium, using any protocol.

The MPLS design takes multiform transport technologies into account to provide a unified data-carrying service, attempting simultaneously to preserve traffic engineering and out-of-band control, a very attractive characteristic for large-scale deployment. MPLS is the way to consolidate many IP networks into a single one. Due to this obvious potential, it is urgent to develop means and tools to better understand its functioning and complexity.
MPLS normally runs at the backbone of Service Providers networks, being deployed across an extensive set of expensive equipment. In order to turn the study of MPLS feasible, emulation was considered as the best solution. Currently, there are very good available tools to recreate, in a lab environment, quite complicated scenarios.

However, the computational demand of the emulation is proportional to the complexity of the project, becoming quickly unfeasible in a single machine.

Fortunately, distributed computing or Cloud computing are suitable and novel approaches to solve this computation problem.

So, this work aims to create some lab experiments that can illustrate/demonstrate relevant aspects of the MPLS technology, using the Dynamips emulator driven by the computational resources that were made available by the Amazon ec2 cloud computing facilities. The utilization of these emulation tools allows testing real networks and service scenarios in a controlled environment, being able to debug their configurations and optimize their performance before deploying them in real operating networks.

Table of contents

Index of Illustrations

List of abbreviations

AMI	Amazon Machine Image
API	Application programming interface
AS	Autonomous System
ASBR	Autonomous Systems Boundary Routers
ATM	Asynchronous Transfer Mode
AToM	Any Transport over Multiprotocol Label Switching
Bc	committed Burst
Be	excess Burst
BGP	Border Gateway Protocol
CAR	Committed Access Rate
CCNA	Cisco Certified Network Associate certification
CE	Customer Edge
CEF	Cisco Express Forwarding
COPS	Common Open Policy Server
CoS	Classes of Service
CPU	Central Processing Unit
CR-LDP	Constraint-based Routed LDP
CSPF	Constrained Shortest Path First
DiffServ	Differentiated Services
DLCI	Data-Link Connection Identifier
DSCP	Differentiated Services Code Point
eBGP	external BGP
ECR	Egress Committed Rate
EIGRP	Enhanced Interior Gateway Routing Protocol
FEC	Forwarding Equivalence Class
FRR	Fast Reroute
FRTS	Frame Relay Traffic Shaping
GNS-3	Graphical Network Simulator
GPL	General Public License
GTS	Generic Traffic Shaping
GUI	Graphical User Interface

HDLC	High Level Data Link Control
I/O	Input/Output
iBGP	internal BGP
ICR	Ingress Committed Rate
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol
IntServ	Integrated Services
IOS	Internetwork Operating System
ISP	Internet Service Provider
IS-IS	Intermediate System To Intermediate System
L2TPv3	Layer 2 Tunneling Protocol Version 3
L3VPN	Layer 3 Virtual Private Networks
LAN	Local Area Network
LDP	Label Distribution Protocol
LFIB	Label Forwarding Information Base
LIB	Label Information Base
LSR	Label Switched Router
MP-BGP	Multiprotocol BGP extensions
MPLS	Multiprotocol Label Switching
NNHOP	Next-Next-Hop
NVRAM	Non-Volatile Random Access Memory
OOP	Object-oriented programming
OSPF	Open Shortest Path First
P	Provider
PBR	Policy-Based Routing
PE	Provider
PHB	Per-Hop Behavior
PIM	Protocol-Independent Multicast
POP	Point of Presence
PSN	Packet-Switched Network
QoS	Quality of Service
RADIUS	Remote Authentication Dial In User Service
RD	Route Distinguisher

RESV	Resource Reservation Protocol
RID	Router ID
RSVP	Reservation Protocol
SLA	Service Level Agreement
SP	Service Provider
SPF	Shortest Path First
TCP	Transmission Control Protocol
TDP	Time Division Multiplexing
TE	Traffic engineering
TLV	Type Length Value
ToS	Types of Service
TTL	Time-to-Live
UDP	User datagram protocol
VC	Virtual Circuit
VCI	Virtual Channel Identifier
VoIP	Voice over Internet Protocol
VPI	Virtual Path Identifier
VPLS	Virtual private LAN service
VPN	Virtual Private Network
VPWS	Virtual Private Wire Service
VRF	Virtual Routing and Forwarding
WAN	Wide Area Network

I. Introduction

The Internet and its infrastructure experienced an extraordinary exponential growth over the past several years and placed a significant strain on the backbone and the end service providers. Besides the massive increase in the number of users, the number and type of emerging services brought the connection speeds to a multifold increase.

The old paradigm of ordinary data applications required only store and forward capability in a best effort manner, while newer applications like voice, multimedia traffic and real-time e-commerce are pushing towards higher bandwidth and better guarantees, irrespective of the dynamic changes or interruptions in the network [1].

Multiprotocol Label Switching (MPLS) brings the sophistication of a connection-oriented protocol to the connectionless IP world. Based on simple improvements in basic IP routing, MPLS generates performance enhancements and service creation capabilities to the network.

To honor the service level guarantees, the service providers not only have to provide large data pipes (which are also more cost-intensive), but also look for architectures which can provide and guarantee Quality of Service (QoS) and optimal performance with minimal increase in the cost of network resources [2].

With conventional IP packet forwarding, any change in the information that controls the forwarding of packets is communicated to all devices within the routing domain. This change always involves a period of convergence on the forwarding algorithm. A mechanism that can change how a packet is forwarded, without affecting other devices within the network, is certainly desirable.

MPLS is a highly scalable, protocol agnostic, high-performance, data-carrying mechanism used to get packets from one place to another through a series of hops. It belongs to the family of packet-switched networks, which directs and carries data from one network node to the next. The data packets are assigned a label and the content of the label by itself influences packet forwarding decisions, freeing the infrastructure from further examining the packets [3].

The MPLS technology enables service providers to offer additional services to their customers, scale their current offerings, and exercise more control over their growing networks by using its traffic engineering (TE) capabilities.

I.1 Guidelines

I.1.1 Motivation and objectives

MPLS and its subcategories are very well established in nowadays network infrastructures and the clear tendency is to get broader deployment along with a parallel development of sub functionalities aiming to address the explosive growth on network usage and the subsequent complexity of its inherent traffic.

In order to better understand this cascade of technology layers that allows information to travel so efficiently across the globe, it is very important to study MPLS and understand its main functionalities, advantages and drawbacks.

Currently, this subject is already extensively documented, but there is clearly a lack of deployment examples that allow learning simply by doing. Based on this fact, this dissertation presents some use-oriented experiments that describe and demonstrate practical and significant examples of MPLS scenarios.

Since MPLS is a complex framework that is integrated in the core elements of the network infrastructure, it is not viable to test complex scenarios using real equipment due to its very high cost. So, the usage of emerging router emulation tools that, to a certain extent, allow to simulate real life examples, present a convenient alternative.

The emulation of real scenarios involving sophisticated equipment will consume an amount of resources that is proportional to the complexity of the scenarios. Very small configurations of few routers are practicable in a personal computer, but in order to develop more significant experiments, more elaborated solutions are required. Several options exist and will be explored, including the state of the art on computational power, Cloud computing.

The usage of emulation tools in a cloud computing environment can be used to test real network configurations, in a controlled way, before deploying them in real operation conditions. This can have a great impact for network managers/operators, who can use this framework to test new configurations, change routing algorithms and protocols, define new Quality of Service mechanisms and create new network services.

The main objectives of this work are:

1) Propose a series of scenarios that can illustrate some of the most important MPLS functioning mechanisms:

- MPLS Traffic Engineering
- MPLS VPNs
- Ipv6 over MPLS
- Quality-of-service over MPLS

2) Propose a set of practical experiments, involving the manipulation of the proposed network scenarios, which can lead to an active perception of the most important aspects of the previously described topics.

3) Evaluation of the best solutions for an effective deployment of the emulation scenarios.

I.1.2 Structure of the Dissertation

This dissertation is divided into eight functional parts:

1. Introduction - This chapter addresses the introductory aspects inherent to any work. Besides the work description, it defines the main objectives of the study and the structure of the document.
2. MPLS - State of the Art - This chapter gives an overview of the contemporary MPLS technology in order to give the background knowledge that is necessary to understand the practical part emulation scenarios.
3. Emulation - This chapter focuses on diverse aspects of modern emulators that are used to recreate real routers with the purpose of studying and testing network configurations.
4. MPLS-TE - This chapter presents a series of experiments that exemplify practical applications of MPLS Traffic Engineering. The different experiments are based on a network scenario where all intervening routers are totally configured: then, guidelines are provided to highlight the most relevant aspects of the configurations, while instructions are given to test and view what mechanisms are in place, what happens when interfaces or routers are disabled/enabled and what sort of traffic is flowing through the interfaces.
5. MPLS-VPN - This chapter is dedicated to VPN simulation; the presented scenario intends to show the most relevant VPN features and creates a basis for forthcoming MPLS features, as VPNs are the most popular MPLS application. Several cases will be tested, following the previously established method where the network starts with all its routers already configured. A guidance is provided to clarify what is happening in the network scenario, as well as suggestions to interact with the configured network.

6. MPLS-IPv6 - IPv6 traffic will inevitably increase in the next few years; this chapter shows several methods that allow to include IPv6 traffic in current network infrastructures, besides including more realism in the network scenario with the inclusion of a multiple autonomous system structure.
7. MPLS-QoS - This chapter is dedicated to Quality-of-service issues, presenting and demonstrating configurations and strategies that were designed to provide some quality of service guarantees.
8. Conclusions - This chapter gives an overview of the developed work, presents a summary of the main achievements and gives some suggestions for future work.

II. MPLS - State of the Art

II.1 Introduction to MPLS

The fundamental problem of IP routed networks lies on the fact that packets are routed based on the destination IP address. As a result, the forwarding table is derived based only on the routing table that results from shortest path protocols. However, there are cases where the forwarding path that is derived through the routing table may become congested or experience longer delays than expected. Within this paradigm, there is no choice but to forward the traffic down the path defined by the routing mechanisms.

MPLS combines the performance and capabilities of Layer 2 switching with the proven scalability of Layer 3, allowing service providers to meet the challenges of the explosive growth in network utilization while providing the opportunity to differentiate services without sacrificing the existing network infrastructure [4]. It is an improved method for forwarding packets through a network using information contained in labels that are attached to the IP packets. The labels are inserted between the Layer 3 header and the Layer 2 header, in the case of frame-based Layer 2 technologies, and are contained in the Virtual Path Identifier (VPI) and Virtual Channel Identifier (VCI) fields, in the case of cell-based technologies such as ATM [5].

The primary objective of MPLS is to create a flexible networking fabric that provides increased performance and stability. However, the real power of MPLS lies in other applications that were made possible, ranging from traffic engineering (TE) to peer-to-peer Virtual Private Networks (VPN) [6].

II.2 Benefits of MPLS

Label-based switching technology allows MPLS-enabled devices, such as routers and ATM switches, to make forwarding decisions based on the contents of a simple label, rather than performing a complex route lookup based on the destination IP address. This technique brings many benefits to IP-based networks [3,7]:

- Flexible classification of packets and optimization of network resources;

- Label distribution through various protocols, such as Border Gateway Protocol (BGP), Label Distribution Protocol (LDP), Reservation Protocol (RSVP), and Protocol-independent Multicast (PIM);
- The coexistence of different distribution protocols in the same Label Switched Router (LSR).
- Redundancy of numbering and global label allocation, as labels have only local significance.
- The introduction of modular value-added applications such as traffic engineering, quality of service, multicast and VPN.
- Facilitation of the evolution of legacy services via Any Transport over MPLS (AToM) and introduction of Layer 2 VPNs, as the cost of maintaining Frame Relay and ATM infrastructures becomes prohibitive.

Therefore, MPLS provides the predictability of routing performance required to support differentiated services and the capability to offer tight Service Level Agreements (SLAs) associated to differentiated service constructs [8].

As illustrated in figure II.1, MPLS facilitates the integration of multiple services over a common switching platform, contributing to the reduction of operating expenses. MPLS TE can reduce the management burden for IP-based services via the creation of alternative paths and facilitating the deployment of QoS. VPNs can be easily deployed across the Network, allowing the transport of all kinds of traffic, like for example IPv6.

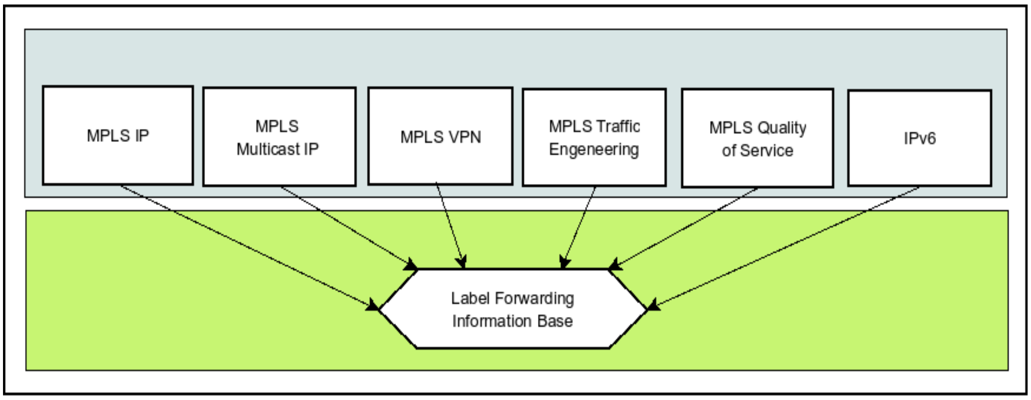


Image II.1: MPLS – Integrated services

II.3 MPLS Architecture

II.3.1 MPLS Operation

MPLS networks use labels to forward packets. At the ingress node, packets are assigned to a particular Forwarding Equivalence Class (FEC), as the packet enters the network [3]. Then the FEC to which the packet is assigned is encoded as a short fixed-length value, known as a label. The packets are then labeled, being ready to be forwarded. At subsequent hops, there is no further analysis of the packet network layer header. The label is used as an index in a table, which specifies the next hop and a new label. The old label is replaced with the new label and the packet is forwarded to its next hop [9].

II.3.2 MPLS Node Architecture

MPLS architecture is separated into two planes [10], as illustrated below in image II.2:

- Forwarding-plane, or data plane, which is used to transport the packets (or label swapping algorithm).
- Control-plane or signaling-plane, programmed into hardware, which is analogous to routing information and is responsible to determine where the packet is to be sent to.

This separation enables applications to be developed and deployed in a scalable and flexible manner.

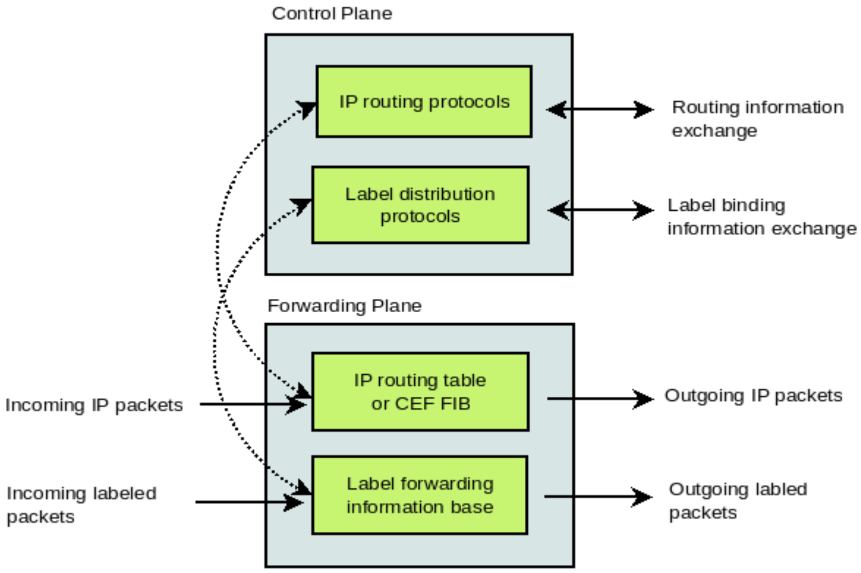

Control Plane
IP routing protocols
Routing information exchange
Label distribution protocols
Label binding information exchange
Forwarding Plane
Incoming IP packets
IP routing table or CEF FIB
Outgoing IP packets
Incoming labeled packets
Label forwarding information base
Outgoing labled packets

Image II.2: MPLS - Node architecture

II.3.2.1 Forwarding Plane

The MPLS forwarding plane is responsible for forwarding packets based on the values contained in the attached labels. The forwarding plane uses the Label Forwarding Information Base (LFIB), maintained by the MPLS node, to forward labeled packets [11]. Therefore, the algorithm uses te information containing the LFIB as well as the information containing the label value [12]. Each MPLS node maintains two relevant tables to perform MPLS forwarding: the Label Information Base (LIB) and the LFIB. The LIB contains all the labels assigned by the local MPLS node and the mappings of these labels to labels received from its MPLS neighbors. The LFIB uses a subset of the labels contained in the LIB for actual packet forwarding [3].

II.3.2.1.1 MPLS Label

The label shown below in Image II.3 is a 32-bit fixed-length identifier used to identify the Forwarding Equivalence Class (FEC), usually of local significance. The label, attached to the packet, represents the FEC to which that packet is assigned to.

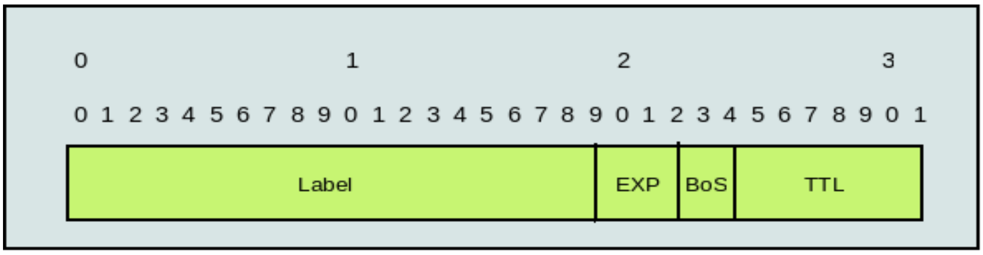


Image II.3: MPLS Label

The fields of the MPLS label have the following functionality [2]:

- Label field (20 bits): Carries the actual value of the MPLS label.
- Class of Service (CoS) field (3 bits): Affects the queuing and discard algorithms applied to the packet as it is transmitted through the network.
- Stack field (1 bit): Supports a hierarchical label stack.
- TTL field (8 bits): Provides conventional IP TTL functionality.

II.3.2.1.2 Label Stack

MPLS allows label stacking, where more than one label header can be attached to a single IP packet [3]. In packet-based MPLS, the top of the stack appears right after the link layer header, and the bottom of the label stack appears right before the network layer header. Packet forwarding is performed by using the label values of the label located on the top of the stack [13].

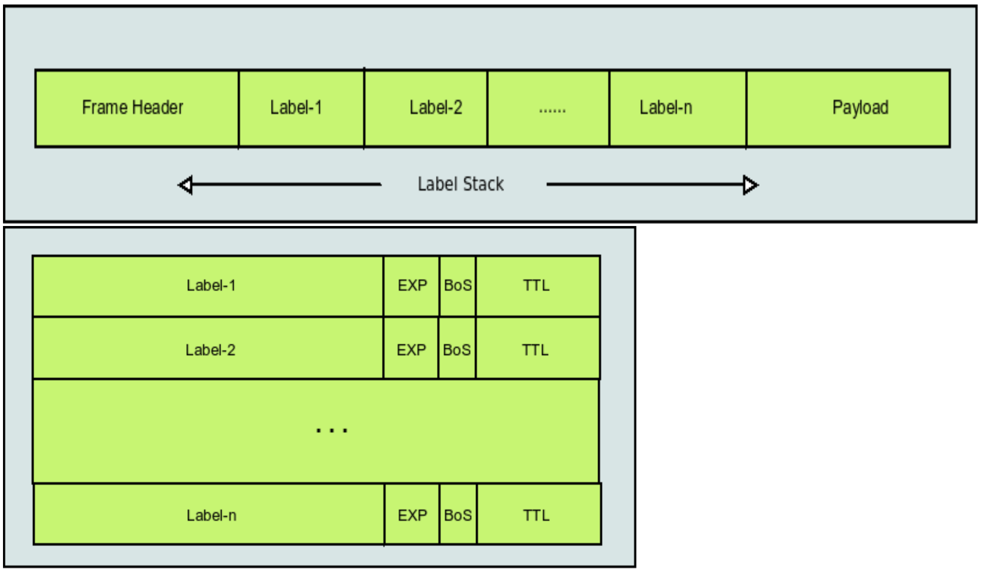


Image II.4: MPLS - Label Stack

II.3.2.1.3 Label Forwarding Algorithm

Label switches use a forwarding algorithm based on label swapping. There are two tables, LIB and LFIB, where values are kept in order to make decisions. The LIB table holds prefixes and labels from all the neighbors and the LFIB holds incoming and outgoing labels for a prefix for the best path(s) [3]. After an incoming label match is found, the MPLS node replaces the label in the packet with the outgoing label from the sub-entry and sends the packet over the specified outgoing interface to the next hop specified by the sub-entry. If the sub-entry specifies an outgoing queue, the MPLS node places the packet in the specified queue. The physical interface where the packet arrived to select a particular LFIB, which is used to forward the packet, is used if the MPLS node maintains multiple LFIBs for each one of its interfaces [3].

II.3.2.2 Control Plane

The MPLS control plane is responsible for populating and maintaining the LFIB. All MPLS nodes must run an IP routing protocol to exchange IP routing information with all other MPLS nodes in the network. Link-state routing protocols, such as OSPF and IS-IS, are frequently chosen protocols, due to the fact that they provide each MPLS node with a view of the entire network.

The LFIB is built with the labels exchanged with adjacent MPLS nodes [3]. MPLS uses a forwarding paradigm that can be combined with a range of different control modules. Every control module has the responsibility to assign and distribute a set of labels and to maintain other relevant control information [14]. IGPs are used to define reachability, binding and mapping between FEC and next-hop addresses.

II.3.3 MPLS Elements

II.3.3.1 Label-Switched Router (LSR)

The LSR is a device that implements the MPLS control and forwarding components. The LSR forwards a packet depending on the label that is enclosed in the packet. It can also forward native Layer 3 packets [15].

Edge LSRs are located at the Point of Presence (POP) boundaries of an MPLS network and apply labels to packets [3]. They also execute a label disposition and can remove labels at the egress point of the MPLS domain [16]. Edge LSRs can also perform conventional IP forwarding.

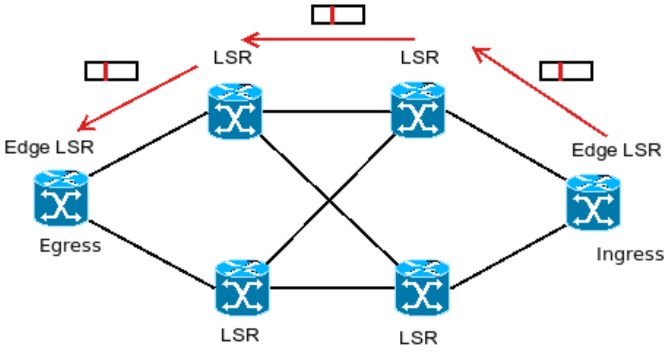


Image II.5: MPLS - Label Switched Routers

II.3.3.2 Label-Switched Path (LSP)

The LSP is a configured connection between two LSRs. It is a specific traffic path through a MPLS network [3]. LSPs are provisioned using Label Distribution Protocol (LDP) or the previous Cisco Tag Switching Protocol (TDP), Resource Reservation Protocol (RSVP), or extensions to routing protocols such as Multiprotocol BGP.

II.3.3.2.1 LSP Establishment

LSP establishment can be performed as:

- Independent control
- Ordered control

The independent method gives faster convergence and establishment of LSPs, due to the fact that the LSR is able to establish and advertise label bindings at any time, without delay or without waiting for messages to be distributed in order from one side of the network to the other [17]. LSP establishment immediately follows the convergence of the routing protocols. In the ordered control method, bindings are transmitted across the network before the LSP is created. The ordered control method provides better loop prevention capabilities [18].

II.3.4 Label Distribution Protocol (LDP)

The LDP is used in combination with standard network layer routing protocols to distribute label-binding information between LSR devices [5,19]. LDP lets an LSR distribute labels to its LDP peers using the Transmission Control Protocol (TCP), which results in reliable delivery of LDP information with flow control and congestion-handling mechanisms.

When an LSR assigns a label to a FEC, it needs to let its relevant peers know of this label and its meaning [3]. LDP is used for this purpose. A set of labels from the ingress LSR to the egress LSR in a MPLS domain defines a LSP. Labels are maps of network layer routing to the data link layer switched paths. LDP helps in establishing an LSP by using a set of procedures to distribute the labels among the LSR peers. LDP provides a LSR discovery mechanism to let LSR peers locate each other and establish communication [20].

II.3.5 Loop Detection and Prevention in MPLS

The LDP relies on information collected by the Layer 3 routing protocols and, therefore, is approachable to routing loops unless the Layer 3 protocol itself can avoid loop prevention [3]. There are two ways for MPLS-level traffic to be affected if routing loops are not prevented:

- *LSP control packet looping* - The packets used for LSP establishment are forwarded in an endless routing loop, and the LSP is never properly established end-to-end. This continues until the routing loop breaks [21].
- *MPLS data packet looping* - Data packets injected into an established looping LSP continue to be label-switched in the LSP until the routing loop breaks [3].

II.3.5.1 Loop Control in MPLS

II.3.5.1.1 Loop Survival

The LSPs are allowed to form loops, but looping packets are not allowed to impact the transmission of non-looping packets [3]. MPLS nodes that have the ability to perform TTL decrement for LSPs have this capability [20].

II.3.5.1.2 Loop Detection

It allows an LSP to form a loop, but it can detect and disconnect the loop in a short period of time [3]. Examples of protocols supporting loop detection are LDP and Cisco TDP.

II.3.5.1.3 Loop Prevention

Loop Prevention prevents looping paths. With this method, two LSPs are classified [21,22] as:

- *Nonstate merging LSPs:* A single incoming link state exists for each outgoing link state.
- *State Merging LSPs:* For each outgoing link state, multiple incoming link states exist.

II.4 MPLS Traffic Engineering

IP networks occasionally manifest poor efficiency, because they are not equipped with dynamic redundancy and have no means to evaluate the characteristics of the network capacity constraints or type of flowing traffic when making routing decisions.

In a MPLS traffic-engineered network, LSPs can be shifted from a congested or broken path to an alternative path by means of built in tunnels, as shown below in image II.6, resulting in efficiency improvements over the traditional operational methods of IP networks [23]. MPLS TE allows the definition of explicit paths across the network and steering traffic over these paths. Redundant pre-computed explicit paths can be configured, thereby providing a fall-back mechanism. TE can also perform unequal cost load balancing across tunnels [24].

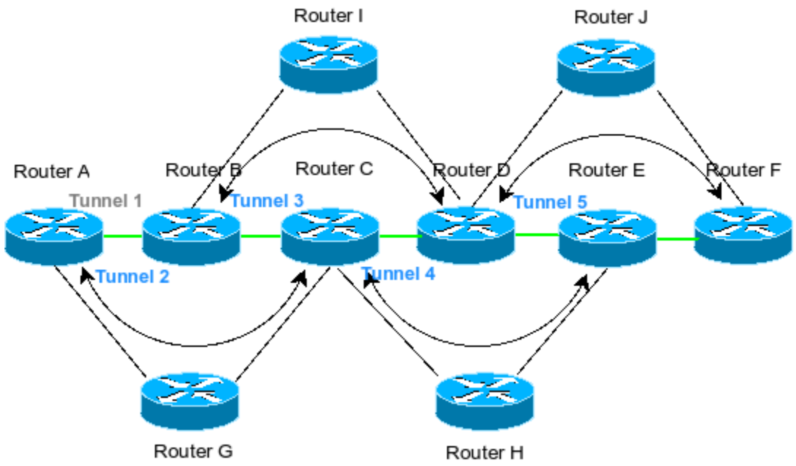


Image II.6: MPLS - Traffic Engineering

The following are some advantages of MPLS TE:

- MPLS-TE features are integrated into Layer 3, optimizing IP traffic routing given the constraints imposed by backbone capacity and topology.
- IP traffic can be routed across the network based on the traffic type requirements (VoIP) and the availability of network bandwidth resources.
- Constraint-based routing is applied to traffic, assuring that the chosen path for the traffic flow is the shortest path that meets the resource requirements or constraints in terms of bandwidth requirements, media requirements, and traffic flow priority.
- Link and node failures that change the backbone topology are dynamically salvaged by adapting to a new set of constraints; these new routes are either the product of automatic mechanisms or off-line pre-computed paths.
- The use of unequal-cost load sharing allows different paths from the set of paths learned from the IGP.
- Link bandwidth and size of the traffic flow are taken into account when determining explicit routes across the backbone.
- The need to manually configure the network devices in order to set up explicit routes is replaced, relying on the MPLS TE functionality to understand the backbone topology and the automated signaling process.

Three basic applications for MPLS TE are:

1. <u>Network utilization optimization:</u>

   This feature is accomplished by fully meshing TE-LSPs between a given set of routers, followed by sizing LSPs regarding the bandwidth that they will have to accommodate and allowing LSPs to find the best path that meets their bandwidth demands.

2. Unexpected congestion management:

   In normal circumstances, traffic forwarding decisions are made using regular IGP protocols. Once congestion is detected, TE-LSPs forwarding mechanisms take the responsability of addressing the congestion problem. This approach has the advantage of keeping most of the network on IGP routing only.

3. Handling link and node failures:

   MPLS TE has a component called FRR that drastically minimize packet loss upon a link or node (router) failure.

II.4.1 Prerequisites

In MPLS TE, the following prerequisites have to be enabled in the network [25]:

- A release of the Cisco IOS Software that supports MPLS TE;
- The Cisco Expedited Forwarding (CEF) feature;
- A link-state routing protocol (OSPF or IS-IS) as the IGP routing protocol;
- TE globally enabled on the network routers;
- A loopback interface to use as MPLS TE RID;
- Basic TE tunnel configuration.

II.4.2 Information Distribution

IGP protocols, like OSPF or IS-IS, are used together with MPLS-TE to distribute information about available resources in the network. Three major pieces of information are distributed [25]:

- Available bandwidth information per interface, broken out by priority to allow some tunnels to preempt others;
- Attribute flags per interface;
- Administrative weight per interface.

Each one of these three pieces is advertised on a per-link basis for all the links that are involved in MPLS TE.

Regarding the process of setting up and tearing down tunnels across interfaces, the amount of available bandwidth on the devices varies in accordance with the respective reservations across the Network. This tunnel infrastructure contributes with its own weight bandwidth, reducing accordingly the amount of available bandwidth; tearing down tunnels in a particular interface, releasing the total amount of available resources as well the amount of available bandwidth [11]. Some large MPLS TE networks have a high number of tunnels, whose flooding traffic can greatly impact the network performance due to the significant network bandwidth consumption and the high usage levels of the routers' CPU resources [25].

Adding to this side effect, it is fundamental to assure that the topology information advertised by the router is reasonably up to date. If all the bandwidth on a particular link is reserved and this fact is not advertised to the rest of the network, together with the fact that the network is out of sync with the current needs, can lead to setup failures and other suboptimal effects.

In an effort to mitigate these adversities, there are three rules of flooding thresholds [25]:

1. Flood significant changes immediately.
2. Flood insignificant changes periodically, but more often than the IGP refresh interval.
3. If a change that has not yet been flooded is known to cause an error, flood immediately.

II.4.3 Path Calculation and Setup

Several factors are significant in the process of calculating paths for TE tunnels, resulting in the shortest possible path from all paths whose link characteristics match the characteristics needed for the TE tunnel:

- Path setup option;
- Setup and holding priority;
- Attribute flags and affinity bits;
- Re-optimization.

II.4.3.1 Constrained Shortest Path First (CSPF)

This shortest path algorithm extension method entails itself with the pruning of those links that violate a given set of constraints, followed by its own run of the shortest path algorithm. A constraint can be minimum bandwidth required per link, end-to-end delay, maximum number of links traversed, include/exclude nodes [26].

II.4.3.2 Resource Reservation Protocol (RSVP)

RSVP is a signaling protocol for TE tunnels, being responsible for creating the structure that allows labels to propagate hop by hop and to ensure that link admissions happen at the interfaces of the

LSRs that the TE tunnel crosses [21].

After a path calculation, performed with the CSPF, this path needs to be signaled across the network, which is accomplished using RSVP, along with RSVP extensions for MPLS TE [25]. Image II.7 illustrates the process of a tunnel establishment: first, path discovery occurs with CSPF, then RSVP performs back and forth signaling and, finally, the TE tunnel is established.

The RSVP task is to signal and maintain resource reservations across the network, which is done at the control-plane layer, whereas the forwarding-plane is not present in this framework in the traffic policing. Three basic functions are relevant to the signaling process:

- Path setup and maintenance;
- Path tear-down;
- Error signaling.

Like any dynamic environment, RSVP needs to periodically refresh its reservations in the network by re-signaling them; PATH and RESV messages are used to carry out this task [27]. The TE head end router sends the PATH messages to the tail end router, whereas the RESV messages take the exact but opposite path back to the head end router [25], as it is explicit in the above image. The best path is computed at the head end router of a TE tunnel using the TE database, considering bandwidth and other constraints.

Alternatively, the path is defined by an explicit path option configured by the user on the tunnel interface [11]. In either case, the head end router knows the exact path that the TE tunnel should take. Each hop (LSR) that the TE tunnel should cross is put into an Explicit Route Object (ERO), which is basically an ordered list of interface IP addresses, with one IP address per LSR. The PATH message is sent from the head end router to the next-hop router. This next-hop router removes his own IP address from the ERO, sees which is the next IP address, and sends the PATH message to the next hop [28].

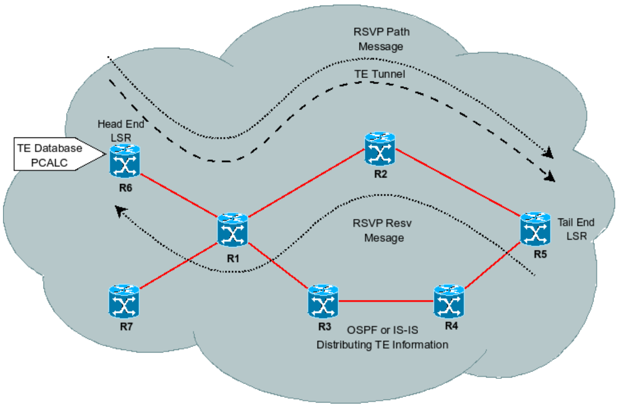
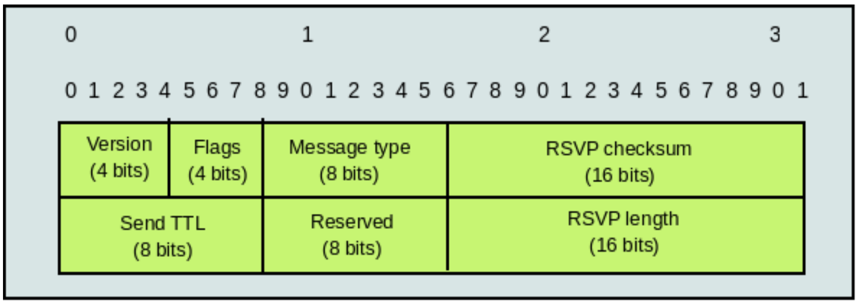


Image II.7: Resource Reservation Protocol (RSVP)

II.4.3.2.1 RSVP Packets

Every RSVP message is composed by a common header, followed by one or more objects. The number of objects in a message depends on exactly what the message is trying to accomplish. A RSVP message example is shown below in image II.8.



0 1 2 3

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

Version (4 bits)	Flags (4 bits)	Message type (8 bits)	RSVP checksum (16 bits)
Send TTL (8 bits)		Reserved (8 bits)	RSVP length (16 bits)

Image II.8: RSVP Packets

II.4.3.2.2 RSVP and Labels

Besides signaling the path for the TE tunnel, RSVP is also responsible for carrying the MPLS label, so that the packets can be label-switched along the path of the TE tunnel [11]. Image II.9 illustrates this process: the PATH label request message carries a Label Request object and, when the tail end router receives this object, it assigns a label to this TE tunnel LSP and advertises it to the upstream router in a Label object in the RESV message. This label is the incoming label in the LFIB of the tail end router. The upstream router receives the label from the tail end router and puts this label as the outgoing label in the LFIB for this TE tunnel LSP. The router assigns a label from the global label table to this TE tunnel LSP and sends it in a Label object in the RESV message to its upstream router. This label becomes the incoming label in the LFIB for this TE tunnel LSP. The process continues like this until the RESV message reaches the head end router of the TE tunnel LSP.

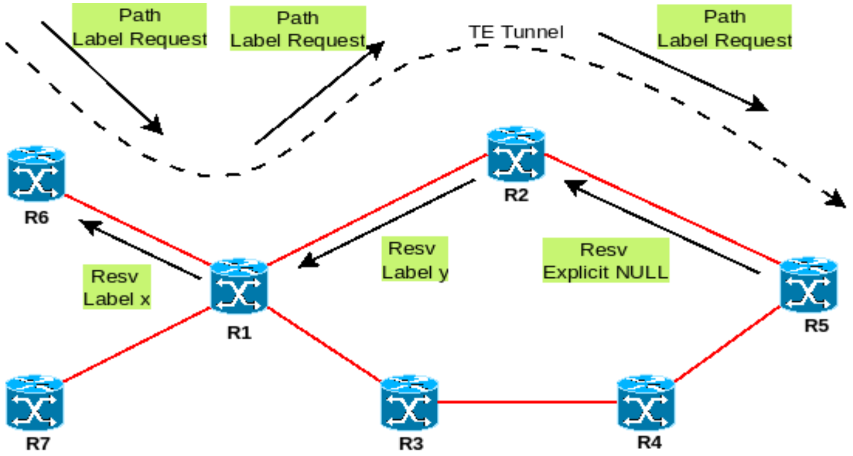


Image II.9: RSVP and Labels

II.4.4 Forwarding Traffic Down Tunnels

There are three methods to forward traffic down a tunnel interface [25]:

- Static routes
- Policy routing
- Autoroute

II.4.4.1 Static Routes

Static routes over TE tunnels work in a similar way to static routes that point to any point-to-point interface, allowing any traffic to be routed through the interface.

II.4.4.2 Policy-Based Routing (PBR)

PBR uses a configured policy on the incoming interface to send traffic to a specific next hop. The difference from the usual methods is that routing is solely based on the destination IP address [28].

II.4.4.3 Autoroute

Autoroute is present on the tunnel interface of the MPLS TE tunnel head end router, so that the LSR can insert IP destinations, using the *announce* command, configured into the routing table with the TE tunnel as the next hop or outgoing interface. In essence, autoroute installs the routes announced by the tail-end router and its downstream routers into the routing table (forwarding table) of the head-end router as directly reachable through the tunnel [29]. The following image, Image II.10, represents an example where different tunnels (1 and 2) are selected using autoroute.

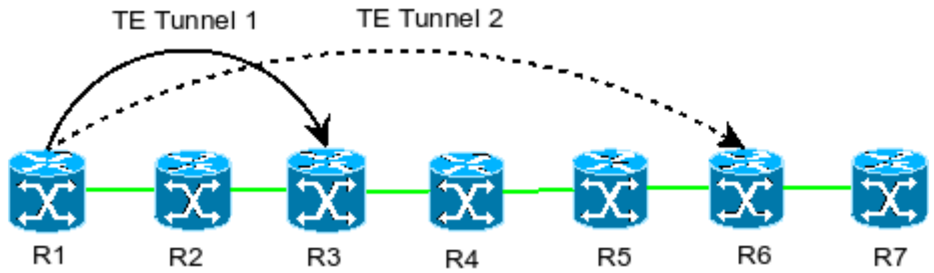


Image II.10: Policy-Based Routing (PBR) - autoroute

II.4.5 Protection and Restoration

Networks are hazard prone environments where many dysfunctions can occur. From a router's perspective, there are two kinds of failures in the network: link failures and node failures.

MPLS TE, and its ability to steer traffic away from the IGP derived shortest path, helps mitigate packet loss and latencies associated with link or node failures in the network [21]. This ability is known as MPLS TE Protection.

Protection is the set of procedures that, when applied to selected resources, aim to minimize traffic loss upon failure. Protected resources could either be viewed as physical resources (link or nodes) or logical resources, like paths [25]. Protection can be divided into:

- Path protection

◦ Local protection, which can be broken into two types: Link protection and Node protection.

II.4.5.1 Path protection

Path protection, in essence, can be described as the establishment of additional alternative LSPs in parallel with an existing primary LSP, where the additional resource acting as backup is used only in case of failure. This LSP is sometimes called the backup, secondary, or standby LSP [25].

Path protection is simple in concept, where backup LSPs are built along paths that are as diverse as possible from the LSP they are protecting. This ensures that a failure along the path of the primary LSP does not also affect the backup LSP. Each primary LSP is backed up by a standby LSP. Both the primary and backup LSPs are configured at the headend. Both are signaled ahead of time in the control plane. Image II.11 illustrates this concept, considering two available paths between R1 and R3, the primary and the backup path:

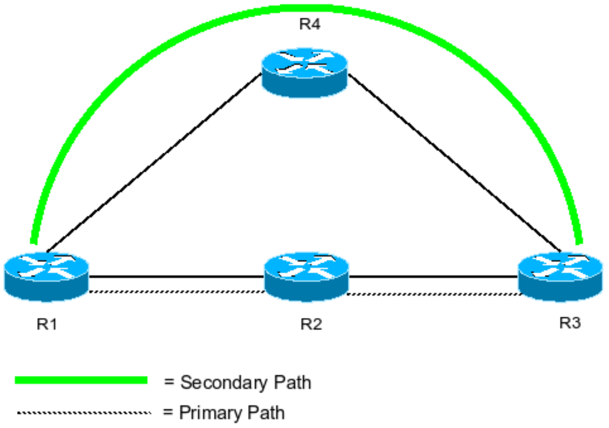


Image II.11: Path protection

II.4.5.2 Local protection or Fast Reroute (FRR)

Local protection is the term used when the backup or protection tunnel is built to cover only a segment of the primary LSP. Local protection, like path protection, requires the backup LSP to be previously signaled. In local protection, the backup LSP is routed around a failed link (in link protection) or node (in node protection), and primary LSPs that would have gone through that failed link or node are instead encapsulated in the backup LSP [30].

- Link protection

Backup tunnels that bypass only a single link of the LSP's path provide link protection. They protect LSPs if a link along their path fails by rerouting the LSP's traffic to the next hop (bypassing the failed link). These are referred to as Next-Hop (NHOP) backup tunnels because they terminate at the LSP's next hop beyond the point of failure. Image II.12 highlights the case where the link between R2 and R3 is protected via NHOP tunnel.

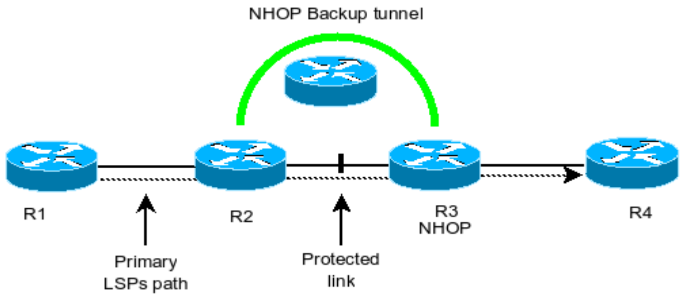


Image II.12: Link protection

- Node protection

Backup tunnels that bypass next-hop nodes along LSP paths are called Next-Next-Hop (NNHOP) backup tunnels because they terminate at the node following the next-hop node of the LSPs, thereby bypassing the next-hop node. They protect LSPs by enabling the node located upstream to a link or node failure to reroute the LSPs and their traffic around the failure to the next-hop node. NNHOP backup tunnels also provide protection from link failures because they bypass the failed link and the node. In the following example, Image II.13, the backup tunnel allows the bypass of an entire faulty segment (in this case composed by R3 and its respective links), besides exclusively protecting links. Of course that in the real world this segment failure could account for the blackout of entire network segments.

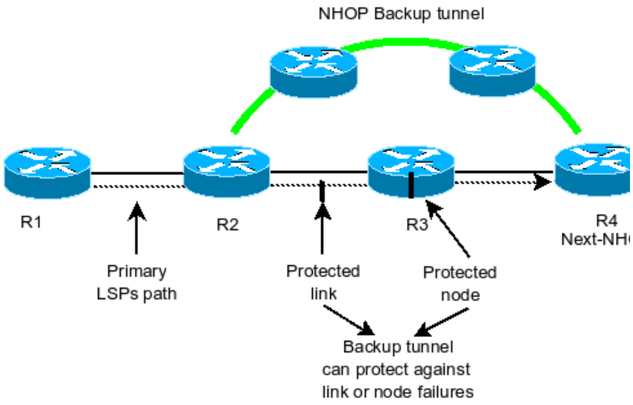


Image II.13: Node protection

II.4.5.3 Benefits of MPLS TE Protection

The most important benefits of MPLS TE protection are:

- Delivery of Packets During a Failure;
- Multiple Backup Tunnels Protecting the Same Interface [31];
- Redundancy: If one backup tunnel is down, other backup tunnels protect LSPs [31];
- Increased backup capacity: If the protected interface is a high-capacity link and there is no single backup path with equal capacity, multiple backup tunnels can protect that high-capacity link [31];
- Scalability: A backup tunnel can protect multiple LSPs and multiple interfaces. This is called many-to-one (N:1) protection. N:1 protection has significant scalability advantages over one-to-one (1:1) protection, where a separate backup tunnel must be used for each LSP that needs protection [31].

II.5 MPLS Virtual Private Networks (VPN)

VPNs based on MPLS combine the benefits of the overlay VPN model, such as isolation and security, with the benefits of the peer-to-peer VPN model, such as simplified routing, easier provisioning, and better scalability [10].

MPLS based VPNs use a combination of connectionless VPNs between the customers and service providers with connection-oriented VPNs in the network core, including the following features [31]:

- The link between the Customer Edge (CE) routers and the Provider Edge (PE) routers is connectionless, requiring no additional configuration on the CE devices.

- PE routers use a specific forwarding paradigm where a distinct IP routing and forwarding table (VRF) is used for each customer.
- The customer address space is extended to 64-bit route distinguisher turning the Universal 32-bit IP addresses globally unique within the service providers' backbone, resulting in 96-bit addresses called VPNv4 addresses.
- Routing between the PE routers within the Internet Service Provider's (ISP) global VPN network is accomplished by a modified version of the BGP, with MPLS extensions, called MP-BGP.
- Virtual circuits (LSPs) are used by the PE routers to transport the customer's datagrams between PE routers. Additional MPLS labels are attached in front of the customer's IP datagrams to ensure proper forwarding from ingress PE routers towards the destination CE router.
- The establishment of the LSPs between all PE routers is done automatically based on the IP topology of the P-network.
- Mapping between the customer destination addresses occurs automatically by creating LSPs leading towards the egress PE routers based on the BGP next-hops.

Image II.14 illustrates a simplification of what a real world VPN scenario would look like, where connections between sites coexist side by side sharing edge provider's routers. The core is a generic MPLS network where the VPN decisions are made at the borders and the customers address spaces may overlap, for example, between VPN1 and VPN2 at site1.

In a typical MPLS-VPN network, the CE routers and PE routers exchange the customer routes using any suitable IP routing protocol. These routes are inserted into VRFs on the PE routers.

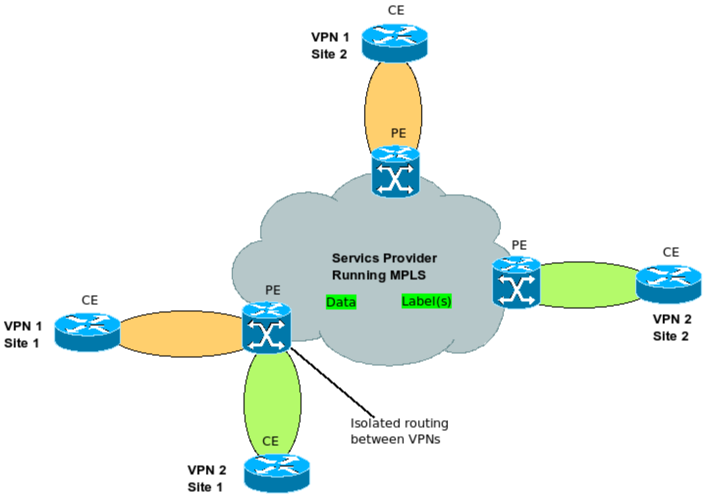


Image II.14: MPLS VPNs

II.5.1 VPN Routing and Forwarding Tables

According to networking standards, customer networks will likely be configured with overlapping addresses due to the usage of private IP addresses. This represents one of the major challenges for the successful deployment of peer-to-peer VPN implementations. The solution found in MPLS/VPN technology to address this issue relies on the following: each VPN has its own routing and forwarding table in the router, so any customer or site that belongs to that VPN has only access to the set of routes contained within that table [32]. PE-routers in MPLS/VPN deployed networks contain a number of per VPN routing tables and a global routing table that is used to reach other routers in the providers network, as well as external globally reachable destinations [32]. Effectively, a number of virtual routers are created in a single physical router; by joining this apparatus, more structures are associated with each virtual router than just the virtual IP routing table [33]:

- A forwarding table derived from the routing table that based on CEF technology.
- A set of virtual interfaces where the traffic will be forwarded to.
- A set of rules responsible for controlling the import and export of routes from and into the VPN routing table; these rules make overlapping VPNs coexistence possible.
- A set of routing protocols/peers that inject information into the VPN routing table.
- Router variables associated with the routing protocol that is used to populate the VPN routing table.

Image II.15 is a simplified sketch of what happens on routers regarding routing and forwarding packets in a MPLS-VPN network.

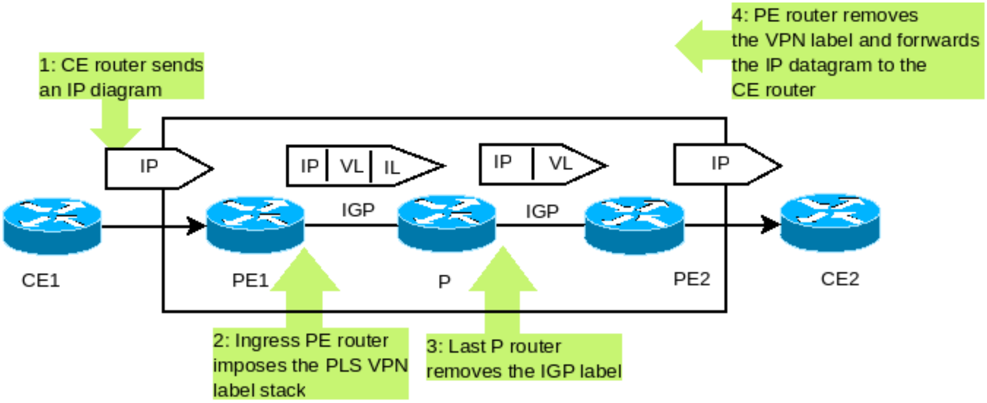

4: PE router removes the VPN label and forwards the IP datagram to the CE router
1: CE router sends an IP diagram
IP
IP | VL | IL
IP | VL
IP
IGP
IGP
CE1
PE1
P
PE2
CE2
2: Ingress PE router imposes the PLS VPN label stack
3: Last P router removes the IGP label

Image II.15: VPN Routing and Forwarding

When customer routes are placed into Virtual Routing and Forwarding (VRF), the PE routers allocate a separate MPLS label that will be needed for VPN data forwarding to each customer route. The customer routes and associated MPLS labels are transported across the P-network using the multiprotocol BGP. Additional BGP attributes are used to control the exchange of routes between VRFs [32].

II.5.2 Virtual Private Networks overlapping

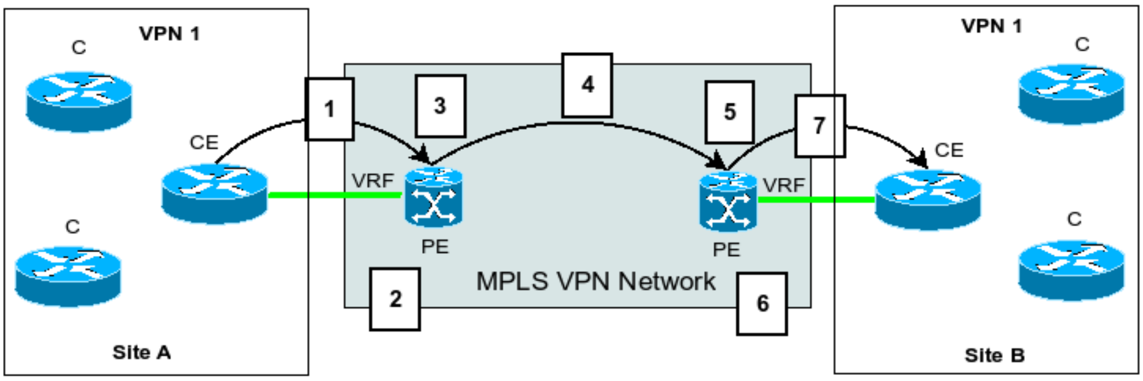
The MPLS/VPN architecture includes the concept of site, meaning that a VPN is composed by one or multiple sites. A VPN is essentially a collection of sites sharing common routing information, which means that a site may belong to more than one VPN if it holds routes from separate VPNs [32]. This provides the ability to build intranets and extranets.

II.5.3 Route Targets

Every VPN route is tagged with one or more route targets when it is exported from a VRF. A set of route targets with a VRF can also be associated, and all routes tagged with at least one of those route targets will be inserted into the VRF. The route target is the closest approximation to a VPN identifier in the MPLS/VPN architecture [34].

II.5.4 VPN Routing Information propagation in the Provider Network

Potentially, a large number of routes can be transported in the provider network. Therefore, BGP is the ideal candidate for routing protocol as it is proven to be stable and capable to handle such large number of involved entities [32]. Customer VPN routes are made unique by adding the Route Distinguisher (RD) to each IPv4 route, turning them into VPNv4 routes, so that all customer routes can safely be transported across the MPLS VPN network [11]. The process of propagating routes and their respective information is illustrated in the following Image II.16, where the different steps are clarified.



1: IGP or eBGP advertises IPv4 route
2: PPv4 route is inserted into VRF routing table
3: IPv4 route is redistributed into MP-BGP. RD is added to IPv4 route to make it a VPNv4 route. RTa are added.
4: iBGP advertises VPNv4 route with MPLS lable and RTs
5: RTs indicates to which VRF the route is imported. RD is removed from VPNv4 route.
6: IPv4 route is inserted into VRF routing table.
7: IGP or eGBP advertises IPv4 route.

Image II.16: Propagation of VPN Routing Information in the Provider Network

II.5.5 VPN Packet Forwarding at the Core

The packets cannot be forwarded as pure IP packets between sites. P routers cannot forward them because they do not have the VRF information from each site. In MPLS this problem is solved by labeling the packets. So, P routers must have the correct forwarding information to forward the packets. The most common way is to configure Label Distribution Protocol (LDP) between all P and PE routers so that all IP traffic is label-switched between them [10]. The packet enters the PE router on the VRF interface as an IPv4 packet. It is forwarded throughout the MPLS VPN network with two labels. P routers forward the packet by looking at the top label, which is swapped at each P router. The labels are removed at the egress PE router and the packet is forwarded as an IPv4 packet onto the VRF interface toward the CE router [12]. The correct CE router is found by looking at the VPN label. Packet forwarding is shown below in Image II.17.

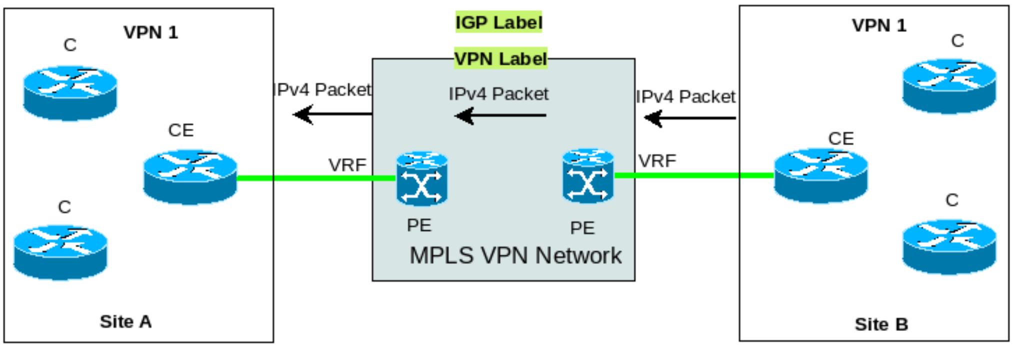


Image II.17: VPN packet forwarding

II.6 IPv6 over MPLS

IPv6 is the next generation of IP. It has a larger address space and more features, and thus is the direction where many service providers are moving to. Every device connected to the Internet needs an IP address but, with the rapid success of the Internet, assigning public addresses to all devices is not an option due to exiguous number of IPv4 addresses. IETF has standardized a new address family called IPv6, with 128 bits of address space, a much larger address space than the 32 bits of IPv4. In addition to addressing, IPv6 also provides improved security and data integrity, auto configuration, multicasting, and anycasting capabilities.

The transport of IPv6 datagrams across an MPLS-enabled IPv4 backbone, known as IPv6 provider edge router (or 6PE), provides a scalable solution to the IPv6 early-deployment problem [35]. It has the following characteristics [36]:

- The IPv6 protocol is deployed only on selected PE routers.
- The PE routers use Multiprotocol BGP sessions to exchange IPv6 routes across the backbone.
- MPLS labels are assigned to IPv6 routes by the PE routers and exchanged directly between the PE routers.
- The IPv6 datagrams are transported across the MPLS backbone by using a two-level MPLS label stack. The first label in the label stack is the LDP-assigned label of the egress PE router. The second label in the label stack is the PE-assigned IPv6 label.

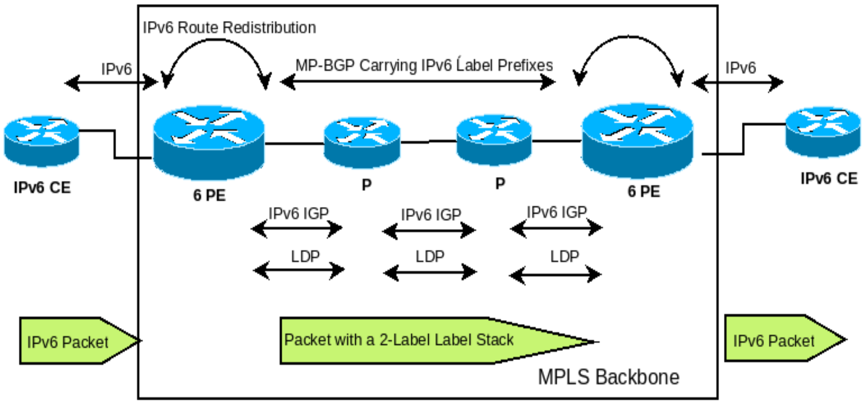


Image II.18: IPv6 over MPLS

II.6.1 IPv6 PE

Tunneling of IPv6 packets can be done using the underlying IPv4, being transported across the MPLS network. Another form of tunneling is using the label as an encapsulation instead of another IPv4 header. This method of transporting IPv6 packets across the MPLS network by using the label as a mux/demux field and the tunneling encapsulation is called IPv6 PE (6PE) [8].

In this environment, each PE router is a dual network stack device containing IPv6 and IPv4 stacks. The edge router interacts with the customer IPv6 and IPv4 routers separately. The CE devices do regular PE-CE routing by using either dynamic routing protocols or static routing. In addition to IPv4 routes, the PEs peer with each other directly or by using route reflectors to exchange IPv6 routes. The core routers learn only IPv4 addresses that are advertised in the IGP. For the core routers IPv6 is not visible, so they cannot reach IPv6 addresses. The labeled IPv6 packets are forwarded on the IGP-based, label-switched paths (IGP LSP) [8].

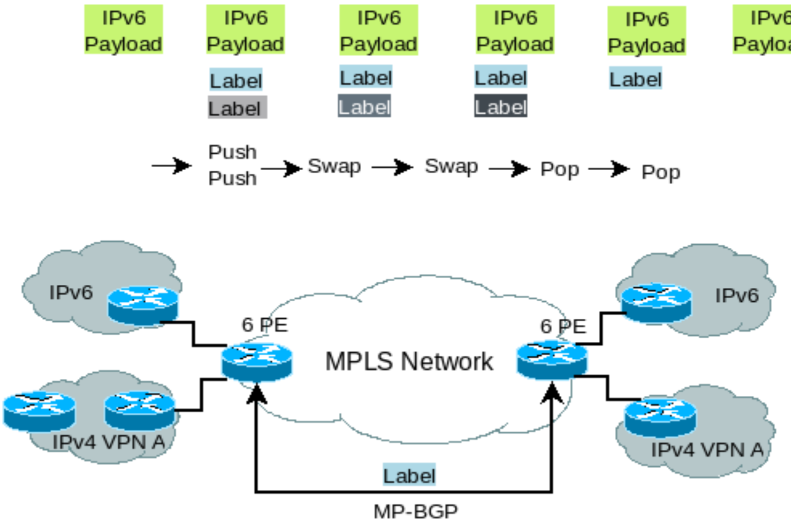


Image II.19: IPv6 PE

II.6.1.1 Multiprotocol BGP Extensions

IPv6 address information is distributed between PEs using BGP extensions that carry IPv6 addresses using the address family identifier to distribute the IPv6 prefixes within BGP [37]. A label, which is associated with that IPv6 network prefix, is also advertised as part of BGP extended communities. This is similar to the IPv4 VPN extensions. Then, PEs exchange this IPv6 prefix and label information using the iBGP session between them, as it is also done with the L3VPN. Just as the PEs advertise the IPv4 reachability, they also advertise IPv6 reachability with the ability to resolve the IPv6 routes to an IPv4 next hop [8].

II.6.1.2 Packet Path

The PE routers are dual-stacked in order to support IPv6 and IPv4 on the same port or on different ports. When IPv6 packets come from the CEs, the PE looks for the IPv6 forwarding table and finds the IPv4 BGP next hop [8]. Afterwards, it implements the IPv6 label assigned to the IPv6 prefix by the egress PE, which then imposes the IGP or tunnel label to get to the egress PE (IPv4 BGP next hop) using the Packet Switched Network (PSN) tunnel or a label-switched path [8]. The packet is switched through the network core using the top label. When the packet arrives at the egress PE, the egress PE detects the label and forwards the IPv6 packet to the destination interface that corresponds to that prefix. The difference to the IPv4 VPN case is that all IPv6 addresses are in one address space.

There is no difference in packet handling compared to the VPNv4 case, with the exception that all IPv6 CEs/addresses are in one domain and can be reached via IPv6 unless route filtering is applied [28]. The network core is not able to understand IPv6 frames and is running standard IPv4 with IGP routing protocols (e.g. OSPF and IS-IS). The P routers are only label-switching packets based on the top label or PSN tunnel label, without knowing the IPv6 label [8].

II.6.2 IPv6 VPNs (6VPE)

Service providers may need to keep the traffic of one VPN separated from the traffic of another in the IPv6 space, much like the IPv4 VPNs. There are two ways to achieve traffic separation between IPv6 devices: one approach is to perform route tagging of IPv6 routes via the BGP communities and then filter the IPv6 routes based on those tags; the second method is to separate the IPv6 routes between VRFs [38].

II.6.2.1 Route Tagging

This method allows the PE to assign a tag to any route as part of a BGP attribute, being this tag called a community. PE can then filter the routes to exclude those that do not belong to its community of interest by assigning route tags, upon the process of importing the routes [8]. Routes are, in this way, prevented from being populated in the PEs when they are not required. This is a simple model where all IPv6 traffic belonging to one VPN is tagged with the VPN name. When PEs receive the routes, they filter them based on policies. By tagging routes and selectively importing/filtering the routes, extranets can also be built and, considering the current network size and elaboration, efficient filtering techniques may be needed [39].

II.6.2.2 Virtual Routing and Forwarding with IPv6

As mentioned before, another way to propagate IPv6 VPNs is to use similar VRF structures to separate one IPv6 route per VRF, in a separation of routes similar to the one used in the IPv4 model. CEs are connected to the PEs and are placed in a VRF, where the PEs have a separate routing and forwarding table per VRF/VPN, and IPv6 routes from the CE are populated in these VRFs [38]. Peering between PEs is achieved by the exchange of IPv6 routes in iBGP using the IPv6 address family extension of the MP-BGP, a process that is similar to the well known VPNv4 [8]. Labels are then distributed in the same way as the IPv4 prefixes, as well as the functionality of separating one IPv6 address from another using route distinguisher and route target. Route targets can be imported or exported in the same manner as IPv4 routes to create intranets and extranets [40].

II.6.3 Packet Path

The Packet Path across the network is illustrated below: when IPv6 packets come into the PE from the CE of a VPN, the PE looks for the IPv6 route in the VRF and finds the egress IPv4 BGP next hop and VPN label associated with that IPv6 prefix. It imposes the VPN label and then forwards it onto the PSN tunnel or onto a label-switched path towards the egress PE [8].

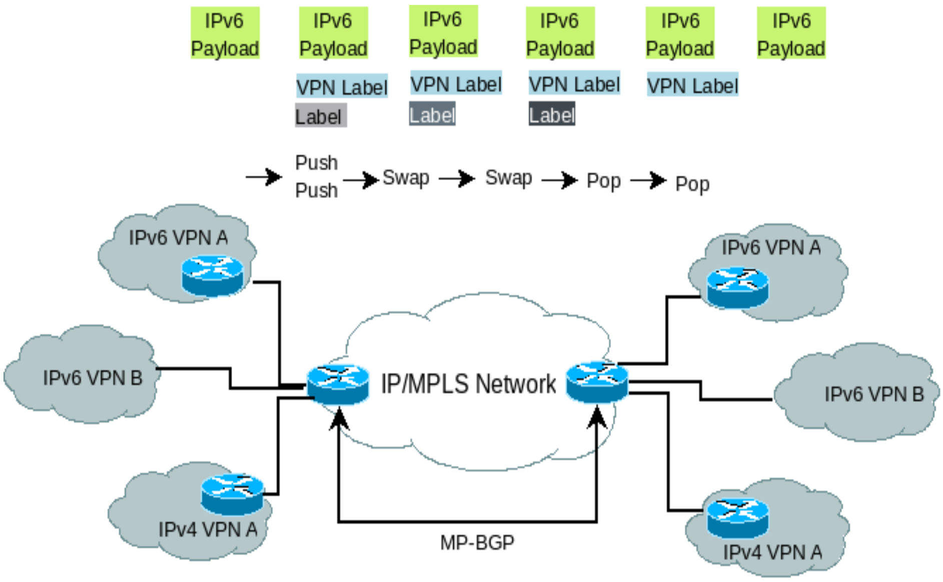


Image II.20: Packet Path for a VPN based scenario

II.6.4 The Coexistence of IPv4 and IPv6

IPv4 L3VPNs and IPv6 L3VPNs can run together on the same PE and sharing the same VRF table, used to store IPv6 and IPv4 addresses belonging to the VPNs. The CEs routers advertise IPv4 and IPv6 addresses, while at the PE IPv4 and IPv6 addresses are independently advertised with route targets [41]. The fact that IPv4 and IPv6 L3VPNs can coexist will enable an easy network migration from IPv4 to IPv6, while in the transition period the coexistence of services is seamless, regardless of the IP protocol version. MPLS spares the network operators from deploying IPv6 tunnels through the IPv4 network or the upgrade of the entire IPv4 network to IPv6 or to dual stack, turning it a lot harder to manage and operate [35]. Resuming, the MPLS infrastructure provides the means to easily deploy IPv6 by layering it progressively in the network while keeping the operation of the current IPv4 paradigm.

II.6.5 IPv6 at the Network Core

In the MPLs world, the native IPv6 operations don't differ from the IPv4, because the label space, size, signaling, and forwarding do not change with the IP protocol version.

When IPv6 becomes universally used from the core network to the end users, all routing protocols and label-distribution protocols will then be IPv6-capable [42]. TE in an IPv6 core would imply that all TE features would work with IPv6 prefix mapping, and the forwarding of IPv6 traffic would be done using TE tunnels. The tunnels themselves would be signaled using RSVP with IPv6 extensions, where techniques used with fast reroute could also be used to protect IPv6 traffic from link or node failures [8].

II.6.6 Inter-Autonomous system (AS) with 6PE and v6VPNs

Inter-AS capability is a fundamental piece in the Backbone, because provider networks are seldom single autonomous system (AS) networks. IPv6 information must be able to be carried across ASs, establishing the link between remote PEs. The peering model at the Autonomous System Boundary Routers (ASBR) follows the IPv4 peering model [8]. As it was already mentioned, both ASBRs have to be configured with IPv4 and IPv6 addresses.

II.7 MPLS – Quality of Service (QoS)

Service providers that offer IP services over an MPLS backbone must support IP QoS over their MPLS infrastructure. This means supporting IP QoS over MPLS VPNs or MPLS traffic-engineered paths. MPLS can help service providers offer IP QoS services more efficiently over a wider range of platforms, such as ATM LSRs. Certain useful QoS capabilities, such as guaranteed-bandwidth LSPs, can be supported over MPLS networks [43].

Two models for IP QoS implementation are defined, IntServ and DiffServ [3]:

- *IntServ* follows the signaled QoS model, in which the end hosts signal their QoS needs to the network for reservation of bandwidth and device resources.
- *DiffServ* works on the provisioned QoS model, in which network elements are set up to service multiple classes of traffic with varying QoS requirements.

The IntServ and DiffServ models can be driven off a policy base using the COPS protocol.

II.7.1 Integrated Services

IntServ provides for an end-to-end QoS solution based on end-to-end signaling, state maintenance (for each RSVP flow and reservation), and admission control at each network element. It also specifies various signaling protocols. RSVP is an IntServ signaling protocol that is used to make requests for QoS using the IntServ service classes.

II.7.1.1 IntServ Service Classes

IntServ defines two service classes: guaranteed service and controlled load. These service classes can be requested via RSVP.

- *Guaranteed Service:* provides for hard bounds on end-to-end delay and assured bandwidth for traffic that conforms to the reserved specifications. Guaranteed service requires every flow using the service to be queued separately, which often results in low network utilization [43].
- *Controlled Load:* provides for a better-than-best effort and low delay service under light to moderate network loads. Thus, it is possible in theory to provide the requested QoS for every flow in the network, provided that it is signaled using RSVP and the resources are available [45].

II.7.1.2 RSVP

RSVP is the IntServ signaling protocol that lets applications signal QoS requirements to the network. The network then acknowledges the QoS request with a success or failure reply. RSVP carries classification information, including the source and destination IP addresses and UDP port numbers, so that flows with particular QoS requirements can be recognized within the network. RSVP also carries Tspecs, Rspecs and information on the desired service class. RSVP carries this information from the application to each and every network element along the path from sender to receiver [46].

II.7.1.3 MPLS Implementation of IntServ

MPLS can be enabled on LSRs by associating labels with flows that have RSVP reservations. Packets for which an RSVP reservation has been made can be considered FECs, with the respective indentifying label. Bindings created between labels and the RSVP flows must be distributed between the LSRs: on receipt of an RSVP PATH message, the host responds with a standard RSVP RESV message [47]. As shown in image II.21, LSR3 receives the RESV message, allocates a label from its pool of free labels, and sends out an RESV message with a LABEL object and the value of the label (7) to LSR2 and it also assigns this label as the incoming label in its LFIB. LSR2 in turn creates an entry in its LFIB with label 7 as the outgoing label, it then allocates a new label (3) to use as the incoming label, which is sent upstream to LSR1. As the RESV message with the LABEL object proceeds upstream, an LSP is established along the RSVP path, and each LSR can associate QoS resources with the LSP.

RSVP extensions can be used to distribute labels as part of the resource reservation process and establish an LSP with reserved resources. Such an LSP is known as a guaranteed bandwidth LSP [3].

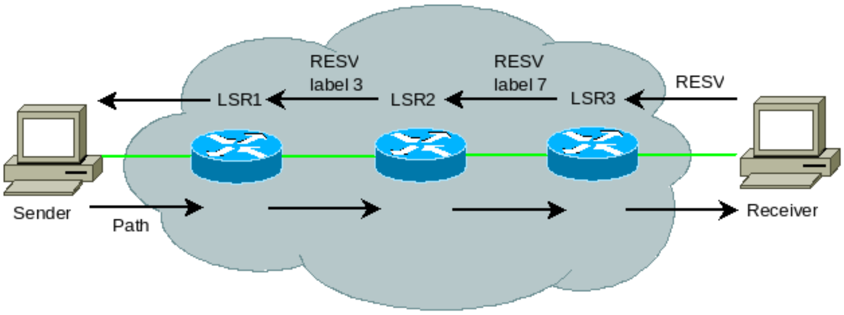


Image II.21: RSVP signaling

II.7.1.4 IP Precedence

The IntServ RSVP per-flow approach to QoS, described in the preceding section, is clearly not scalable and leads to complexity of implementation. IP precedence defined by the IETF has simplified the approach to IP QoS by adopting an aggregate model for flows by classifying various flows into aggregated classes and providing the appropriate QoS for the classified flows [48]. Packets are classified at the edge of the network into one of eight different classes. This is accomplished by setting three precedence bits in the ToS field of the IP header.

II.7.2 Differentiated Services

The DiffServ uses a different approach model where traffic is divided into a small number of classes and resources are allocated on a per-class basis. A 6-bit DSCP is responsible for the differentiation, marking the packet's IP header with the respective class. The DSCP is carried in the ToS byte field in the IP header. Six bits can result in the implementation of 64 different classes.

II.7.2.1 Per-Hop Behavior (PHB)

Network elements or hops along the path examine the value of the Differentiated Services Code Point (DSCP) field and determine the QoS required by the packet. This action is known as a PHB and is illustrated in the following Image II.22. Each network element has a table that maps the DSCP found in a packet to the PHB that determines how the packet is treated. The DSCP is a number or value carried in the packet, and PHBs are well-specified behaviors that apply to packets.
A collection of packets that have the same DSCP value and cross a network element in a particular direction, is called Behavior Aggregate (BA). PHB refers to the packet scheduling, queuing, policing, or shaping behavior of a node on any given packet belonging to a BA [3].

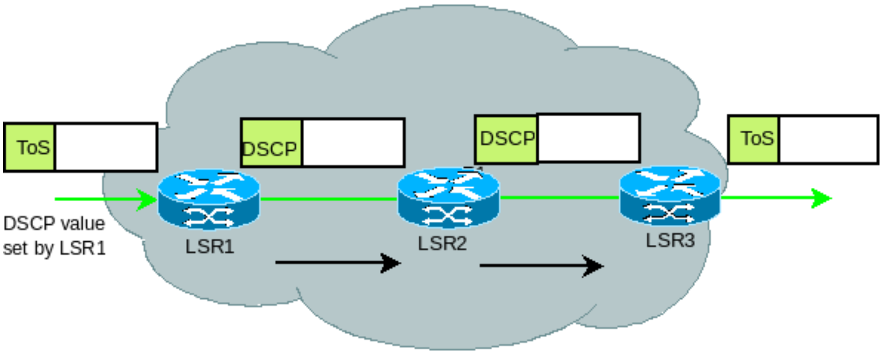


Image II.22: Per-Hop-Behaviour

II.7.2.2 Diffserv Architecture

The DiffServ region can span from one to several DiffServ domains. Each region is in turn configured using the DSCP and the different PHBs, where the entire IP path traveled by the packet must be DiffServ-enabled. A DiffServ domain is composed by DiffServ ingress nodes, DiffServ internal nodes in the core and DS egress nodes. A DiffServ ingress or egress node might be a DS boundary node, connecting two DS domains [49].

II.7.2.3 DiffServ Mechanisms

The DiffServ model only defines the use of the DSCP and the PHBs. The PHBs simply describe the forwarding behavior of a DiffServ-compliant node. The model does not specify how these PHBs may be implemented. A variety of queuing, policing, metering and shaping techniques may be used to affect the desired traffic conditioning and PHB [50].

Traffic Policing

Traffic conditioning can be done using committed access rate (CAR), thus providing PHB for AF classes at the edge and in the core of a DS domain. Different actions are taken upon metering packets, depending on whether the packet in question conforms to, violates, or exceeds the configured average rate. Token bucket, an algorithm responsible to manage the amount of data injected in a particular Network, is used to regulate the traffic. The amount contained below the token bucket parameter Bc is within the configured access rate. Traffic between Bc and Be is excess traffic. Traffic that is more than Bc + Be is dropped. A packet can be transmitted, dropped, or remarked with a different DSCP value, depending on the configured policy [51].

Traffic Shaping

In this approach, packets are buffered rather than simply dropped in the case of congestion. This can be done generically by configuring an average rate, Bc, and Be. FRTS can also be employed to make the traffic slow down when congestion is reported by the Frame Relay switch [52].

II.7.2.4 MPLS DiffServ implementation

MPLS LSRs do not examine the contents of the IP header and the value of its DSCP field as required by DiffServ. The MPLS header has a 3-bit field called Exp and this field is copied into the MPLS Exp field at the edge of the network. Each LSR along the LSP maps the Exp bits to a PHB, MPLS packet's CoS can also be set to a different value, as determined by a service offering [3]. This feature allows the service provider to set the MPLS Exp field instead of overwriting the value in the customer's IP precedence field, leaving the IP header intact and available for the customer use. The customer-configured CoS is not changed as the packet travels through the MPLS backbone. The LSPs created this way are known as E-LSPs or Exp-LSPs. E-LSPs can support up to eight PHBs per LSP [3].

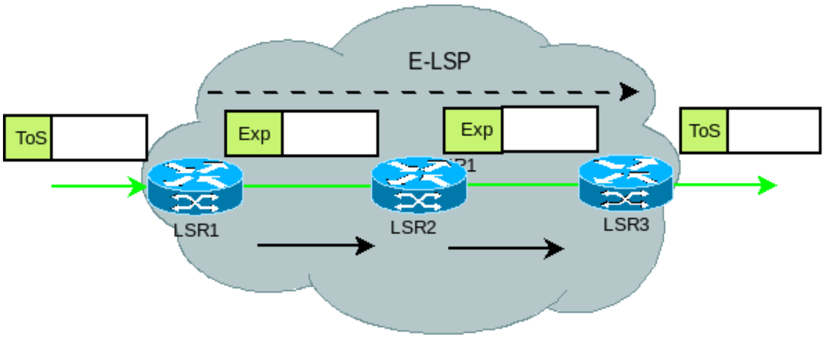


Image II.23: DiffServ Exp header

II.7.3 MPLS VPN Support of QoS

MPLS VPNs are popular as private-WAN alternatives, and its migration from a private-WAN to a MPLS VPN is a significant paradigm shift when addressing QoS designs. The reason for this is that the enterprise customer subscribers must closely cooperate with their service providers to ensure end-to-end service-level. To achieve end-to-end service levels, both enterprise and service-provider QoS designs must be consistent and complimentary. Two models are used to describe QoS in the VPN context - the pipe model and the hose model.

II.7.3.1 Pipe Model

In the pipe model, the service provider provides the VPN customer with certain QoS guarantees for traffic flows between one CE router and another within the same VPN [54]. This model can be represented as a pipe between two CE routers. Any traffic that enters this pipe gets certain QoS guarantees, such as a guaranteed minimum bandwidth between the two CE routers. The PE router at the head end of the pipe can specify which traffic flows are allowed to use the pipe. The MPLS QoS pipe model is similar to the QoS model in the ATM or Frame Relay world. However, the pipe model is unidirectional, permitting traffic pattern in an asymmetric manner, which allows for different traffic rates in either direction, between CE routers.

II.7.3.2 Hose Model

In the hose model, the service provider supplies a customer with certain guarantees for the traffic that a particular CE router would send to and receive from other CE routers in the same VPN [54]. It is easy for a customer to implement the hose model for MPLS QoS within the VPN, because the customer does not have to perform a detailed traffic analysis or capacity planning and specify the traffic distribution between various CE routers.

The two parameters that are used in the hose model are the Ingress Committed Rate (ICR) and the Egress Committed Rate (ECR) [55]. The ICR is the traffic rate at which the CEs in the VPN can receive from a particular CE, and the ECR is the traffic rate at which the CEs in the VPN can send traffic to a particular CE router. The ICR and ECR values are independent of each other and do not need to be the same.

II.8 Any Transport over MPLS

AToM is a direct consequence of the MPLS VPN huge success. Instead of creating a VPN service at Layer 3 like MPLS VPN, AToM creates VPNs at Layer 2, sometimes referred as L2VPN. The AToM intelligence is limited to the PE routers, being in this way a edge technology that employs the MPLS backbone for its transport. AtoMs limitation lies in the fact that it is a point-to-point service, referred to as Virtual Private Wire Service (VPWS).

II.8.1 Transporting Layer 2 Frames over MPLS backbone

Two solutions are available for transporting Layer 2 frames across the network [10]:

- Carry the traffic across an MPLS backbone, which is the AToM solution.
- Carry the traffic across an IP backbone, which is the L2TPv3 solution.

The same architecture concept is used in both the AToM and L2TPv3 solutions, where the difference lies on the underlying network that is carrying the service [56]: AToM is based on pseudowires that are able to carry the customer Layer 2 traffic from edge to edge across the MPLS enabled backbone network. The pseudowire is a connection between the PE routers and emulates a wire that is carrying Layer 2 frames using tunneling mechanisms.

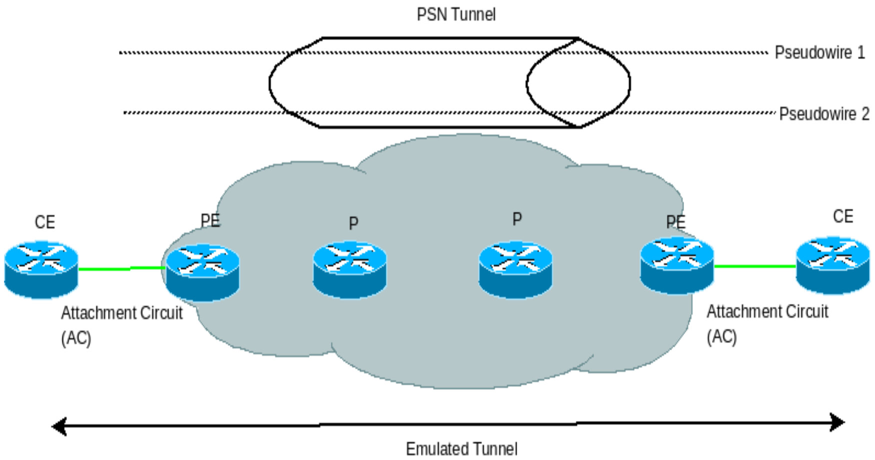


Image II.24: Transporting Layer 2 Frames over the MPLS backbone

II.8.2 AToM Architecture

In networks that use AToM, all routers in the service provider network run MPLS, and the PE routers have an Attachment Circuit (AC) towards the CE router. The PE router receives Layer 2 frames on the AC and encapsulates them with labels before sending them onto the PSN tunnel towards the remote PE. At the remote PE, the label(s) are removed and the frames are sent towards the remote CE [10].

II.8.2.1 Data Plane of AToM

As the ingress PE receives a frame from the CE, it forwards the frame across the MPLS backbone to the egress LSR with two labels: the tunnel label and the VC label. In an AToM network, each pair of PE routers must run a targeted LDP session between them [8]. The targeted LDP session signals characteristics of the pseudowire and, most importantly, advertises the VC label. The VC label is always the bottom label in the label stack, identifying the egress AC on the egress PE. The tunnel label is the top label in the label stack and tells all intermediate LSRs to which egress LSR the frame must be forwarded [8]. This process is illustrated bellow in Image II.25:

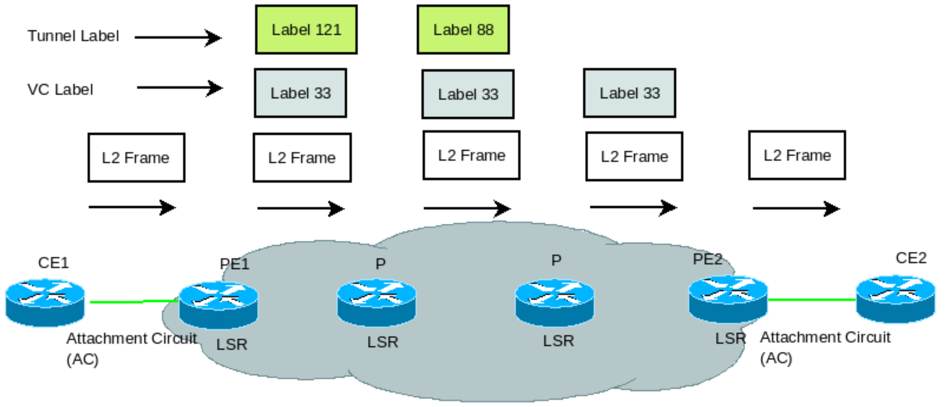


Image II.25: Data Plane of AToM

II.8.2.2 Signaling the Pseudowire

A targeted LDP session between the PE routers signals the pseudowires. In essence, the signaling LDP protocol sets up and maintains the pseudowires between the PE routers [55]. LDP has been extended with new Type Length Value fields (TLVs) to perform this job. The main purpose of this LDP session between the PE routers is to advertise the VC label that is associated with the pseudowire. This label is advertised in a Label Mapping message using the downstream unsolicited label advertisement mode [55].

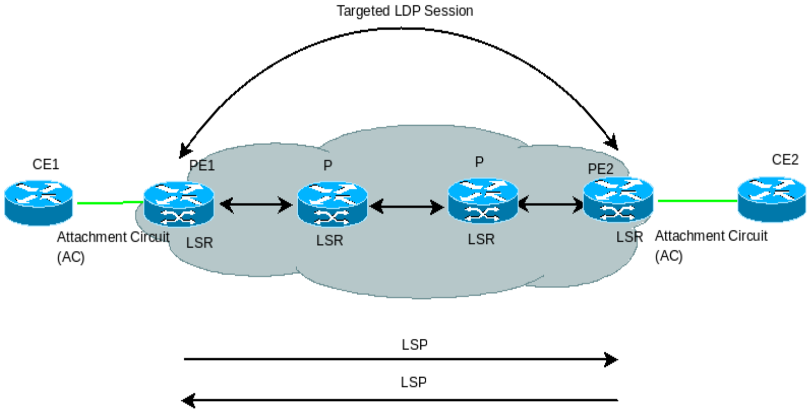


Image II.26: Signaling the pseudowire

II.8.3 Transported Layer 2 Protocols

AToM supports several Layer 2 protocols that can be transported across the MPLS network. ACs on both sides of the MPLS network are of the same encapsulation type. This is referred to as like-to-like functionality [56].

- For HDLC, the customer-facing interface on the PE router needs the encapsulation HDLC command.

- For PPP, the customer-facing interface on the PE router needs the encapsulation ppp command
- Frame Relay can be carried across the MPLS network in two fashions: DLCI-to-DLCI or port-to-port. In the DLCI-to-DLCI method, one VC is carried over one pseudowire. In the port-to-port method, all VCs on one port are carried over one pseudowire.
- Carrying ATM over MPLS (ATMoMPLS) is possible with the ATM adaptation layer 5 (AAL5) encapsulation whereby ATM cells are used to transport larger frames. Another possibility is ATM cell relay mode. In ATM cell relay mode, each individual ATM cell is transmitted over the MPLS network or multiple cells are packed together if cell packing is used.
- The AToM solution to carry Ethernet over MPLS is strictly point-to-point. In essence, all Ethernet frames are carried from one ingress PE to one egress PE router. This is the equivalent of LAN-to-LAN bridging over point-to-point WAN links. The connection is not multipoint, so no emulation of a LAN-like functionality occurs whereby one frame enters the ingress PE and is broadcasted to all PE routers that belong to the same Layer 2 VPN.

II.8.4 AToM Tunnel Selection

Traffic can be steered through the MPLS network over a specified path using the AToM Tunnel selection feature [10]. Enabling this feature requires setting up an MPLS TE tunnel from PE to the next PE router with the specification that the AToM traffic should take the TE tunnel instead of the default shortest labeled path. In addition, it is possible to specify the fallback behavior to the default path when the primary TE tunnel fails.

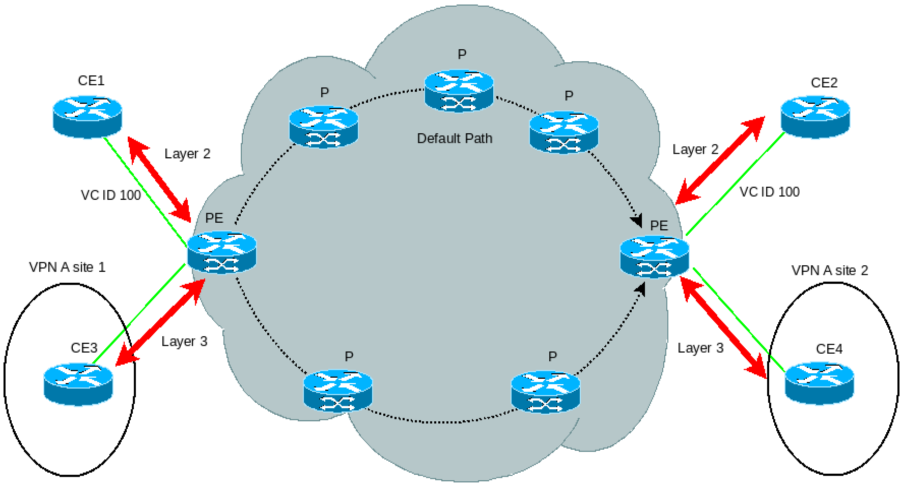


Image II.27: AToM Tunnel Selection

II.9 Virtual Private LAN Service

VPLS is a way to provide Ethernet based multipoint to multipoint communication over IP/MPLS networks [57]. It allows geographically dispersed sites to share an Ethernet broadcast domain by connecting sites through pseudowires. VPLS is a virtual VPN technology that allows the extension of the LAN at each site, to the edge of the provider network, where a switch or bridge is emulated by the provider network to connect all customer LANs in a single bridged LAN [58].

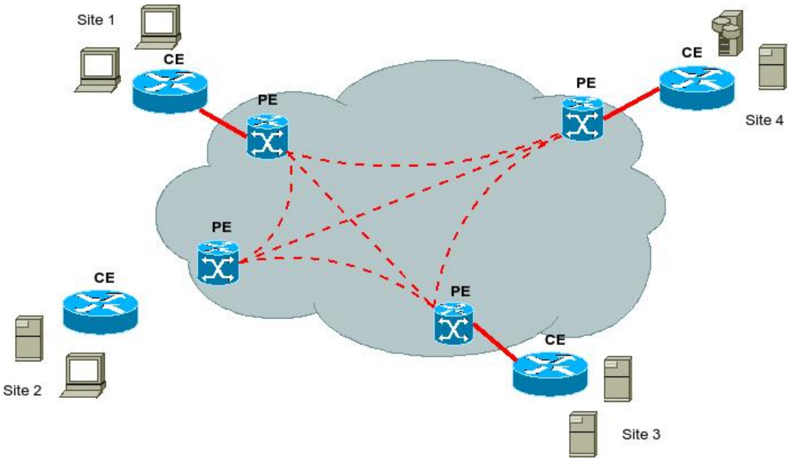


Image II.28: Virtual Private LAN Service

Like MPLS based IP-VPNs, VPLS is a multipoint service, where the difference lies on the fact that VPLS is able to transport non-IP traffic. VPLS is a layer-2 multipoint VPN that enables to interconnect multiple sites in a single bridged domain over a provider managed IP-MPLS network [58]. From the customer perspective, they all appear to be in the same LAN, regardless of their physical location. VPLS uses an Ethernet service interface with the customer, simplifying the LAN/WAN boundary and allowing flexible and rapid provisioning.

II.9.1 VPLS Architecture

A VPLS service emulates a LAN or the functionality of an Ethernet switch, capable of the following features [60]:

- Forwarding of Ethernet frames;
- Forwarding of unicast frames with an unknown destination MAC address;
- Replication of broadcast and multicast frames to more than one port - loop prevention;
- Dynamic learning of MAC addresses;
- MAC address aging.

II.9.1.1 VPLS Components

Like the previously mentioned MPLS underlying technologies, VPLS networks require three main components, the CE, the PE and the core MPLS network [59]. The CE device is a router or a switch located at the customer premises. The PE device is where all the intelligence of the VPLS network resides. This is where the VPLS originates and terminates. The PE also has all the necessary tunnels set up to connect to the other PEs of the VPLS network. The IP/MPLS core network interconnects the PEs. It does not participate in the VPLS functionality. Traffic is switched based on the MPLS labels. The basis of a multipoint VPN service like VPLS is the full mesh of the MPLS tunnels that are set up between all the PEs participating in the VPN service. For every VPLS instance, the full mesh of pseudowires is created between all PEs that participate in the VPLS instance.

II.9.1.2 VPLS Data Plane

In the data plane, the transported frames look the same as Ethernet frames in the AToM model. Two labels are imposed on the Ethernet frame. The top label or tunnel label identifies the tunnel (LSP) that the frame belongs to. That means, it forwards the frame from the local or ingress PE to the remote or egress PE. The bottom label is the VC label, and it identifies the pseudowire: The egress PE looks at the VC label; it uses the VC label to determine to which attachment circuit the frame should be forwarded to [8].

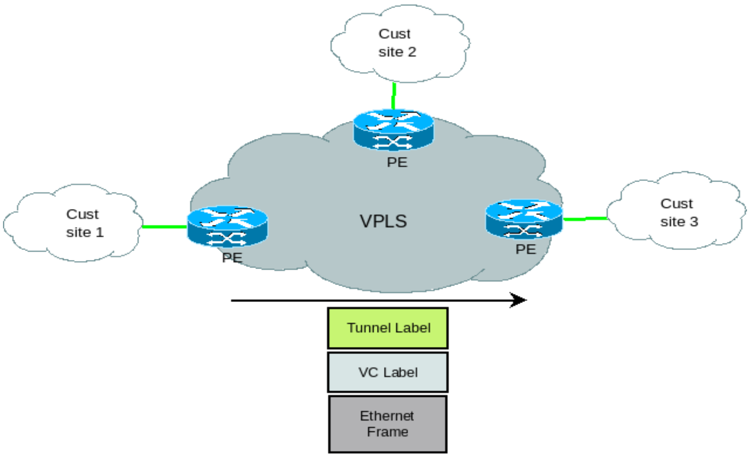


Image II.29: VPLS Data Plane

II.9.2 VPLS Advantages

VPLS has significant advantages for customers, like being simpler and more cost effective to operate than a traditional service. Customers benefit because they can connect all of their sites to an

Ethernet VPN that provides a secure, high speed and homogenous network. Moreover, VPLS provides a logical next step in the continuing evolution of Ethernet from a 10 Mbps shared LAN protocol to a multi-Gbps global service [59].

II.9.3 VPLS Signaling

The VPLS principle lies on emulated LAN that requires full mesh connectivity. There are two methods to establish this mesh [56]:

- using BGP;
- using LDP.

PE routers use the Control plane to establish communication, for auto-discovery and signaling. Auto-discovery is the process of finding other PE routers participating in the same VPN or VPLS. Signaling is the process of establishing pseudowires.

The pseudowires are built using the Data plane, which allows the traffic flow between PEs. Using LDP as the advertising mechanism, each PE router must be configured to participate in a given VPLS and, in addition, be given the addresses of other PEs participating in the same VPLS. A full mesh of LDP sessions is then established between PEs. LDP is then used to create an equivalent mesh of pseudowires between those PEs.

In this WAN constellation, loop avoidance becomes mandatory, being achieved by restricting the PEs to never forward received frames to another PE. PEs participating in a VPLS based VPN must appear as an Ethernet bridge to connected CE devices. Received Ethernet frames must be managed ensuring that CEs can be simple Ethernet devices. When a PE receives a frame from a CE, it inspects the frame and learns the CE MAC address, storing it locally along with LSP routing information. It then checks the frame destination MAC address. If it is a broadcast frame, or the MAC address is not known to the PE, it floods the frame to all PEs in the mesh [56].

III. Emulators - State of the Art

This chapter focuses on diverse aspects of modern emulators used to recreate real routers for the purpose of studying and testing network configurations.

The market is crawling with many different tools, mainly devoted to training for the Cisco certification exams. This dissertation will only refer the ones that are powerful and complete enough to emulate the Cisco IOS and support MPLS.

While simulators are useful for some applications, they attempt to recreate the Cisco IOS using a programming language. On the other hand, emulators take the real IOS router code and emulate the hardware behind it, so that it is possible to run the real IOS together with its characteristics and functionalities (based on the limitations of the emulated hardware, of course).

III.1 Available Software

Cisco has a parallel career path with its own study program, the so called Cisco certifications. Many people engage in this study program and motivate programmers to develop simulation tools that can help on the practice acquisition.

Many simulators and emulators are available, both commercial and open source. The following are some examples:

- **SemSim** [61] - Free Trial Version of Cisco router simulator with free Cisco Certified Network Associate certification (CCNA) router simulation lab & 20 flash cards. It is demo, easy to download, with good training and functionality features.
- **Boson NetSim** [62] - Good demo versions of the Cisco router simulators. The demo download files can be very big if downloaded on a dial-up connection (nearly 40Mb) but the demo offers good functionalities to get started.
- **CertExams** [63] - The demo version of the Cisco router simulation is limited to a few labs, and only up to global configuration mode is supported. The downloadable package size is about 13Mb.
- **RouterSim's Network VisualizerT 4.0.** [64] - A good functional demo, with more eye candy at the interface level. It is a 37Mb package but there are video captures of the program on the download page to help deciding to download this free router demo or not.
- **MIMIC CCNA Virtual Lab** [65] - Good functional Cisco router simulator that has a

number of good, practical features with an intuitive interface. The free trial version can be a little off-putting to download due to all the questions that have to be filled out.

- **TechExams** [66] - A free Java-based Cisco router simulation that can be used only after registering in the site. The simulator has training prompts to help.
- **Packet Tracer** [67] **-** A powerful router simulator created by Cisco Systems and provided for free distribution to faculty, students, and alumni who are or have participated in the Cisco Academy program. The purpose of Packet Tracer is to offer students and teachers a tool to learn the principles of networking, as well as develop Cisco Technology specific skills. Version 5.2 is available for Windows and Linux operating systems.
- **NetworkSims** [68] - A Cisco simulator which supports a wide range of Cisco products, Cisco router, Cisco switch, Cisco PIX/ASA and Cisco wireless. It has various topologies, and fault-finding challenges, along with fun activities. It covers a wide range of devices and includes online tests. It has been rated as a useful simulator for teaching, and is seen as being a safe study site from CertGuard.

III.2 Dynamips Cisco Emulator

Dynamips [69] is an emulator program written by Christophe Fillot with the objective of emulating Cisco routers. Dynamips runs diverse Operating system platforms including Linux, Mac and Windows, being able to emulate the Cisco hardware routing platforms by directly booting an actual Cisco IOS software image into the emulator. It allows testing diverse IOS functionalities on a desktop PC or deployed in server, without the need of an actual physical Cisco device. Dynamips scales the emulation of complex scenarios composed by multiple routers, supporting currently different networking medium such as Ethernet, Serial link, ATM, and POS interfaces for the 1700, 2600, 3600, 3700, and 7200 hardware platforms [70].

Dynamips is in active development and there are a few add-ons written for it. One of the most popular is Dynagen, which is a front-end add-on that allows the use of an INI configuration file to provision Dynamips emulator networks. The source code is licensed under the GNU GPL [71].

III.2.1 Resource Utilization

Dynamips emulates the hardware in such a way that can run IOSs directly. Regarding resource usage, it is extensive concerning RAM and CPU: for example, running an IOS image that requires 256 MB of RAM will allocate 256 MB of RAM in the virtual router instance, using consequently

256 MB of RAM from the main system. In parallel, Dynamips also allocates memory to cache JIT translations, (by default) 64 MB of RAM per instance on Unix systems. Summing this two portions and multiplying the result by the number of routers deployed will account for the total working size. There are mechanisms that in practice allow to control the amount of actual RAM used, and keep it in a reasonable size. Dynamips architecture uses memory mapped files for the routers virtual memory, which are visible in the working directory as temporary "ram" files and are equal to the size of the virtual routers RAM size. The OS have the ability to cache in RAM the sections of the mmap files that are being used, significantly reducing the total final amount of consumed RAM.

On the other hand, it is a lot harder to keep Central Processing Unit (CPU) usage from becoming abusive, especially when applied to large scenarios, because operation depends on emulating a router's CPU instruction-by-instruction. There is a function that can be used once the scenario has run, the so called "Idle-PC" process for a given IOS image. This is found when the CPU is idle, so it dutifully executes all the instructions that make up IOS's idle routines just as it would execute the instructions that perform "real" work. Although CPU utilization decreases drastically, it still raises steadily with the number of routers used in parallel, especially because routers are machines that strongly depend on a real time response. Delays virtually render this kind of emulation practically useless. This subject will be continued later in this chapter by regarding to distributed computing and cloud computing.

III.3 Dynagen

When elaborated scenarios must be deployed with multiple routers running in parallel and being interconnected with multiple and diversified network mediums (like Ethernet, Serial, Gigabit), it becomes useful to use a front-end for Dynamips that is in charge of managing this complexity. Dynagen simplifies building and working with virtual networks, using a .ini configuration file to provision Dynamips with the required information. In this file all aspects of the network can be specified. It takes care of connecting the right port adapters, generating and matching NIO descriptors, specifying bridges, frame-relay, ATM switches, etc. It also provides a management CLI for listing devices, suspending and reloading instances, determining and managing idle-pc values, performing packet captures, etc.

Dynagen works with a text-based interface, which uses the "Hypervisor" mode for communication with Dynamips. Resuming, Dynagen [71]:

- Uses a simple, easy to understand configuration file for specifying virtual router hardware configurations;

- Simple syntax for interconnecting routers, bridges, frame-relay and ATM, and Ethernet switches. No need to deal with NetIOs;
- Can work in a client/server mode, with Dynagen running on your workstation communicating with Dynamips running on a back-end server. Dynagen can also control multiple Dynamips servers simultaneously for distributing large virtual networks across several machines. You can also run Dynamips and Dyngen on the same system;
- Provides a management CLI for listing devices, starting, stopping, reloading, suspending, resuming, and connecting to the consoles of virtual routers.

Dynagen is written in Python and is thus compatible with the platforms for which there is a Python interpreter. The design is modular, with a separate OOP API for interfacing with Dynamips [71]. Other Python applications that use API to provision virtual networks or to provide other front-ends are useful: e.g. GNS-3 (Graphical Network Simulator-3), a Graphical User Interface (GUI) front-end using this library, which is currently under development.

III.3.1 Network File configuration

Dynagen network, device interconnections, and hardware configurations that constitute the virtual lab are stored in a single "network file". In this file, with a simple syntax, it is possible to program the previously mentioned features:

- The type of Router instance;
- On each Router, the interfaces and their respective interconnections with other devices.

A network configuration file can look like this:

```
autostart = False
[localhost:7200] #The host and port where Dynamips will be running, that can be either on the same machine as
Dynagen or on a different machine, allowing to distribute the computing load .
   workingdir = /tmp #The working directory where the the  RAM of the virtual router instance will be temporary
stored, this is the reason that this file is mapped in a temporary directory.
   udp = 10000 #This port is responsible for the internal interconnections between routers
   [[3640]] #Router type, in this case a 3600 series
          image = /home/dynagen/ios/c3640-jk9s-mz.124-16a.bin.unpacked
#path to were the IOS image is stored, unpacking the IOS is another way to speedup the booting process.
            ram = 128 # RAM size allocated for the Router, each model as minimum required amount.
            disk0 = 0
            disk1 = 0
            idlepc = 0x60521bc8
            mmap = True
            ghostios = True #See memory optimization
            sparsemem = True #See memory optimization
    [[ROUTER NetworkA]] #Router name a subsquente configurations
        cnfg = /home/dynagen/labs/te/config/NetworkA.cfg #Path to were a configuration file is stored, this configuration will be used at the router boot time and it contains its configurations.
        Model = 3640 #Model reference
        console = 2000 #Console port that will be used to connect to the router using telent as it would be in real life the console port be.
        slot0 = NM-1FE-TX #slot type there are currently a number of different slot types suported allowing to have multiple interface types.
        f0/0 = CE1 f1/1 #Interface connection configuration, with the source port well as port a router destination of the connection.
....
```

III.3.2 Idle-PC values

As most computing systems, Dynamips has long idle times that could be better used by the OS in some other process, instead of allocating fixed computing resources. The “idlepc” command determines the most likely points in the code that represent an idle loop in IOS [72].

Applying this feature, it is possible to occasionally “hibernate” the virtual router in the idle moments and then bring it back to “life” when it is performing useful work. Once applied, it is possible to significantly reduce the CPU consumption on the host without reducing the virtual router’s capacity to perform real work. This process has to run thoroughly the first time a lab is deployed and, once calculated, the value stored in the database will be automatically applied whenever a router in one of the labs uses this image.

This statistic values are particular to an IOS image, every IOS version will have its own, and even different feature sets of the same IOS version will end up with a different one [71]. However, Idle-PC values are not particular to any host, operating system, or to the version of Dynamips, so the “.ini” files from one system can be freely copied, merged, shared with other scenario, without having to performe the calculation every time. More on this subject can be referred to the Dynagen tutorial, where everything is explained in a detailed and comprehensive way [71].

III.3.3 Memory Usage Optimizations

Labs can consume a large amount of real and virtual memory. The "ghostios" and "sparemem" options were added to address both of these issues, respectively.

The **ghostios** option can significantly reduce the amount of real host RAM needed for labs with multiple routers running the same IOS image. With this feature, instead of each virtual router storing an identical copy of IOS in its virtual RAM, the host will allocate one shared region of memory that they will be used by all routers. For example, running 10 routers with the same IOS image, having 60 MB of size, will allow savings of up to 9*60 = 540 MB of real RAM upon running the lab. Enabling ghostios is as simple as specifying "ghostios = true" in the network file [71]. This option can be used in several places:

- If used at the top level, ghostios is applied to all router instances in the lab;
- If used at the defaults section (e.g. "[[7200]]"), it only applies to that model of router on that dynamips server;

Note that ghostios (and all other top level parameters for that matter) cannot be specified at the server level. They will be ignored. "ghostios" cannot also be specified at the router level. Typical usage is to specify "ghostios = true" at the top level. Dynagen is smart enough to use only ghostios if there is more than one router using the same IOS image.

When enabled, additional files will be saved in the same directory as the router nvram files. These are the files that contain the shared memory region. Other files typically created with a router instance are present as well (log, nvram, and possibly bootflash files). Measuring the amount of host memory saved with ghostios can be a little intricate due to the complexities of memory management in modern OSs [71].

The "**sparsemem**" feature does not conserve real memory, but instead reduces the amount of virtual memory used by the router instances. This can be important, because some OS limit a single process to few GB of virtual memory. For example, on Windows, after the VM space used by cygwin and other libraries that dynamips depends on, this only leaves room for 4 router instances of 256 MB each! Enabling sparsemem will only allocate virtual memory on the host that is actually used by IOS in that router instance, rather than the entire amount of RAM configured. This can allow to run more instances per dynamips process, instead of running multiple dynamips processes [71,72].

III.3.4 Real Network connection

It is possible to connect real interfaces on the host computer to the Dynamips virtual router interfaces. This is done by bridging real and virtual interfaces, allowing the virtual network access the real world and vice-versa. This feature can be very interesting as it allows some operations that cannot be so easily virtualized in Dynamips, like for example traffic generators [73]. In Linux systems, this is done with the NIO_linux_eth NIO descriptor:

```
f0/0 = NIO_linux_eth:eth0
```

This bridges this router's F0/0 interface to the eth0 interface of the host. Packets that exit f0/0 are dumped out on to the real network through eth0, and return packets are forwarded back to the virtual router instance accordingly [73].

III.3.5 Dynamic Configuration Mode

Dynagen has merged with Pavel Skovajsa's confDynagen fork, supporting following version 0.11.0 Dynamic Configuration Mode [71]. This significant enhancement allows to dynamically change the lab by editing the ".net" config on the fly without needing to exit Dynagen or even stop running devices. It is possible to add/change/remove all lab/router options that are available in the .net files without restarting the running lab. Hypervisors enjoy the same flexibility and can also be added/changed/removed without restarting operations.

Dynagen supports running without specifying a NET file to load; this will run Dynagen with empty lab, and it is possible to create the new lab in configuration mode. This is rather unwieldy, but when creating a big lab from the beginning it can be done.

III.3.6 Client / Server and Multi-server Operation

The Dynamips "Hypervisor" mode that is used by Dynagen is a TCP/IP communications channel, so the Dynagen client can run on a different machine than the Dynamips emulator [71]. This is done by specifying a host other than "localhost" in the network file.

The DNS name or the IP address of the remote host must be used instead of "localhost" when identifying the system. The reason is that the other server defined below will use this name when talking to the local system. Connecting to a device on another system is as simple as specifying it the same way as it would be on the local system. Any connectivity method or device supported by

Dynamips can then be used (Ethernet, Serial, ATM, Bridges, Ethernet switches, Frame Relay Switches, etc.). This “transparent” connectivity is available in Dynagen versions later than 0.4. The .ini file would look like:

```
autostart = False
[Server:7200]
    workingdir = /tmp
    udp = 10000
    [[3640]]
             image = /home/dynagen/ios/c3640-jk9s-mz.124-16a.bin.unpacked
                ram = 128
                disk0 = 0
                disk1 = 0
                idlepc = 0x60521bc8
                mmap = True
                ghostios = True
                sparsemem = True
    [[ROUTER NetworkA]]
        cnfg = /home/dynagen/labs/te/config/NetworkA.cfg
        model = 3640
        console = 2000
        slot0 = NM-1FE-TX
        f0/0 = CE1 f1/1
[Server2:7202] #This can be running on a different server as is here exemplified or it can be running on the same server and simply have the router instances running in separate hypervisors with the objective of improving efficiency as explained below in text.
    workingdir = /tmp
    udp = 10030
      [[7200]]
 image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER CE1]]
        cnfg = /home/dynagen/labs/te/config/CE1.cfg
        console = 2011
        slot1 = PA-2FE-TX
        f1/0 = PE1 f2/0
        f1/1 = NetworkA f0/0
        slot0 = NM-1FE-TX
        f0/0 = CE3 f1/1
…
…
```

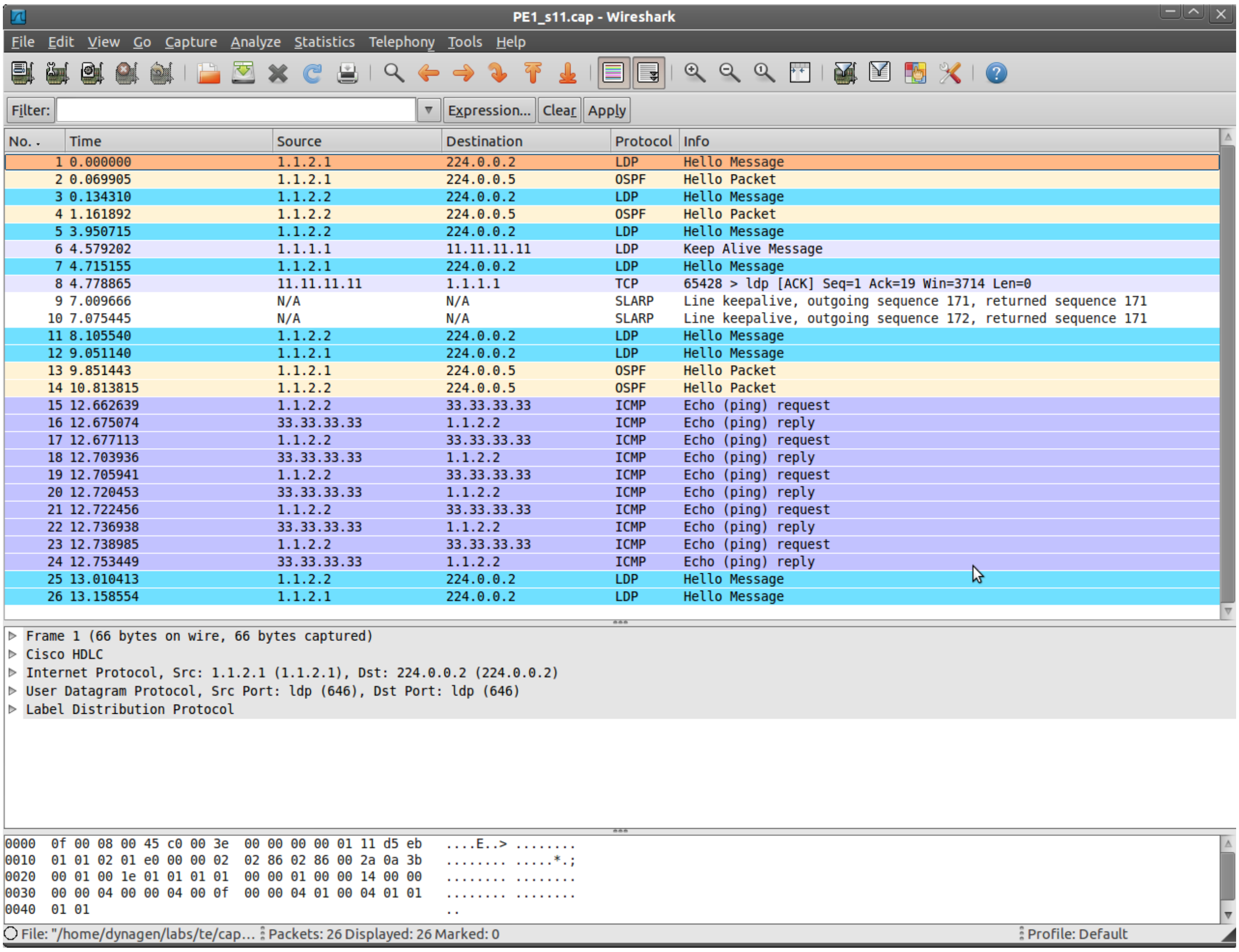
Here, Dynagen will be talking to servers named “Server” and “Server2”, also specifying the TCP ports where the Dynamips process is listening on as 7200 and 7202. Port 7200 is the default, so it is not actually necessary in an instance with a single hypervisor. Setting up Dynamips to listen on different ports and servers has to be specified here.

When using remote servers, it is necessary to specify the working directory for the specific lab, because Dynamips needs to store several files in this directory, including the NVRAM for the virtual router, as well as the bootflash, logfiles, and some other working files [71]. When running Dynamips and Dynagen on the same machine, it is not necessary to specify the working directory, because Dynagen uses the same directory as the network file. But in a distributed setup, the network file is on the client and the working files are on the host. So, the fully qualified path to the working directory must be specified on the Dynamips host [71]. Any host based firewalls running on all Dynamips servers must permit the necessary traffic. This includes the Dynamips server port (defaults to TCP 7200), the console ports (e.g. TCP 2000, 2001, …) and the ports used by the NIO connections between interfaces, which start at UDP 10000 and increases from there [71]. Today, computing machines use multi-threading and multi-core processors. Thus, it is advantageous to run each router instance in a different hypervisor, so that CPU and RAM usage are managed by the OS instead of fully trusting Dynamips for the effect. This can be done by launching different hypervisor instances per router, which can be easily done with the help of a script and then spiting the Routers in the .net file as shown below:

```
#!/bin/bash
#Launching different Hypervisors per Router instance well as, different router types, 3600 and 7200
sudo dynamips -H 7200 /home/dynagen/ios/c3640-jk9s-mz.124-16a.bin.unpacked &
sudo dynamips -H 7201 /home/dynagen/ios/c3640-jk9s-mz.124-16a.bin.unpacked &
sudo dynamips -H 7202 /home/dynagen/ios/c3640-jk9s-mz.124-16a.bin.unpacked &
sudo dynamips -H 7203 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7204 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7205 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
…
```

III.3.7 Packet Capture

Dynamips/Dynagen can capture packets on virtual Ethernet or Serial interfaces and write the output to a capture file to be used with applications like tcpdump, Wireshark, or any other application that can read the libpcap capture file format.


PE1_s11.cap - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter:
Expression...
Clear
Apply

No.	Time	Source	Destination	Protocol	Info
1	0.000000	1.1.2.1	224.0.0.2	LDP	Hello Message
2	0.069905	1.1.2.1	224.0.0.5	OSPF	Hello Packet
3	0.134310	1.1.2.2	224.0.0.2	LDP	Hello Message
4	1.161892	1.1.2.2	224.0.0.5	OSPF	Hello Packet
5	3.950715	1.1.2.2	224.0.0.2	LDP	Hello Message
6	4.579202	1.1.1.1	11.11.11.11	LDP	Keep Alive Message
7	4.715155	1.1.2.1	224.0.0.2	LDP	Hello Message
8	4.778865	11.11.11.11	1.1.1.1	TCP	65428 > ldp [ACK] Seq=1 Ack=19 Win=3714 Len=0
9	7.009666	N/A	N/A	SLARP	Line keepalive, outgoing sequence 171, returned sequence 171
10	7.075445	N/A	N/A	SLARP	Line keepalive, outgoing sequence 172, returned sequence 171
11	8.105540	1.1.2.2	224.0.0.2	LDP	Hello Message
12	9.051140	1.1.2.1	224.0.0.2	LDP	Hello Message
13	9.851443	1.1.2.1	224.0.0.5	OSPF	Hello Packet
14	10.813815	1.1.2.2	224.0.0.5	OSPF	Hello Packet
15	12.662639	1.1.2.2	33.33.33.33	ICMP	Echo (ping) request
16	12.675074	33.33.33.33	1.1.2.2	ICMP	Echo (ping) reply
17	12.677113	1.1.2.2	33.33.33.33	ICMP	Echo (ping) request
18	12.703936	33.33.33.33	1.1.2.2	ICMP	Echo (ping) reply
19	12.705941	1.1.2.2	33.33.33.33	ICMP	Echo (ping) request
20	12.720453	33.33.33.33	1.1.2.2	ICMP	Echo (ping) reply
21	12.722456	1.1.2.2	33.33.33.33	ICMP	Echo (ping) request
22	12.736938	33.33.33.33	1.1.2.2	ICMP	Echo (ping) reply
23	12.738985	1.1.2.2	33.33.33.33	ICMP	Echo (ping) request
24	12.753449	33.33.33.33	1.1.2.2	ICMP	Echo (ping) reply
25	13.010413	1.1.2.2	224.0.0.2	LDP	Hello Message
26	13.158554	1.1.2.1	224.0.0.2	LDP	Hello Message

```
▷ Frame 1 (66 bytes on wire, 66 bytes captured)
▷ Cisco HDLC
▷ Internet Protocol, Src: 1.1.2.1 (1.1.2.1), Dst: 224.0.0.2 (224.0.0.2)
▷ User Datagram Protocol, Src Port: ldp (646), Dst Port: ldp (646)
▷ Label Distribution Protocol
```

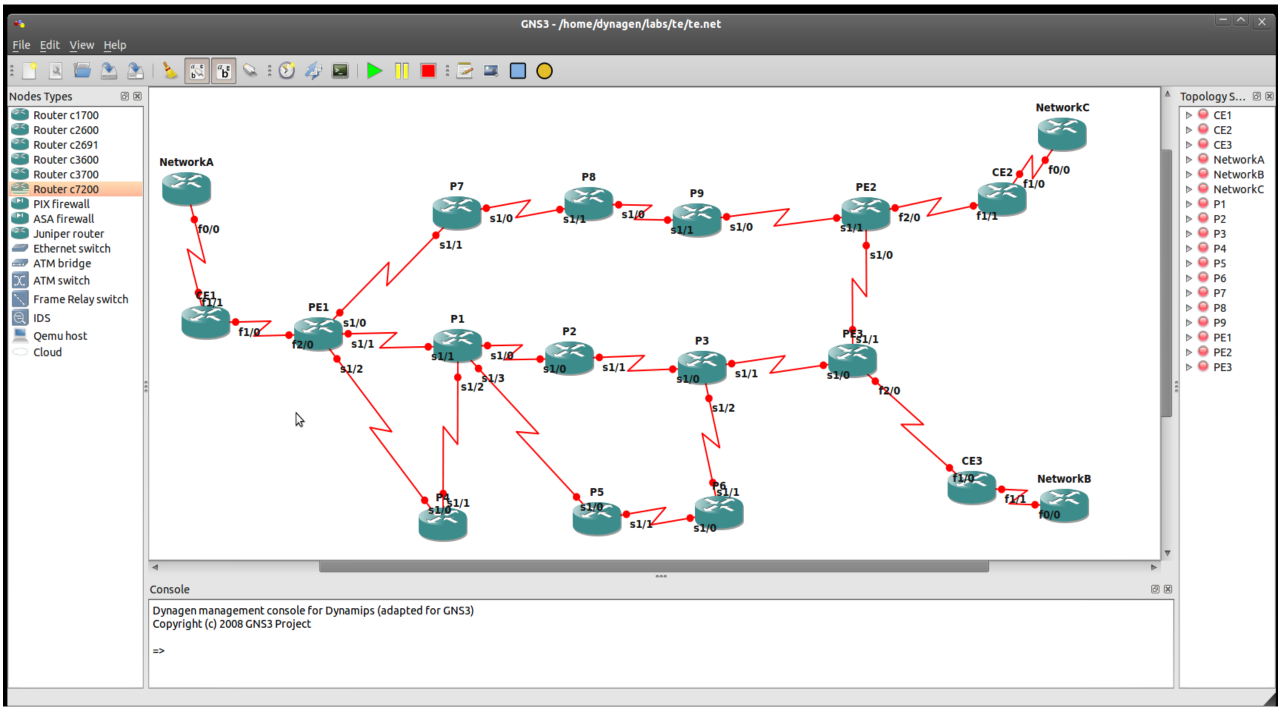
```
0000  0f 00 08 00 45 c0 00 3e  00 00 00 00 01 11 d5 eb   ....E..> ........
0010  01 01 02 01 e0 00 00 02  02 86 02 86 00 2a 0a 3b   ........ .....*.;
0020  00 01 00 1e 01 01 01 01  00 00 01 00 00 14 00 00   ........ ........
0030  00 00 04 00 00 04 00 0f  00 00 04 01 00 04 01 01   ........ ........
0040  01 01                                              ..
```

File: "/home/dynagen/labs/te/cap... Packets: 26 Displayed: 26 Marked: 0 Profile: Default

Image III.1: Dynamips/Dynagen Packet Capture

III.4 GNS3

GNS3 stands for Graphical Network Simulator and it is a graphical front-end based on the underling technologies dynagen and dynamips, developed in open source. It is free program that may be used on multiple operating systems, including Windows, Linux, and MacOS X.



```
Dynagen management console for Dynamips (adapted for GNS3)
Copyright (c) 2008 GNS3 Project

=>
```

Image III.2: Example Graphical Network Simulator 3 (GNS 3)

III.4.1.1 Features overview

- Design of high quality and complex network topologies.
- Emulation of many Cisco IOS router platforms, IPS, PIX and ASA firewalls, JunOS.
- Simulation of simple Ethernet, ATM and Frame Relay switches.
- Connection of the simulated network to the real world.
- Packet capture using Wireshark.

III.4.1.2 Labs examples

There are a variety of lab examples available in the Internet, being the most famous forum 7200emu.hacki.at [74]. Here, it is possible to find “HOWTOs” tutorials and troubleshoot instructions to the most common problems.

III.4.1.3 Limitations

As most graphical interfaces, GNS3 is affected by some random bugs, but its biggest drawback is related with scalability. It was designed to simulate small scenarios. Despite the fact that it can run remote hypervisors to distribute the computing burden, it is limited in the way it deploys resources.

As previously mentioned, it is possible to optimize memory and processor usage by configuring “ghostios” , “sparemem” and idlePC, and the automated process is suboptimal by keeping the size of the simulation up to a dozen of Routers. Remote deployments, like cloud computing are not supported.

III.5 Cloud Computing

Cloud computing is a paradigm shift that follows the shift from mainframe to client server, which ocurred in the early 1980s. Details are abstracted from the users, who no longer control the technology infrastructure ("in the cloud") that supports them. Cloud computing describes a new supplement, consumption, and delivery model for IT services based on the Internet, and typically involves over-the-Internet provision of dynamically scalable and often virtualized resources.

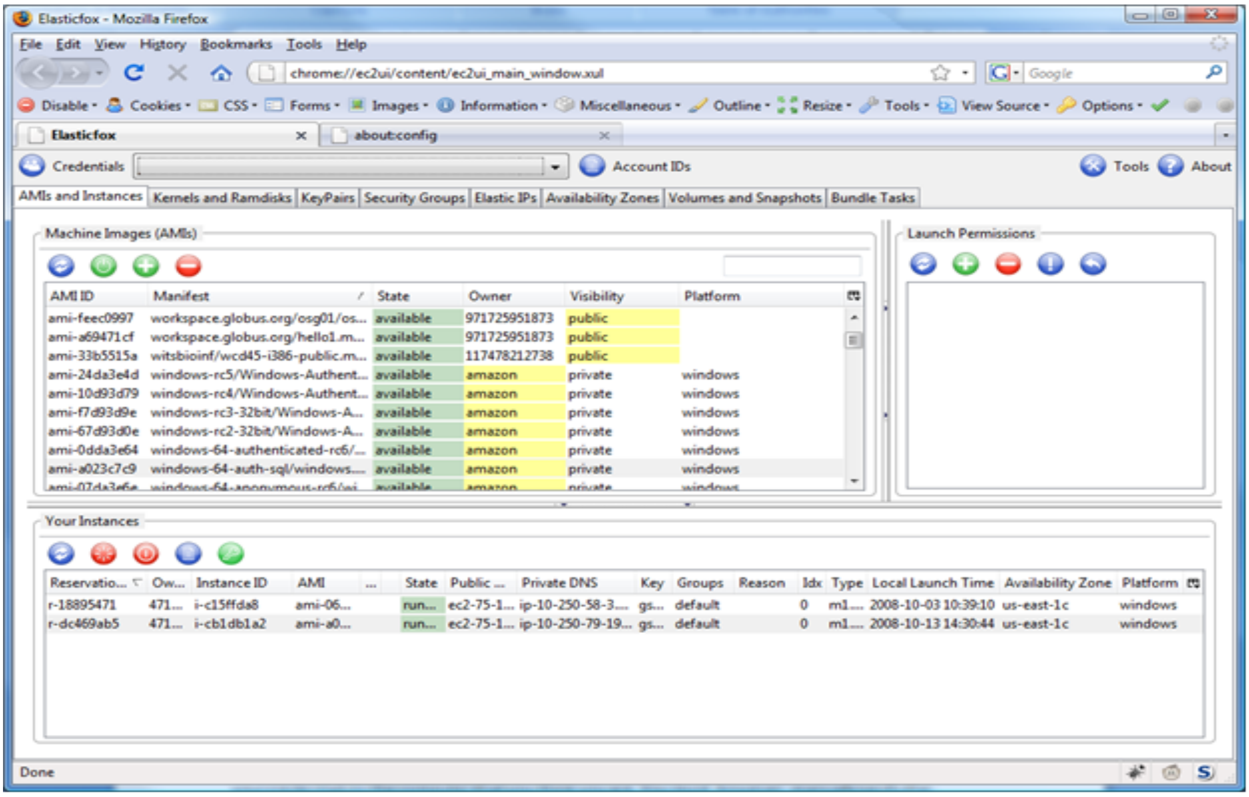
In this new ethereal environment, customers do not own the physical infrastructure, avoiding capital expenditure by renting usage from a third-party provider. They consume resources as a service and pay only for the resources they use.

Sharing "perishable and intangible" computing power among multiple tenants can improve utilization rates, as servers are not unnecessarily left idle (which can reduce costs significantly, while increasing the speed of application development). The overall computer usage rises dramatically, as customers do not have to engineer for peak load limits, using the resources to their possible limit.

III.5.1.1 Amazon EC2

Elastic Compute Cloud (Amazon EC2) [75] is a web service that provides resizable compute capacity in the cloud.

Amazon EC2 presents a true virtual computing environment, allowing the use of web service interfaces to launch instances with a variety of operating systems, load them with custom application environment. Management is easily achieved through the web console, where complete control over the virtual machines is provided.



AMI ID	Manifest	State	Owner	Visibility	Platform
ami-feec0997	workspace.globus.org/osg01/os...	available	971725951873	public	
ami-a69471cf	workspace.globus.org/hello1.m...	available	971725951873	public	
ami-33b5515a	witsbioinf/wcd45-i386-public.m...	available	117478212738	public	
ami-24da3e4d	windows-rc5/Windows-Authent...	available	amazon	private	windows
ami-10d93d79	windows-rc4/Windows-Authent...	available	amazon	private	windows
ami-f7d93d9e	windows-rc3-32bit/Windows-A...	available	amazon	private	windows
ami-67d93d0e	windows-rc2-32bit/Windows-A...	available	amazon	private	windows
ami-0dda3e64	windows-64-authenticated-rc6/...	available	amazon	private	windows
ami-a023c7c9	windows-64-auth-sql/windows....	available	amazon	private	windows
ami-07da3e6e	windows-64-anonymous-rc6/wi	available	amazon	private	windows

Reservatio...	Ow...	Instance ID	AMI	...	State	Public ...	Private DNS	Key	Groups	Reason	Idx	Type	Local Launch Time	Availability Zone	Platform
r-18895471	471...	i-c15ffda8	ami-06...		run...	ec2-75-1...	ip-10-250-58-3....	gs...	default		0	m1....	2008-10-03 10:39:10	us-east-1c	windows
r-dc469ab5	471...	i-cb1db1a2	ami-a0...		run...	ec2-75-1...	ip-10-250-79-19...	gs...	default		0	m1....	2008-10-13 14:30:44	us-east-1c	windows

Image III.3: Elastic Compute Cloud (Amazon EC2)

Few steps are needed to run and deploy a virtual machine from scratch:

- There are available, pre-configured, templated images to get up and running immediately, being also possible to create an Amazon Machine Image (AMI) containing the desired applications, libraries, data, and associated configuration settings.
- Configuration of the security and network access to the virtual server, called Amazon EC2 instance.
- There are different instance type(s) and operating systems available; then start, terminate, and monitor as many instances of necessary AMI as needed, using the web service APIs or the variety of management tools provided.
- It is possible to run in multiple locations, utilize static IP endpoints, or attach persistent block storage to the instances.
- Payment is exclusively due to resource consumption, like instance-hours or data transfer.

There are three tariffing models that can be adequate:

1. On-Demand Instances allow taxation based on compute capacity by hour, with no long-term commitments.
2. Reserved Instances allows a one-time payment for each reserved instance and in turn receive a significant discount on the hourly usage charge for that instance.

3. Spot Instances allow customers to bid on unused Amazon EC2 capacity and run those instances for as long as their bid exceeds the current Spot Price. The Spot Price changes periodically based on supply and demand, and customers whose bids meet or exceed it gain access to the available Spot Instances. Spot Instances can significantly lower the Amazon EC2 costs.

This is the perfect infrastructure to run large scenarios, as the computing power, memory and I/O of Amazon EC2 are very large.

IV. Scenario 1: MPLS - Traffic Engineering

IV.1 Introduction

Although MPLS label switching provides the underlying technologies the ability to forward packets, it does not provide all the components for Traffic Engineering (TE) support.

TE refers to the process of selecting the paths chosen by data traffic in order to facilitate efficient and reliable network operations, while simultaneously optimizing network resource utilization and traffic performance. The goal of TE is to compute a path from one node to another, complying with some constraints (e.g. bandwidth, administrative requirements), optimally tuned with respect to some scalar metric. Once the path is computed, TE assures the establishment and maintenance of the forwarding state along the path.

Three basic real-life applications for MPLS TE are:

1. Optimizing network utilization;
2. Handling unexpected congestion;
3. Handling link and node failures.

- *Optimizing network utilization* is sometimes called the strategic method of deploying MPLS-TE, or the full-mesh approach. This strategy consists of building a full mesh of MPLS TE-LSPs between a given set of routers, size those LSPs according to how much bandwidth they will expect across their interfaces and let the LSPs find the best meeting their demands. This full mesh of TE-LSPs allows to prevent congestions to some extent by spreading LSPs across the network along bandwidth-aware paths.
- *Handling unexpected congestion* is also known as the tactical approach, or deploying MPLS-TE as needed. Rather than building a full mesh of TE-LSPs between a set of routers ahead of time, the tactical approach involves letting the IGP forward traffic according to its configuration and building TE-LSPs only after congestion is discovered. This allows keeping most of the network on IGP routing only. This might be simpler than a full mesh of TE-LSPs, but allows working around network congestion as it happens. In case of a major network event (e.g. a large outage, an unexpectedly popular new web site or service, or

some other event that dramatically changes the traffic pattern) that congests some network links while leaving others empty, MPLS can deploy TE tunnels in a convenient way in order to remove some of the traffic from the congested links and put it on uncongested paths that the IGP wouldn't have chosen.

- *Handling link and node failures* is a very popular use of MPLS TE for quick recovery of those failures. MPLS TE has a component called Fast Reroute (FRR) that allows drastically reduction of packet loss in the event of link or node failure. MPLS-TE deployment requires some ground infrastructure in order to operate as explained in the following sections.

MPLS allows the elements of TE to be completely under the control of IP. This results in a network that is able to offer IP services by only overlaying a Layer 3 network on a Layer 2 network, providing the same TE benefits without needing to run a separate network and a non-scalable full mesh of interconnected routers.

MPLS TE uses RSVP to automatically establish and maintain a tunnel across the backbone. The path used by a given tunnel at any point in time is determined based on the tunnel resource requirements and network resources, such as bandwidth. Available resource information is flooded via extensions to a link-state-based IGP, such as OSPF. Tunnel paths are calculated at the tunnel head (source router) based on the required and available resources (constraint-based routing). The IGP automatically routes the traffic into these tunnels. Typically, a packet crossing the MPLS TE backbone travels on a single tunnel that connects the ingress point to the egress point.

IV.2 Objectives

In this chapter, the following topics will be addressed:

- Enabling MPLS-TE
- Information distribution
- Forwarding Packets Through the Network Core
- Forwarding Traffic Down Tunnels
  - Static routes
  - Policy routing
  - Autoroute
- FRR
  - Link Protection
  - Node Protection

- Path Protection
- Point-to-Multipoint Traffic Engineering

IV.3 MPLS-TE Core Configuration

Before starting with the configurations, it is important to mention MPLS-TE constraining requirements. Those are the features that have to be enabled in order to deploy basic TE. These requirements apply not only locally to a specific Router, but concern to all network segments that lay on the paths that are used by the TE "Eco-system" and the features that have to be enabled are:

- Cisco Express Forwarding (CEF) enabled
- Multiprotocol Label Switching protocol enabled
- A loopback interface used as MPLS-TE router ID (RID)
- TE enabled in the interfaces
- A link-state routing protocol as the routing protocol (e.g. OSPF)

Next image gives an overview of the Network layout used in this Lab. There are three functional Router types: core routers or provider routers, here designated as P Routers and Edge routers, that can be subdivided into Providers Edge PE and Customer edge CE. The detailed description of each type's role in the Network was already addressed; here, the specific relevant aspects to the respective configurations will be mentioned.

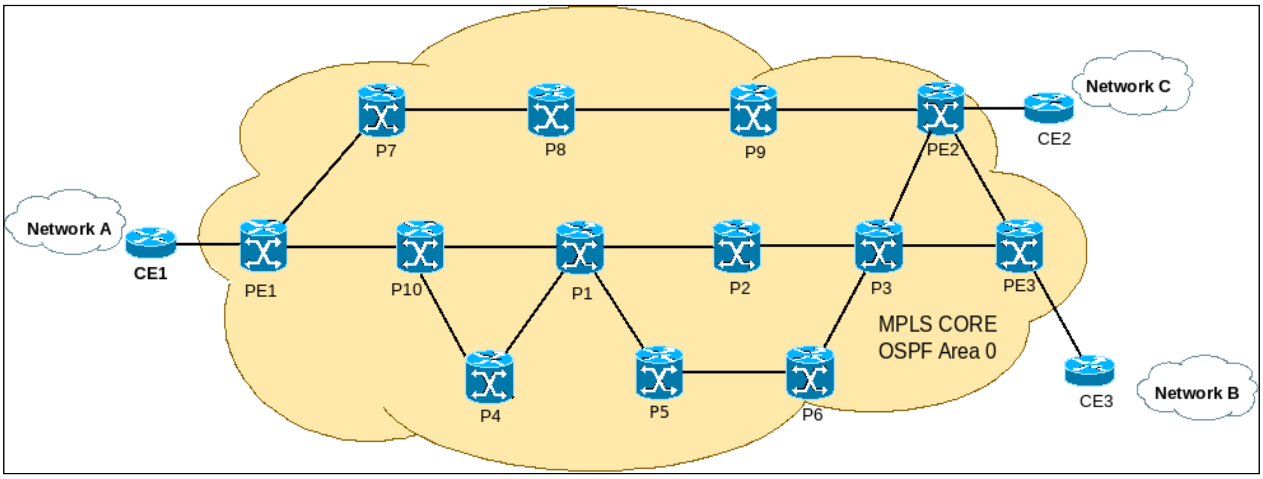


Image IV.1: MPLS TE Diagram

IV.3.1 Enabling MPLS-TE Globally

The obvious starting point is to enable the required features across the Network, so MPLS-TE may be able to function, assuming that the routers interfaces are already configured with every detail needed for basic networking. TE features must be enabled in all routers, as it is explicit in the following example of the PE1 Router configuration:

```
PE1(config)#
ip cef
mpls label protocol ldp
mpls traffic-eng tunnels
interface Loopback0
 ip address 11.11.11.11 255.255.255.255
end
```

This configuration addresses the first three points that were previously mentioned:

1. CEF is enabled;
2. The LDP MPLS-TE signaling protocol and the tunnels are enabled;
3. A loopback interface is configured with a unique IP address.

IV.3.2 Verifying MPLS TE Support

There are several commands that can be used to display information over various aspects, related to router, network, link and many other conditions that may help debugging or understanding what is going on. The following particular command is used to check if CEF is enabled.

```
PE1#show ip cef summary
IPv4 CEF is enabled and running
VRF Default:
 25 prefixes (25/0 fwd/non-fwd)
 Table id 0
 Database epoch:        0 (25 entries at this epoch)
```

Several show commands will be used in the remaining text, together with configuration examples.

IV.3.3 Configuring interfaces to forward MPLS TE traffic

Interfaces have to be instructed to forward tunnels, and several TE parameters are configured here:

```
PE1(config)#
interface Serial1/0
 description Connected to P2 s1/0
 ip address 1.3.1.1 255.255.255.252
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp resource-provider none
 ip rsvp bandwidth 1000
 serial restart-delay 0
end
```

IV.3.4 Verifying interface configuration

```
PE1#sho ip interface brief
Interface              IP-Address       OK? Method     Status      Protocol
Serial1/0              1.1.3.2          YES NVRAM      up          up
Serial1/1              1.1.2.2          YES NVRAM      up          up
Serial1/2              1.1.4.2          YES NVRAM      up          up
Serial1/3              unassigned       YES NVRAM      down        down
FastEthernet2/0        10.1.1.2         YES NVRAM      up
SSLVPN-VIF0             unassigned      NO  unset      up           up              up
Loopback0              11.11.11.11      YES NVRAM      up           up
Tunnel2                11.11.11.11      YES TFTP       up           up
Tunnel5                11.11.11.11      YES TFTP       up           up
```

This command displays summary information on the interfaces of the router, raging from the IP address to the status. Here, two tunnels are present.

IV.3.5 Configuring OSPF for MPLS-TE

The following configures an OSPF routing process for IP.

```
PE1(config)#
router ospf 100
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id Loopback0
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0
```

```
end
```

MPLS-TE is turned on for OSPF area 0. This configuration specifies that the TE router identifier for the node is the IP address associated with interface loopback0.

```
PE1#sho ip ospf mpls traffic-eng fragment

            OSPF Router with ID (11.11.11.11) (Process ID 100)

  Area 0 has 1 MPLS TE fragment. Area instance is 1.
  MPLS router address is 11.11.11.11

  Fragment 0 has 0 link. Fragment instance is 0.
  Fragment advertise MPLS router address
```

IV.4 Traffic Engineering Tunnels

MPLS TE lets you build LSPs across your network to forward traffic. MPLS TE LSPs, also called TE tunnels, let the headend of a TE tunnel control the path its traffic takes to a particular destination. This method is more flexible than forwarding traffic based only on a destination address.

Some tunnels are more important than others. For example, you may have tunnels carrying VoIP traffic and tunnels carrying data traffic that are competing for the same resources. MPLS TE allows you to have some tunnels preempting others. Each tunnel has a priority and more-important tunnels take precedence over less-important tunnels.

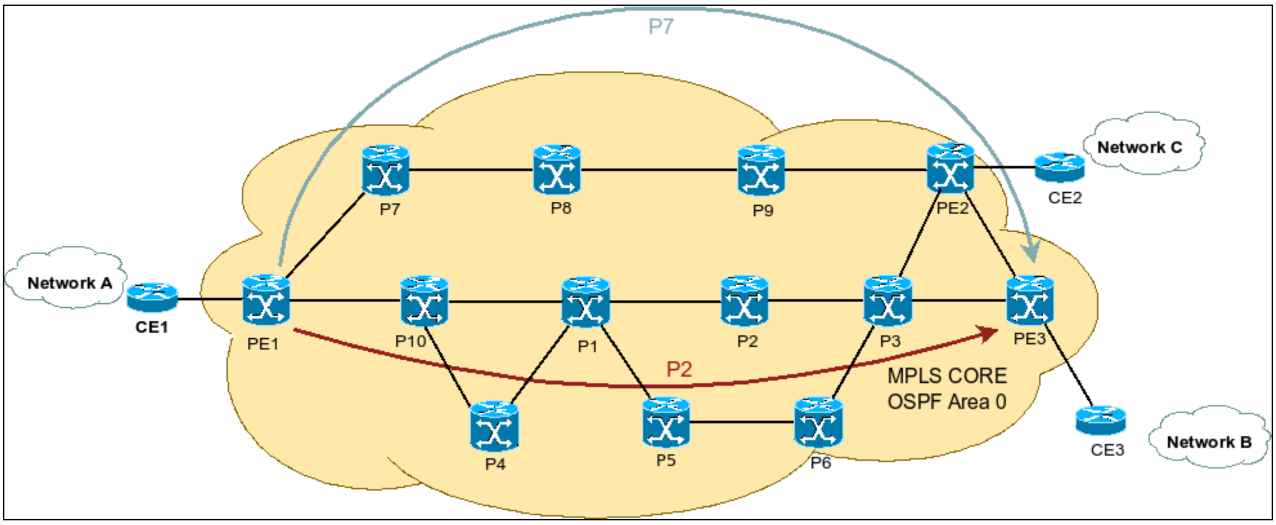


Image IV.2: TE Tunnels

IV.4.1 Configuring an MPLS-TE Tunnel

```
PE1(config)#
interface Tunnel12
 ip unnumbered Loopback0
 tunnel destination 33.33.33.33
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 7 7
 tunnel mpls traffic-eng bandwidth  100
 tunnel mpls traffic-eng path-option 10 explicit name P10123
 tunnel mpls traffic-eng path-option 20 explicit name P7892
 tunnel mpls traffic-eng path-option 50 dynamic
```

This sequence of commands correspond to the following actions:

1. Defines the name of the tunnel.
2. Gives an IP address to the tunnel; MPLS-TE tunnel interfaces should be unnumbered because they represent a unidirectional link.
3. Specifies the destination for the tunnel.
4. Sets the tunnel encapsulation mode to MPLS-TE.
5. Optional field that defines tunnel traffic priority.
6. This option defines the tunnel allocated bandwidth; it is also optional.
7. Configures the bandwidth for the MPLS-TE tunnel.
8. Defines a path-option for this tunnel; this tunnel can use a named IP explicit path or a path dynamically calculated from the TE topology database. The preference is a number from 1 to 1000, different path-option values are tried in preference order, from the lowest to the highest one.
9. It is possible to define multiple path options, if some of them fail; they are chosen sequentially from the lowest to the highest number; in this case, path-option 10 is the first, then path-option 20 and so on.
10. It is generally a good idea to let the last path option in tunnel calculation be dynamic. This is useful because this option ensures that if it is at all possible for a TE tunnel to come up, it will. Dynamic is used if an explicit path is currently unavailable and tells the router that it is supposed to calculate the best path that fits the configured tunnel constraints, such as

bandwidth and affinity bits.

IV.4.2 Creating an Explicit Path

```
PE1(config)#
ip explicit-path name P10123 enable
 next-address 10.10.10.10
 next-address 1.1.1.1
 next-address 2.2.2.2
 next-address 3.3.3.3
 next-address 33.33.33.33
```

In this submode, it is possible to add nodes to the path-option, delete and change them. The path option exp is a strict source route, it is a complete path (paths are almost always complete; incomplete paths are rare). The generation number changes every time the path option is changed in and it as to be updated whenever changes occur.

IV.4.3 Using Multiple Path Options

```
PE1(config)#
ip explicit-path name P10123 enable
 next-address 10.10.10.10
 next-address 1.1.1.1
 next-address 2.2.2.2
 next-address 3.3.3.3
 next-address 33.33.33.33
exit
ip explicit-path name P7892 enable
 next-address 7.7.7.7
 next-address 8.8.8.8
 next-address 9.9.9.9
 next-address 22.22.22.22
 next-address 33.33.33.33
end
```

IV.4.4 Verifying tunnel configurations

```
PE1#show mpls traffic-eng tunnels summary
Signalling Summary:
    LSP Tunnels Process:            running
    Passive LSP Listener:               running
    RSVP Process:             running
    Forwarding:               enabled
    Head: 4 interfaces, 4 active signalling attempts, 4 established
          4 activations, 0 deactivations
          0 SSO recovery attempts, 0 SSO recovered
    Midpoints: 0, Tails: 5
    Periodic reoptimization:            every 3600 seconds, next in 3286 seconds
    Periodic FRR Promotion:             Not Running
    Periodic auto-bw collection:        every 300 seconds, next in 286 seconds
```

This command displays the following information:

1. If the LSP tunnel process is running
2. If the RSVP process is running
3. If forwarding is enabled
4. Periodic reoptimization
5. Periodic FRR promotion
6. Periodic auto bandwidth collection
7. Role of this LSR for various LSPs (how many LSPs this LSR is headend/midpoint/tail for)

```
PE1#sho mpls traffic-eng tunnels brief
Signalling Summary:
    LSP Tunnels Process:      running
    Passive LSP Listener:               running
    RSVP Process:             running
    Forwarding:                         enabled
    Periodic reoptimization:            every 3600 seconds, next in 3384 seconds
    Periodic FRR Promotion:             Not Running
    Periodic auto-bw collection:        every 300 seconds, next in 84 seconds
TUNNEL NAME                       DESTINATION       UP IF     DOWN IF  STATE/PROT

PE1_t12                   33.33.33.33          -          Se1/0     up/up
PE1_t123                            33.33.33.33        -         Se1/0      up/up
PE1_t456                            33.33.33.33        -         Se1/0      up/up
PE1_t7892                       33.33.33.33            -         Se1/0      up/up
PE2_t332                            11.11.11.11       Se1/0     -          up/up
PE3_t12                             11.11.11.11       Se1/0     -          up/up
PE3_t123                            11.11.11.11       Se1/0     -          up/up
PE3_t456                            11.11.11.11       Se1/0     -          up/up
PE3_t7892                       11.11.11.11           Se1/0     -          up/up
```

```
Displayed 4 (of 4) heads, 0 (of 0) midpoints, 5 (of 5) tails
```

This command displays a lot of information, where the most relevant are the tunnel destination, status, and if they are happening in both directions; for example tunnel 7892 just occurs in one direction.

IV.4.5 Forwarding Traffic Down Tunnels

After TE tunnels are established, the next step is to map the traffic into the TE tunnel. There are multiple ways of mapping traffic onto TE tunnels:

- Using Static Routes
- Policy-Based Routing
- Autoroute

In the following Image, there are three tunnels P2, P3, and P8 interconnecting CE1, CE2 and CE3, where traffic is going to be forwarded to. The connection CE1 to CE2 will use static routes, CE2 to CE3 will use policy-based routing and connection CE2 to CE3 will use autoroute.

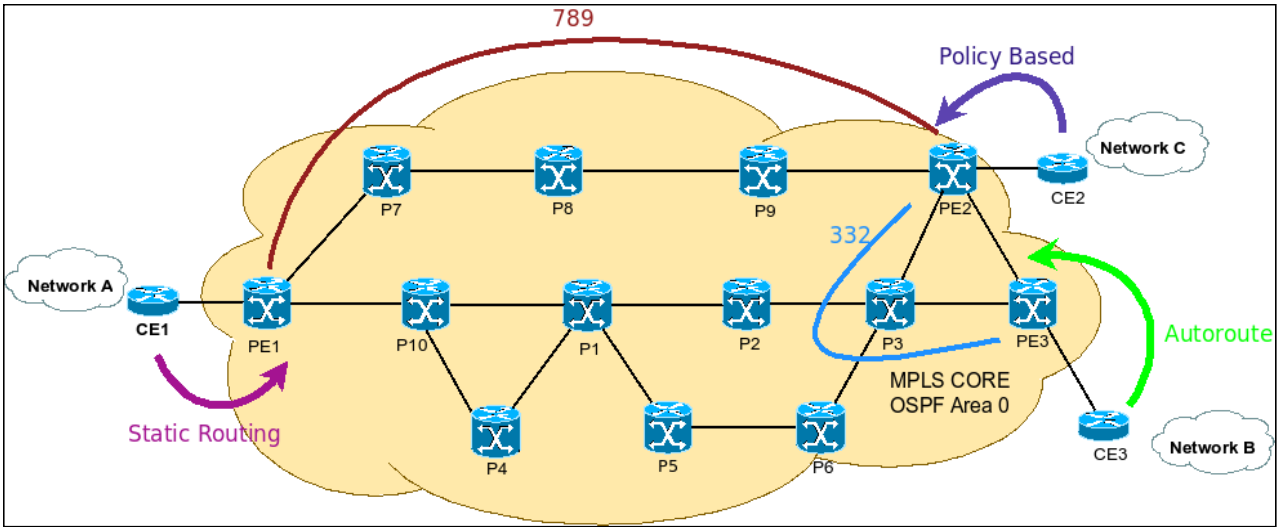


Image IV.3: Forwarding Traffic down Tunnels

IV.4.5.1 Static Routes

The simplest routing method works based on static routes. Concerning tunnels, the process is identical: forwarding traffic down an MPLS-TE tunnel using static routes is as simple as considering the tunnel as a normal point-to-point interface. The connection between CE1 and CE2 using the tunnel 789 that interconnects PE1 and PE2 will be as follows:

```
PE1#(config) ip route 13.1.1.0 255.255.255.0 Tunnel789
```

```
PE2#(config) ip route 10.1.1.0 255.255.255.0 Tunnel789
```

When the destination is the tailend of the TE tunnel, PE2 loopback0 = 22.22.22.22, the same result would be achieved by typing:

```
PE1#(config) ip route 13.1.1.0 255.255.255.0 22.22.22.22
```

A **recursive static route** is a static route that points not to an interface, but to a next hop. Anything that is possible to do with static routes can also be extended to TE tunnels, recursive static routes are and example and work normally as follows:

```
CE1#(config) ip route 13.1.1.0 255.255.255.0 10.1.1.2
```

This command advises the router to send all packets for 13.1.1.0 255.0.0.0 through the same path it would send packets destined for 10.1.1.2, keeping in mind that this route to 10.1.1.1 must be present in the routing table.

IV.4.5.2 Policy-Based Routing

With policy routing, it is possible to specify a criterion that, when matched, results in packets being mapped to a TE tunnel. Policy routing enables the creation of flexible maps for routing traffic in the network.

```
PE2#(config)
access-list 101 permit tcp any any eq echo
interface Serial1/3
 ip policy route-map exp
 route-map exp
 match ip address 101
 set interface Tunnel332
 access-list 101 permit ip any host 13.1.1.2
route-map policybased
match ip address 101
set interface Tunnel332
```

IV.4.5.3 Autoroute

In Autoroute, the TE tunnel is treated like a link connecting the head end and tail end of the tunnel and is placed into the IGP database with a metric. The IGP then thinks it has direct connectivity to the end node and routes traffic over that link (TE tunnel). However, the IGP does not advertise LSAs over this link (TE tunnel).

```
PE3#(config)
```

```
interface Tunnel12
 ip unnumbered Loopback0
 tunnel destination 11.11.11.11
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 7 7
 tunnel mpls traffic-eng bandwidth 100
```

Much of what autoroute does is invisible to the user. The only show command autoroute has is *show mpls traffic-eng autoroute*, which can optionally take the RID of a TE tail as argument.

```
PE3#sho mpls traffic-eng autoroute
MPLS TE autorouting enabled
  destination 11.11.11.11, area ospf 100  area 0, has 1 tunnels
    Tunnel12    (load balancing metric 20000000, nexthop 11.11.11.11)
                (flags: Announce)
```

IV.4.6 Load Sharing

Load sharing can be done based on the packet's source and destination IP addresses or on a per-packet basis.

Per-packet load sharing sends all packets in a round-robin fashion to the different next hops for a given route, without regard for the packet's contents. Per-packet load sharing is often undesirable, because the receiving end could receive packets out of order. For the following discussion, let us assume per-flow (also known as per-destination) load sharing.

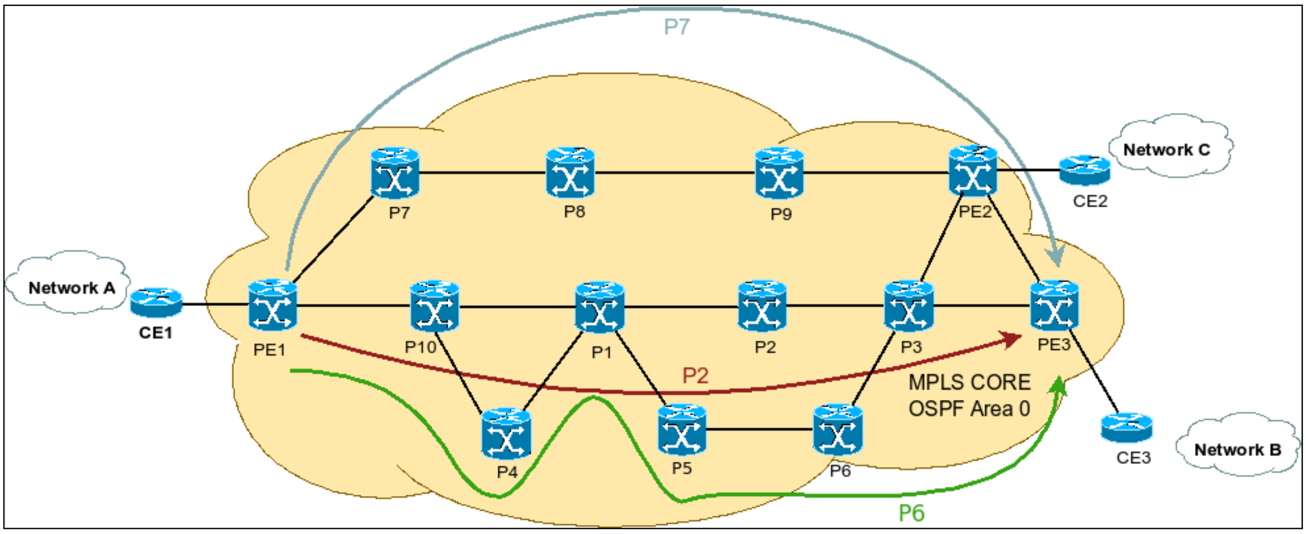


Image IV.4: Load Sharing

It is possible to configure both tunnel mpls traffic-eng bandwidth and tunnel mpls traffic- eng load-share. The configured bandwidth value is reserved through RSVP, and the configured load-share value is used for load sharing on the tunnel headend .

IV.4.6.1 Tunnel Configurations with bandwidth and load-share

```
PE1(config)#
interface Tunnel123
 ip unnumbered Loopback0
 no ip directed-broadcast
 tunnel destination 33.33.33.33
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng bandwidth 10
 tunnel mpls traffic-eng path-option 10 dynamic
!
interface Tunne456
 ip unnumbered Loopback0
 no ip directed-broadcast
 tunnel destination 33.33.33.33
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng bandwidth 30
 tunnel mpls traffic-eng path-option 30 dynamic
 tunnel mpls traffic-eng load-share 30
!
interface Tunnel7892
 ip unnumbered Loopback0
 no ip directed-broadcast
 tunnel destination 33.33.33.33
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng bandwidth 20
 tunnel mpls traffic-eng path-option 20 dynamic
 tunnel mpls traffic-eng load-share 20
```

The amount of bandwidth reserved for a tunnel is purely a control plane number. TE deals with reservations at the control plane only. No policing, specific queuing, or shaping are established on an interface as a result of a TE bandwidth reservation, as shown below:

```
PE3#show ip route 11.11.11.11
Routing entry for 11.11.11.11/32
```

```
Known via "ospf 100", distance 110, metric 321, type intra area
Last update from 11.11.11.11 on Tunnel123, 00:00:01 ago
Routing Descriptor Blocks:
* 11.11.11.11, from 11.11.11.11, 00:00:01 ago, via Tunnel123
    Route metric is 321, traffic share count is 1
  11.11.11.11, from 11.11.11.11, 00:00:01 ago, via Tunnel456
    Route metric is 321, traffic share count is 3
  11.11.11.11, from 11.11.11.11, 00:00:01 ago, via Tunnel7892
    Route metric is 321, traffic share count is 2
```

IV.5 Rerouting

Traffic-engineered networks must be able to respond to changes in network topology and maintain stability. Any link or node failure should not disrupt high-priority network services, especially the higher classes of service. TE is usually enabled in the core network, where the capacity of the links is high. If a link or a router fails, traffic is rerouted avoiding the failure.

IV.5.1 FRR

Fast Reroute (FRR) is a mechanism for protecting MPLS TE LSPs from link and node failures by locally repairing the LSPs at the point of failure, allowing data to continue to flow on them while their headend routers attempt to establish new end-to-end LSPs to replace them. FRR locally repairs the protected LSPs by rerouting them over backup tunnels that bypass failed links or node. Rerouting is a mechanism that minimizes service disruptions for traffic flows affected by an outage, and optimized rerouting re-optimizes traffic flows affected by a change in topology.

Link and node protection with TE is efficient because an idle backup link for every protected link is not needed.

Possible Local protection schemes with TE are: link protection and node protection. The two schemes have one thing in common: the repair is done as close to the point of failure as possible. Both methods provide local repair. As such, they are pretty fast and reroute the LSPs from the protected link onto the backup tunnel in tens of milliseconds.

LSPs can use backup tunnels only if they have been configured as fast reroutable. To do this, enter the following commands at the headend of each LSP.

```
P10(config)#
interface tunnel10
tunnel mpls traffic-eng fast-reroute
```

IV.5.1.1 FRR - Link Protection

With link protection, one particular link used for TE is protected. This means that all TE tunnels that are crossing this link are protected by one backup tunnel.

Backup tunnels that bypass only a single link of the LSP's path provide link protection. They protect LSPs if a link along their path fails by rerouting the LSP traffic to the next hop (bypassing the failed link). These are referred to as next-hop (NHOP) backup tunnels because they terminate at the LSP next hop that is located beyond the point of failure. The backup tunnel is an explicit path tunnel signaled by RSVP.

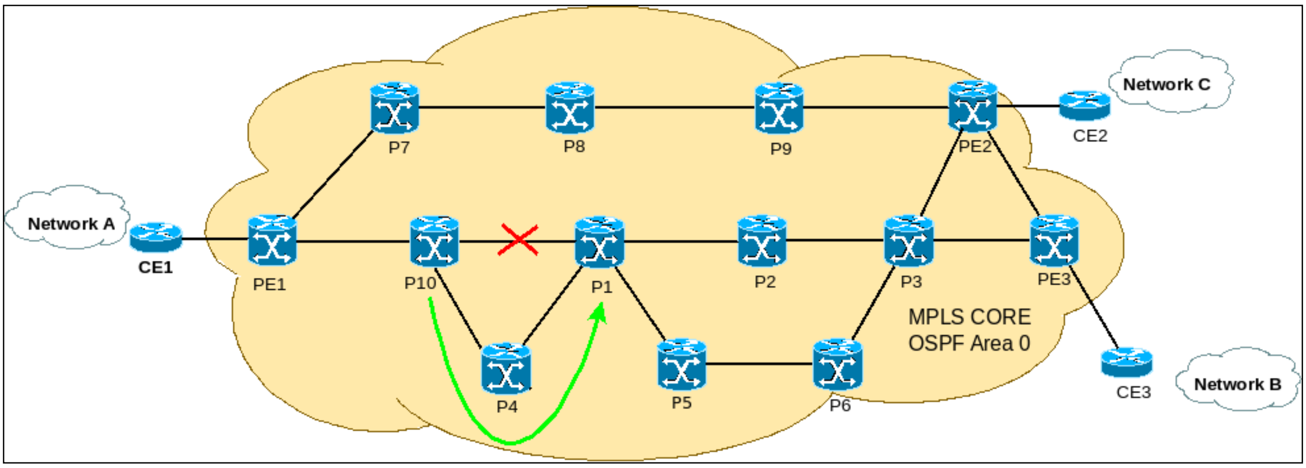


Image IV.5: Link Protection

The next commands configure a Tunnel for Fast Reroute Link Protection, configure the head-end of a primary tunnel (router P10) and assign it for Fast Reroute protection:

```
P10(config)#
interface Tunnel10
 ip unnumbered Loopback0
 tunnel destination 1.1.1.1
 tunnel mode mpls traffic-eng
tunnel mpls traffic-eng fast-reroute
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 1 1
 tunnel mpls traffic-eng bandwidth  500
 tunnel mpls traffic-eng path-option 1 dynamic
```

Establishing a Backup Tunnel around the link to protect (P10-P1) can be done using these commands:

```
P10(config)#
interface Tunnel4
 ip unnumbered Loopback0
 tunnel destination 1.1.1.1
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 2 2
 tunnel mpls traffic-eng bandwidth  500
 tunnel mpls traffic-eng path-option 1 explicit name NHOP
ip explicit-path name NHOP enable
 next-address 10.10.10.10
 next-address 4.4.4.4
 next-address 1.1.1.1
```

IV.5.2 Assigning Backup Tunnels to a Protected Interface

To assign one or more backup tunnels to a protected interface, it is necessary to tell the node that it will be the headend of the backup tunnel.

The following command output shows the status of the tunnel and the information about the parameters associated with the tunnel. In addition, it shows the preferred path chosen by the CSPF process under the explicit-path field.

```
P10#show mpls traffic-eng tunnels brief
Signalling Summary:
    LSP Tunnels Process:     running
    Passive LSP Listener:            running
    RSVP Process:                    running
    Forwarding:                      enabled
    Periodic reoptimization:        every 3600 seconds, next in 3416 seconds
    Periodic FRR Promotion:         Not Running
    Periodic auto-bw collection:    every 300 seconds, next in 116 seconds
TUNNEL NAME                      DESTINATION      UP IF     DOWN IF   STATE/PROT
P10_t0                           1.1.1.1          -         Se1/2     up/up
P10_t1                  1.1.1.1          -         Se1/2     up/up
Displayed 2 (of 2) heads, 0 (of 0) midpoints, 0 (of 0) tails
```

Then, it is possible to verify that the next hop to the destination IP address points to the tunnel interfaces in the IGP routing table. Because we have two tunnels configured on Router P10 (dynamic and explicit) with the same parameters, the traffic to destination 1.1.1.1 is *equally load balanced* among the two paths, because the bandwidths configured on the TE tunnels are the same. Traffic from P10 to P1 is *equally* load balanced across the two tunnels.

```
P10#show mpls traffic-eng tunnels tunnel 0

Name: P10_t0                          (Tunnel0) Destination: 1.1.1.1
Status:
    Admin: up         Oper: up      Path: valid       Signalling: connected
    path option 1, type dynamic (Basis for Setup, path weight 128)

  Config Parameters:
    Bandwidth: 500      kbps (Global)            Priority: 1   1    Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    AutoRoute:  enabled   LockDown:           disabled  Loadshare: 500      bw-based
    auto-bw: disabled
  Active Path Option Parameters:
    State: dynamic path option 1 is active
    BandwidthOverride: disabled  LockDown: disabled  Verbatim: disabled

  InLabel  :  -
  OutLabel : Serial1/2, 16
  RSVP Signalling Info:
       Src 10.10.10.10, Dst 1.1.1.1, Tun_Id 0, Tun_Instance 10
    RSVP Path Info:
      My Address: 1.1.4.2
      Explicit Route: 1.1.4.1 1.2.1.2 1.1.1.1
      Record   Route:   NONE
      Tspec: ave rate=500 kbits, burst=1000 bytes, peak rate=500 kbits
    RSVP Resv Info:
      Record   Route:   NONE
      Fspec: ave rate=500 kbits, burst=1000 bytes, peak rate=500 kbits
  Shortest Unconstrained Path Info:
    Path Weight: 128 (TE)
    Explicit Route: 1.1.4.1 1.2.1.2 1.1.1.1
  History:
    Tunnel:
```

```
    Time since created: 7 minutes, 23 seconds
    Time since path change: 4 minutes, 51 seconds
    Number of LSP IDs (Tun_Instances) used: 10
  Current LSP:
    Uptime: 4 minutes, 51 seconds
```

IV.5.2.1 FRR - Node Protection

The objective with node protection is not to protect only one link, but rather a whole router. Node protection works by creating a next-next-hop (NNHOP) backup tunnel. An NNHOP backup tunnel is not a tunnel to the next-hop router of the PLR, but to the router that is one hop behind the protected router. Therefore, in the case of node protection, the NNHOP router is the MP router.

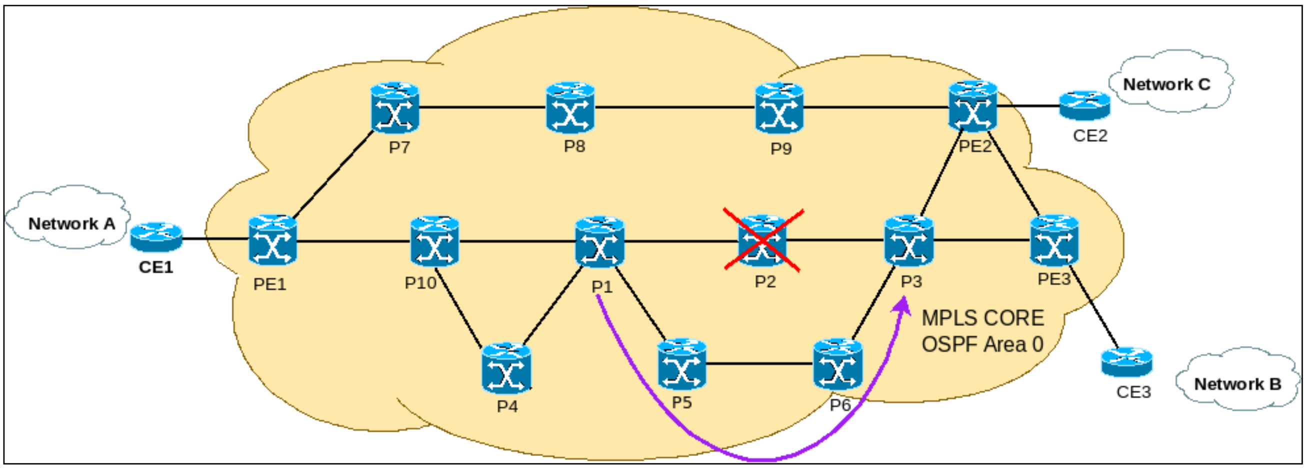


Image IV.6: Node Protection

```
P1#
interface Tunnel 2
ip unnumbered Loopback0
tunnel destination 3.3.3.3
tunnel mode mpls traffic-eng
tunnel mpls traffic-eng path-option 1 explicit name NNHOP
interface serial1/1
ip address 1.1.3.2 255.255.255.0
mpls traffic-eng tunnels
mpls traffic-eng backup-path Tunnel 56
ip rsvp bandwidth 155000
!
ip explicit-path name NNHOP enable

P1#
```

```
!
interface Tunnel 56
ip 1.1.1.1 Loopback0
tunnel destination 3.3.3.3
tunnel mode mpls traffic-eng
tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng priority 7 7
tunnel mpls traffic-eng bandwidth 1000
tunnel mpls traffic-eng path-option 10 explicit name NNHOP
tunnel mpls traffic-eng fast-reroute node-protect
!
ip explicit-path name NNHOP enable
next-address 5.5.5.5
next-address 6.6.6.6
next-address 3.3.3.3
!
 Verifying the NNHOP Tunnel:
P1#show mpls traffic-eng tunnels tunnel 56 protection
```

IV.5.3 Multiple Backup Tunnels

There are two types of multiple backup tunnels:

- Multiple backup tunnels to the same MP
- NHOP versus NNHOP backup tunnels

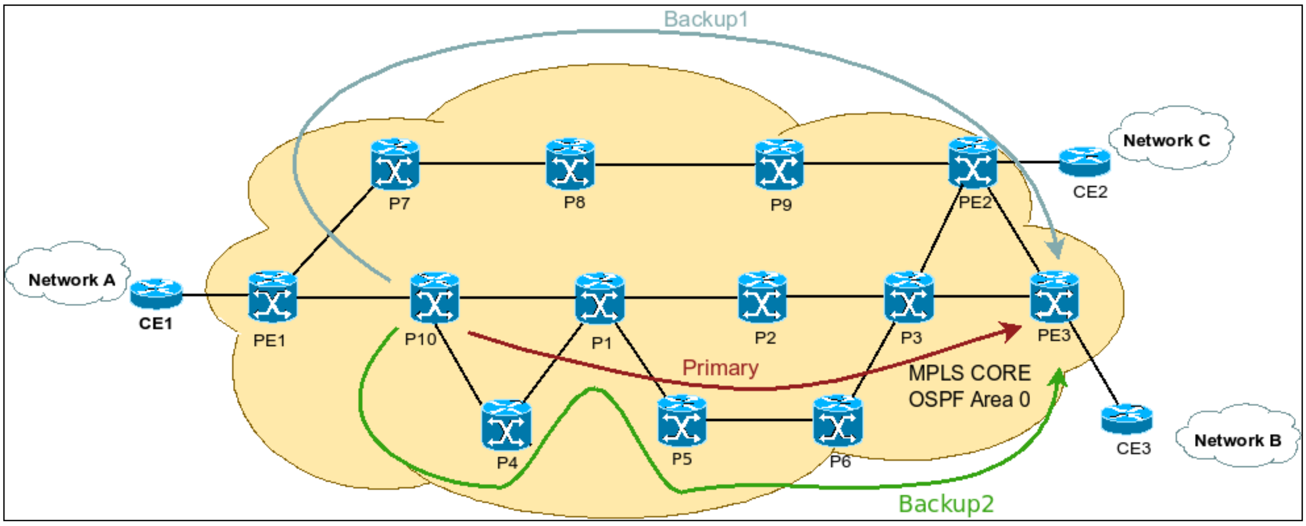


Image IV.7: Multiple Backup Tunnels

IV.5.3.1 Multiple Backup Tunnel Configuration

```
interface Tunne456
 description Link Protection Tunnel (Backup)
 ip unnumbered Loopback0
 no ip directed-broadcast
 tunnel destination 33.33.33.33
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng backup-bw global-pool unlimited
 tunnel mpls traffic-eng path-option 5 explicit name backup1
```

Adding the second and third backup paths is no different from adding the first one.

```
interface Tunnel107
 ip unnumbered Loopback0
 no ip directed-broadcast
 tunnel destination 33.33.33.33
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng backup-bw global-pool unlimited
 tunnel mpls traffic-eng path-option 5 explicit name backup2
```

IV.6 Path Protection

The MPLS Traffic Engineering (TE) Path Protection feature provides an end-to-end failure recovery mechanism (that is, full path protection) for MPLS TE tunnels. A secondary LSP is established in advance, here referred as “Protected” in the following figure, to provide failure protection for the “Primary” LSP. When a failure is detected on the primary LSP, the headend router immediately enables the secondary LSP to temporarily carry the tunnel’s traffic.

IV.6.1 Prerequisites forMPLS-TE Path Protection

- MPLS enabled, with TE configured across the intervenient routers.
- TE tunnel with a primary path option enabled by using the *tunnel mpls traffic-eng path-option* command.
- (RSVP) Graceful Restart in full mode on the routers.

IV.6.2 Restrictions for MPLS-TE Path Protection

- There can be only one secondary path for each primary path option.
- The secondary path will not be signaled with the FRR flag.

- Dynamic diverse paths are not supported.
- Link and node protection cannot be used with path protection on the headend router.
- Path protection cannot be used on an automesh tunnel template because the destinations are different and the same path option cannot be used to reach multiple destinations.

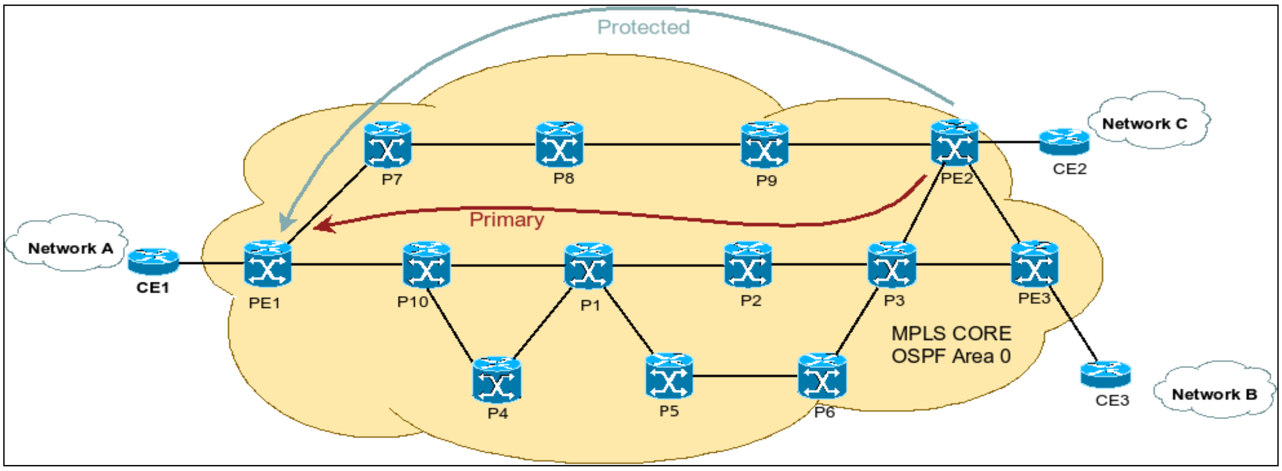


Image IV.8: Path Protection

IV.6.3 Configuring a tunnel

The following commands can be used to configure a tunnel:

```
PE2(config)#
interface Tunnel321
 ip unnumbered Loopback0
 tunnel destination 33.33.33.33
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 7 7
 tunnel mpls traffic-eng bandwidth  500
 tunnel mpls traffic-eng path-option 10 explicit name primary321
 tunnel mpls traffic-eng path-option protect 10 explicit name Protected987
```

IV.6.3.1 Configuring Explicit Paths for Secondary Paths

To specify a secondary path that does not include common links or nodes associated with the primary path, in the case those links or nodes go down, you have to configure an explicit path by performing the following steps.

```
PE2(config)#
```

```
ip explicit-path name primary321  enable
 next-address 1.1.1.1
 next-address 2.2.2.2
 next-address 3.3.3.3
 next-address 33.33.33.33
exit
ip explicit-path name Protected987  enable
 next-address 9.9.9.9
 next-address 8.8.8.8
 next-address 7.7.7.7
 next-address 11.11.11.11
end
```

IV.6.3.2 Verifying the Configuration Path Protection

```
PE1# show running interface tunnel tunnel primaryP321
```

This command shows the configuration of the primary path and protection path options.

```
PE1#show mpls traffic-eng tunnels primaryP321
```

This command shows tunnel path information.

```
PE1#show mpls traffic-eng tunnels tunnel primaryP321 protection
```

This command shows that the primary LSP is up, and the secondary LSP is up and providing protection.

```
PE1#show ip rsvp high-availability database
```

This command displays the contents of the RSVP high availability (HA) read and write databases used in TE.

IV.6.3.3 Enhanced Path Protection

Enhanced path protection provides support for multiple backup path options per primary path option. It is possible to configure up to eight backup path options for a given primary path option. Only one of the configured backup path options is actively signaled at any time.

IV.7 MPLS Point-to-Multipoint Traffic Engineering

The MPLS Point-to-Multipoint Traffic Engineering feature enables forwarding MPLS traffic from one source to multiple destinations. Traffic loss varies by platform. The merged reservation continues upstream towards the head end node, thus creating a multipoint tree from the head end to the tail ends.

A P2MP TE network contains the following elements:

- The headend router, also called the source or ingress router, is where the LSP is initiated.
- The midpoint router is where the sub-LSP signaling is processed. The midpoint router can be a branch point.
- The tailend router, also called the destination, egress, or leaf-node router, is where sub-LSP signaling ends.
- A bud router is a midpoint and tailend router at the same time.
- A P2MP tunnel consists of one or more sub-LSPs. All sub-LSPs belonging to the same P2MP tunnel employ the same constraints, protection policies and so on, which are configured at the headend router.

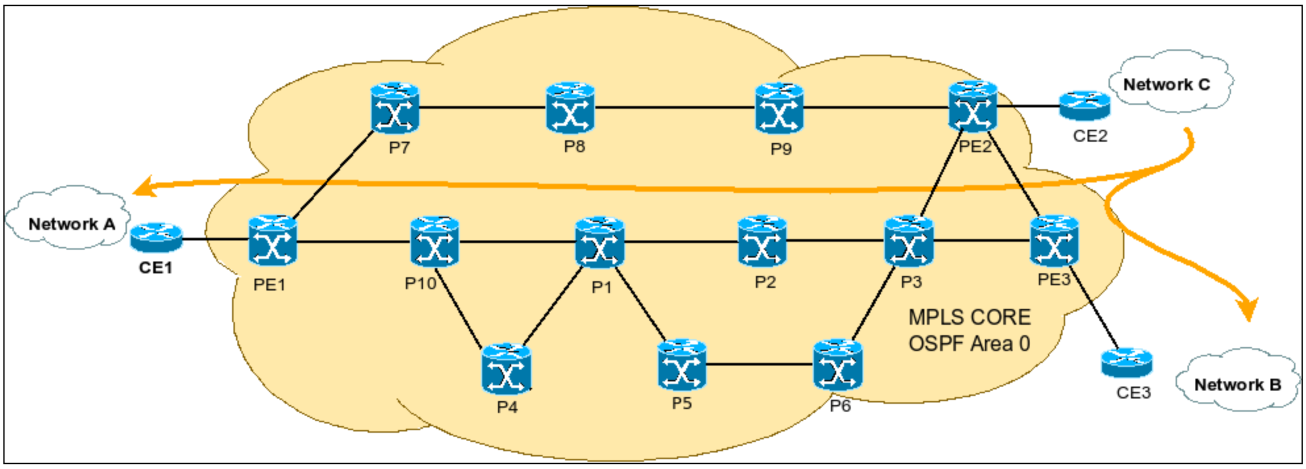


Image IV.9: Point to Multipoint TE

IV.7.1 Prerequisites for MPLS Point-to-Multipoint Traffic Engineering

- RSVP and TE features must be configured on the headend, midpoint, and tailend routers in the MPLS network.
- All routers must be configured for RSVP graceful restart.

IV.7.2 Restrictions for MPLS Point-to-Multipoint Traffic Engineering

- The P2MP TE feature is not supported in inter-area and autonomous system networks. All P2MP TE sub-label switched paths (LSPs) must originate and terminate in the same IGP and autonomous system domain.
- For P2MP sub-LSPs, only link protection, not node and path protection are supported.

- The P2MP TE feature does not support Protocol Independent Multicast (PIM) sparse mode. Only PIM source-specific multicast (SSM) is supported.
- Destinations must manually be added and removed at the headend router, dynamic adding and removal of destinations is not supported.

IV.7.3 Configuring the Headend Routers

As part of the configuration, tailend routers must be specified. Explicit paths can be used for the tunnels, the paths can be dynamically created or one can have a combination of dynamic and explicit paths. Because the configuration of the P2MP TE tunnels is done at the headend router, this feature works better in situations where destinations do not change frequently.

```
PE2(config)#
mpls traffic-eng destination list name P2MP
ip 11.11.11.11 path-option 10 dynamic
ip 33.33.33.33 path-option 10 dynamic
!
mpls traffic-eng destination list name P2MP
ip 11.11.11.11 path-option 10 explicit identifier 101
ip 33.33.33.33 path-option 10 explicit identifier 102
!
multilink bundle-name authenticated
!
interface Tunnel1111
description PE2->PE1-PE3
ip unnumbered Loopback0
ip pim passive
ip igmp static-group 99.99.99.1 source 13.1.1.255
ip igmp static-group 99.99.99.2 source 13.1.1.255
tunnel mode mpls traffic-eng point-to-multipoint
tunnel destination list mpls traffic-eng name P2MP-DYN-DST-LIST
tunnel mpls traffic-eng priority 7 7
tunnel mpls traffic-eng bandwidth 10000
!
interface Tunnel2222
description PE5->PE1-PE3
ip unnumbered Loopback0
ip pim passive
ip igmp static-group 99.99.99.3 source 13.1.1.255
ip igmp static-group 99.99.99.4 source 13.1.1.255
tunnel mode mpls traffic-eng point-to-multipoint
tunnel destination list mpls traffic-eng name P2MP
tunnel mpls traffic-eng priority 7 7
tunnel mpls traffic-eng bandwidth 20000
tunnel mpls traffic-eng fast-reroute
!
ip explicit-path identifier 101 enable
next-address 9.9.9.9
next-address 8.8.8.8
next-address 7.7.7.7
next-address 11.11.11.11
!
ip explicit-path identifier 102 enable
next-address 3.3.3.3
next-address 33.33.33.33
```

V. Scenario 2: MPLS - Virtual Private Network

V.1 Introduction

An MPLS-VPN implementation is very similar to a dedicated router peer-to-peer model implementation. From a CE router's perspective, only IPv4 updates, as well as data, are forwarded to the PE router. The CE router does not need any specific configuration to enable it to become part of a MPLS-VPN domain. The only requirement on the CE router is a routing protocol (or a static/default route) that enables the router to exchange IPv4 routing information with the connected PE router.

In the MPLS-VPN implementation, the PE router performs multiple functions: first, it must be capable of isolating customer traffic if more than one customer is connected to the PE router. Each customer, therefore, is assigned an independent routing table similar to a dedicated PE router in the initial peer-to-peer discussion. Routing across the SP (Service Provider) backbone is performed using a routing process in the global routing table. P routers provide label switching between PE routers and are unaware of VPN routes. CE routers in the customer network are not aware of the P routers and, thus, the internal topology of the SP network is transparent to the customer.

The P routers are only responsible for label switching of the packets. They do not carry VPN routes and do not participate in MPLS-VPN routing. The PE routers exchange IPv4 routes with connected CE routers using individual routing protocol contexts. To enable scaling the network to a large number of customer VPNs, multiprotocol BGP is configured between PE routers to carry customer routes.

V.2 Objectives

The objective of this chapter is to address the following topics:

- Provisioning VPN services across the MPLS/VPN backbone
- Configuration of VRFs
- Provider Edge (PE) to Customer Edge (CE) Connectivity
  - Static Routing
  - RIP Version
  - OSPF

  - EIGRP
  - eBGP
- Advanced MPLS VPN Topologies
  - Full-mesh logical topology
  - MPLS/VPN Hub-and-spoke Topology
  - Overlapping VPNs

V.3 *MPLS VPN Basic Configuration*

In this section, the generic configurations required on the routers in the service provider domain to implement MPLS VPN are addressed, as well as the configurations of the PE and P routers. It is assumed that the MPLS backbone is already configured to provide MPLS-related services to prospective customers.

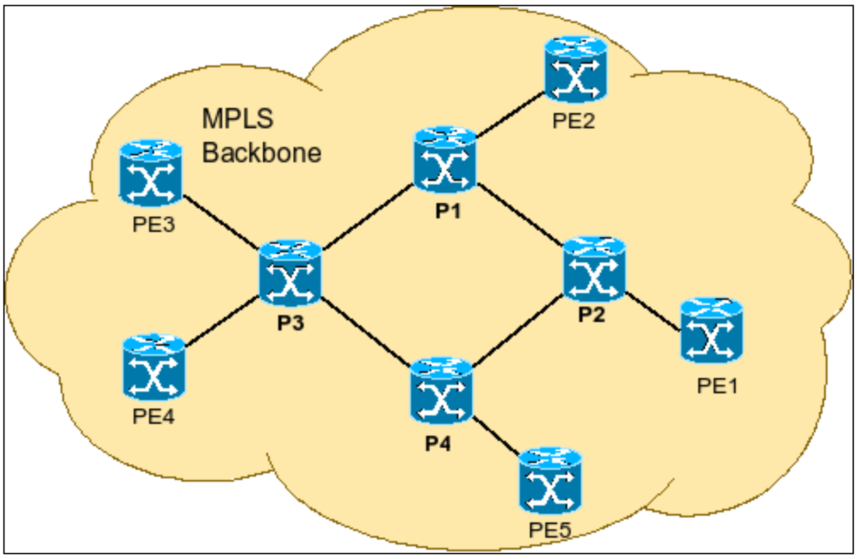


Image V.1: MPLS VPN Basic Configuration

V.3.1 Configuring MPLS Forwarding and VRF Definition on PE Routers

Configuring MPLS forwarding is the first step to provision the service provider's MPLS-VPN backbone. This step ensures the service provider's readiness to provide MPLS-related services to prospective customers. At a minimum, the steps to configure MPLS forwarding on PE routers are:

- **Step 1:** Configuring VRF on PE routers. This results in the creation of a VRF routing table and a Cisco Express Forwarding (CEF) table for the Customer.

```
PE4(config)#ip vrf Net4
```

- **Step 2:** Configuring the RD. The RD creates routing and forwarding tables. The RD is added to the beginning of the customer's IPv4 prefixes to convert them into globally unique VPNv4 prefixes.

  ```
  PE4(config-vrf)#rd 100:140
  ```

  The RD can be used in either of these formats:

  - 16-bit AS number: 32-bit number (for example, 1:100)
  - 32-bit IP address: 16-bit number (for example, 10.10.10.101:1)

  RD has to be unique for that particular VRF. Two VRFs on the same router cannot have similar RD.

- **Step 4:** Associating virtual routing/forwarding instance (VRF) with an interface or subinterface in this Customer. Associating the VRF to an interface results in the removal of the IP address from that interface. This only occurs if the VRF was associated to an interface that had the IP address already configured. This means that the IP address will have to be reconfigured after the VRF is associated with that interface.

  ```
  PE4(config)#interface Fa1/1
  PE4(config-if)#ip add 38.1.1.2 255.255.255.0
  PE4(config-if)# ip vrf forwarding Net4
  ```

V.3.2 Verification of VRF Configuration on PE Routers

The *show ip vrf* command is used to verify if the correct VRF exists on the interface.

```
PE1-AS1#show ip vrf
Name Default RD Interfaces
Customer 100:140 Fa1/1
```

The *show ip vrf interfaces* command provides the list of interfaces that are activated for a particular VRF.

```
PE1#show ip vrf interfaces
Interface IP-Address VRF Protocol
Fa1/0 38.1.1.2 Customer up
Fa1/1 37.1.1.2 Customer up
```

V.3.3 Configuration of BGP PE-PE Routing on PE Routers

Configuring BGP PE-PE routing between the PE routers is the next step in an MPLS-VPN deployment. The purpose of this step is to ensure that VPNv4 routes can be transported across the service provider backbone using MP-iBGP. The P router is transparent to this entire process and, therefore, does not carry any customer routes.

- **Step 1:** Configure BGP routing on PE routers. Enable BGP routing and identify the AS on the PE4 and PE5 routers.

```
PE4(config)#router bgp 4
PE5(config)#router bgp 4
```

- **Step 2:** Configure the MP-iBGP neighbors. Configure the remote MP-iBGP neighbor and use the loopback interface as the source of BGP messages and updates. The *update-source* command has to be used only when the neighbor is peering to your loopback address. This is irrespective of whether it is an iBGP or eBGP neighbor.

```
PE4(config-router)#neighbor 55.55.55.55 remote-as 4
PE4(config-router)#neighbor 55.55.55.55 update-source loopback0
PE5(config-router)#neighbor 44.44.44.44 remote-as 4
PE5(config-router)#neighbor 44.44.44.44 update-source loopback0
```

- **Step 3:** Configuring the address family for VPNv4 under the BGP configuration process. This step allows to enter the VPNv4 address family to activate the VPNv4 neighbors. The iBGP neighbor has to be activated, which is essential for transporting VPNv4 prefixes across the service provider backbone. Using next-hop-self is optional, being primarily used when the service provider has an eBGP PE-CE routing with the customers, because internal BGP (iBGP) sessions preserve the next-hop attribute learned from eBGP peers, which justifies the importance of having an internal route to the next hop. Otherwise, the BGP route would be unreachable.

```
PE4(config-router)#address-family vpnv4
PE4(config-router-af)# neighbor 55.55.55.55 activate
PE4(config-router-af)# neighbor 55.55.55.55 send-community extended
PE5(config-router)#address-family vpnv4
PE5(config-router-af)# neighbor 44.44.44.44 activate
PE5(config-router-af)# neighbor 44.44.44.44 send-community extended
```

- **Step 4:** Configuring the peer VRF IPv4 address family under the BGP configuration process. This step allows to enter the IPv4 networks that will be converted to VPNv4 routes in MP-BGP updates. For simplicity, redistribution of all connected networks is configured into the MP-BGP process.

```
PE4(config-router)#address-family ipv4 vrf Net4
PE4(config-router-af)# redistribute connected
PE4(config-router-af)# exit-address-family
PE5(config-router)#address-family ipv4 vrf Net4
PE5(config-router-af)# redistribute connected
PE5(config-router-af)# exit-address-family
```

V.3.3.1 Verifying BGP PE-PE Routing on PE Routers

After configuring BGP PE-PE routing between the PE routers, verifying that the MP-iBGP neighbors are operational, by issuing any of the following commands, is a crucial step:

```
show ip bgp vpnv4 * summary
show IP bgp vpnv4 all
show ip bgp summary
show ip bgp neighbor ip-address
```

V.3.4 Configuration of P Router

No special configurations need to be performed on the P routers for MPLS-VPN support. Because the P routers only participate in MPLS labeled packet forwarding, the only requirements are those of an LSR in an MPLS network, namely, IGP for NLRI exchange and LDP for label assignment and distribution. CEF needs to be enabled on all interfaces configured for MPLS forwarding.

V.4 PE to CE Connectivity

V.4.1 Static PE-CE Routing

Static PE to CE routing is one of the most common routing techniques used in MPLS-VPN deployments. Static PE-CE routing is an optimal solution for sites having a single PE-CE connection or limited number of subnets in the CE network or both. Static PE to CE routing also prevents the customer or the service provider from intentionally or accidentally flooding each other with false routing information. The service provider, therefore, retains control over the customer routing. However, static PE-CE routing might increase the provider operational and administrative overheads that are necessary to maintain static routes. This is because static PE-CE routing does not provide dynamic rerouting and therefore requires additional configuration for every new prefix on the PE routers and possibly on the CE router in the absence of a default route.

Static PE-CE routing involves the following:

- **On a CE router:** Configuring static routes to specific remote CE networks in the same VPN or configuring a static default route;
- **On a PE router:** Configuring a static VRF route to reach the connected CE router's networks. This static VRF route is redistributed in MP-iBGP and propagated as a VPNv4 prefix to the remote PE router.

V.4.1.1 Configuring Static PE-CE Routing

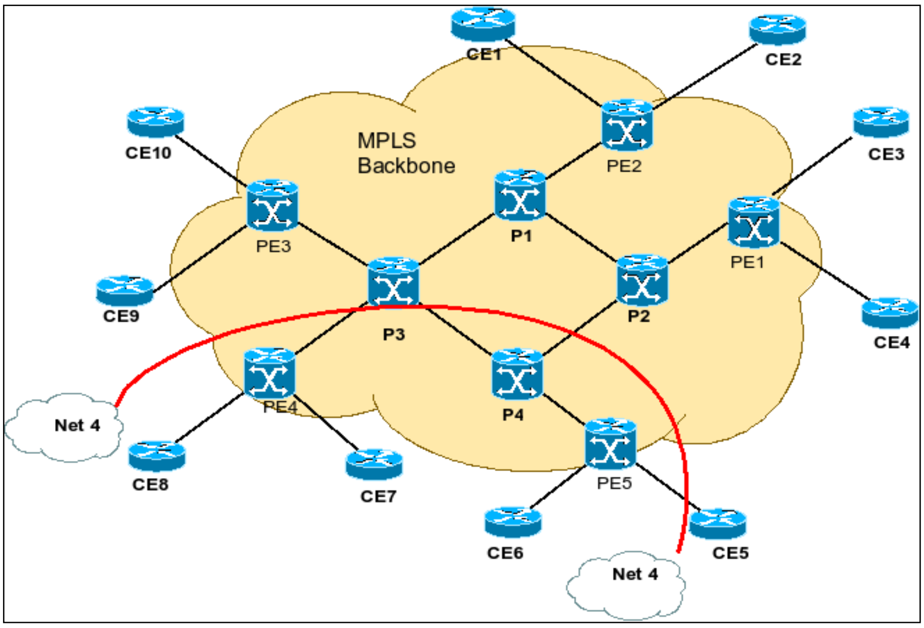


Image V.2: Static PE-CE Routing

Customer A wants to have connectivity between the Site 1 network (172.16.10.0/24) and Site 2 network (172.16.20.0/24). Site 1 and Site 2 belong to the same VPN. Site 1 and Site 2 comprise CE routers CE8 and CE5, which are connected to PE4 and PE5, respectively. A static default route is configured on CE8 and CE5 routers. The MPLS-VPN provider network plans to deploy static PE-CE routing on PE routers PE4 and PE5. The configuration related to defining VRF and its attributes on PE routers, for static PE-CE routing, is presented in the following paragraphs.

```
PE4(config)#ip vrf Net4
PE4(config-vrf)# rd 1:400
PE4(config-vrf)#route-target both 1:400
PE4(config-vrf)#interface FastEthernet1/0
PE4(config-if)# ip vrf forwarding Net4
PE4(config-if)# ip address 38.1.1.2 255.255.255.0
PE5(config)#ip vrf Net4
```

```
PE5(config-vrf)# rd 1:400
PE5(config-vrf)# route-target both 1:400
PE5(config-vrf)#interface Ethernet1/1
PE5(config-if)# ip vrf forwarding Net4
PE5(config-if)# ip address 36.1.1.2 255.255.255.0
```

Configuration of per VRF static route on PE routers

Configure per VRF static route for VRF Net4 on the PE4 and PE5 routers:

```
PE4(config)#ip route vrf Net4  192.168.8.0 255.255.255.0  38.1.1.2
PE5(config)#ip route vrf Net4 192.168.8.0 255.255.255.0  36.1.1.2
```

Configuration of IPv4 address-family and redistribution into BGP

An IPv4 address family for VRF Net4 on the PE4 and PE5 router has to be created. The per VRF static route configured in Step 1 into BGP on the PE4 and PE5 routers has to be redistributed, as well as the connected interface in BGP on PE4 to ensure that the connected interface network is known to PE5, so that it can reach the CE1-A network, 172.16.10.0. Instead of using the redistribute connected command, it is also possible to use the BGP network command to advertise the connected interface.

```
PE4(config)#router bgp 4
PE4(config-router)#address-family ipv4 vrf Net4
PE4(config-router-af)#redistribute static
PE4(config-router-af)#redistribute connected
PE5(config)#router bgp 4
PE5(config-router)#address-family ipv4 vrf Net4
PE5(config-router-af)#redistribute static
PE5(config-router-af)#network 36.1.1.2 mask 255.255.255.0
```

V.4.1.2 Verification of Static PE-CE Routing

Verification of BGP VPNv4 routing tables on PE4 and PE5

The BGP VPNv4 routing tables can be checked by issuing the *show ip bgp vpnv4 vrf VRF-STATIC* command on the PE routers.

```
PE4#show ip bgp vpnv4 vrf Net4
```

Verification of VRF routing table on PE4 and PE5

Verification of the VRF routing table, to determine if routes are received from connected CE and remote CE routers belonging to the same VPN.

```
PE4#show ip route vrf VRF-STATIC
```

Verification of End-to-end connectivity using ping

End-to-end connectivity between the CE8 and CE5 networks can be checked by issuing a ping from CE8 to network 172.16.20.0/24 on CE5 and vice versa.

```
CE8#ping 192.168.6.1
```

V.4.2 RIPv2 PE-CE Routing

Static PE-CE routing can create administrative overheads for the service provider. Service providers, therefore, prefer to run dynamic PE-CE routing protocols for the following reasons:

- To avoid maintaining multiple static routes.
- Customer prefers to run a dynamic routing protocol.
- Customer has a dual-homed connection to the service provider.

In a RIPv2 PE-CE routing environment, an IPv4 routing context is configured for each VRF running RIP on the PE router. The RIP parameters are specified in the VRF routing context. Global RIP parameters, if entered in the RIP router configuration, are inherited by the RIP VRF routing context. These parameters can, however, be overwritten in the routing context.

V.4.2.1 Configuring RIPv2 PE-CE Routing

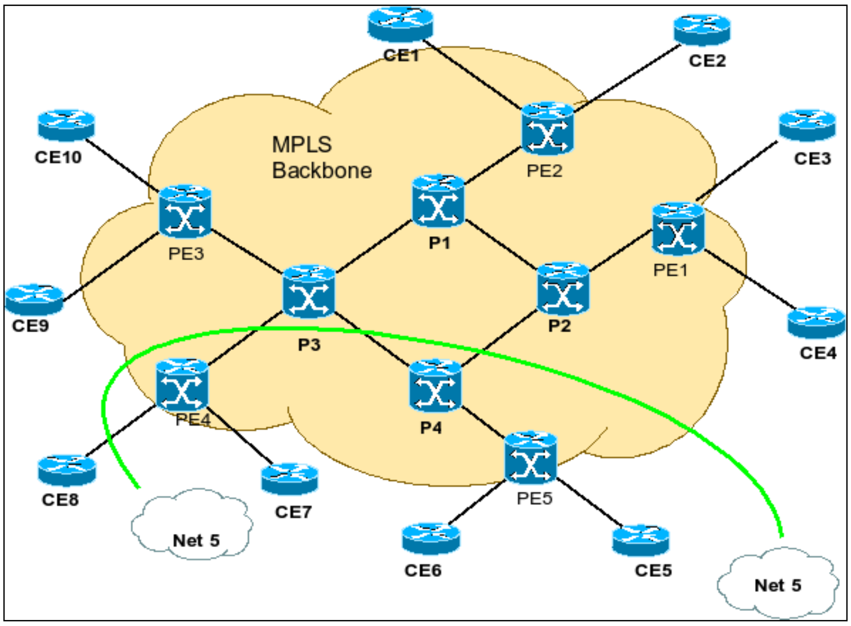


Image V.3: RIP PE-CE Routing

Net5 requires connectivity between sites with CE7 network (192.168.7.0/24) and CE5 network (192.168.50.0/24). CE5 and CE7 belong to the same VPN. CE5 and CE7 are connected to PE5 and PE4, respectively. CE5 and CE7 are already running the RIPv2 routing protocol. RIPv2 PE-CE routing protocol on PE routers PE4 and PE5 is implemented as follows.

The configuration related to defining VRF and its attributes on PE routers for RIPv2 PE-CE routing is provided by:

```
PE4(config)#ip vrf Net5
PE4(config-vrf)# rd 1:500
PE4(config-vrf)#route-target both 1:500
PE4(config-vrf)#interface FastEthernet1/1
PE4(config-if)# ip vrf forwarding Net5
PE4(config-if)# ip address 37.1.1.2 255.255.255.0
PE5(config)#ip vrf Net5
PE5(config-vrf)# rd 1:500
PE5(config-vrf)# route-target both 1:500
PE5(config-vrf)#interface Ethernet1/0
PE5(config-if)# ip vrf forwarding Net5
PE5(config-if)# ip address 35.1.1.2  255.255.255.0
```

Configuration of the RIPv2 PE-CE routing on PE routers is performed as follows:

Configuration of per VRF RIP routing context and RIP parameters on PE routers

Per VRF RIP routing context for VRF RIP under the RIP routing process on PE4 and PE5 have to be configured; the per VRF RIP parameters are configured under the address family:

```
PE4(config)#router rip
PE4(config-router)# version 2
PE4(config-router)# address-family ipv4 vrf Net5
PE4(config-router-af)#network 192.168.0.0
PE4(config-router-af)# no auto-summary
PE4(config-router-af)# exit-address-family
```

Redistribution of per VRF RIP routes in BGP

For redistribution of the per VRF RIP routes into BGP on PE routers PE4 and PE5, the following steps have to be performed:

```
PE4(config)#router bgp 5
PE4(config-router)#address-family ipv4 vrf Net5
PE4(config-router-af)#redistribute rip
```

```
PE5(config)#router bgp 5
PE5(config-router)#address-family ipv4 vrf Net5
PE5(config-router-af)#redistribute rip
```

Redistribution of MP-iBGP VPNv4 prefixes from remote PE4 into RIP per VRF routing context on PE5 router

In RIP PE-CE routing, the RIP metric is copied into the BGP multi-exit discriminator (MED) attribute. This metric can be preserved across the CE network by configuring the metric transparent option during redistribution from BGP into RIPv2, and, by doing so, it is copied back from the BGP MED attribute into the RIP version 2 metric.

```
PE4(config-router-af)#router rip
PE4(config-router)#address-family ipv4 vrf Net5
PE4(config-router-af)#redistribute bgp 5 metric transparent
PE5(config-router-af)#router rip
PE5(config-router)#address-family ipv4 vrf Net5
PE5(config-router-af)#redistribute bgp 5 metric transparent
```

V.4.2.2 Verification of RIPv2 PE-CE Routing

Verification of BGP VPNv4 routing table on PE4-AS1 and PE5-AS1

The BGP VPNv4 routing table has to be checked to see if routes are received properly:

```
PE4#show ip bgp vpnv4 vrf Net5
```

Verification of VRF routing table on PE4 and PE5

The VRF routing table has to be ckecked to see if routes advertised by local and remote CE routers are seen in the VRF routing table:

```
PE4#show ip route vrf Net5
```

Verification of end-to-end connectivity using ping

End-to-end connectivity between the CE7 and CE6 can be checked by issuing a ping from CE7 to network 172.16.20.0/24 on CE6 and vice versa.

```
CE7#ping 192.168.5.1
```

Allocation and Distribution Verification on PE5 have to be labeled:

```
PE5#show mpls forwarding-table
```

Data Forwarding Verification on PE5, is done by:

```
PE5#show ip cef vrf Net5 192.168.5.0
```

V.4.3 OSPF PE-CE Routing

Open Shortest Path First (OSPF) PE-CE routing protocol support was developed for service providers offering MPLS-VPN services to customers who have deployed OSPF as their intra-site routing protocol.

The traditional OSPF domain is divided into backbone (Area 0) and non-backbone areas, where non-backbone areas are connected to Area 0.

To circumvent the issues posed by the traditional OSPF routing model, the MPLS-VPN architecture for OSPF PE-CE routing was expanded to allow transparent customer migration from traditional OSPF routing to the MPLS-VPN routing model by introducing another backbone above the OSPF Area 0. This backbone is called the OSPF or MPLS-VPN superbackbone.

OSPF route propagation in an MPLS-VPN environment depends on the OSPF domain ID. By default, the OSPF domain ID is equal to the process ID configured on the PE router. The domain ID is set in the VPNv4 update when the OSPF route is redistributed into MP-iBGP.

V.4.3.1 Configuring and OSPF PE-CE Routing

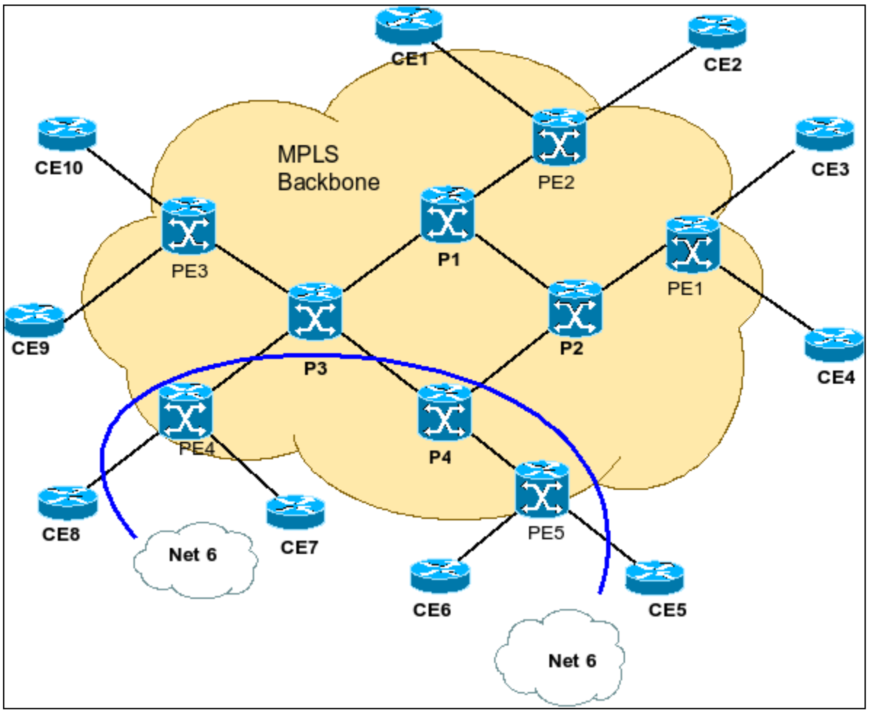


Image V.4: OSPF PE-CE Routing

Per VRF OSPF routing for VRF Customer C on PE routers PE4 and PE5 have to be enabled:

```
PE4(config)#router ospf 6 vrf Net6
PE4(config-router)# router-id 44.4.44.44
PE4(conig-router)# network 35.1.10 0.255.255.255 area 0
PE4(conig-router)# network 38.1.10 0.255.255.255 area 0
```

```
PE5(config)#router ospf 6 vrf Net6
PE5(config-router)# router-id 55.55.55.55
PE5(conig-router)# network 35.1.10 0.255.255.255 area 0
PE5(conig-router)# network 38.1.10 0.255.255.255 area 0
```

Redistribution of OSPF Routes in BGP

In this step, the OSPF routes received from the local CE routers are redistributed in MP-iBGP. It is necessary to include the match command option; otherwise, only OSPF internal routes will be redistributed in BGP.

```
PE4(config)#router bgp 6
PE4(config-router)#address-family ipv4 vrf Net6
PE4(config-router-af)#redistribute ospf 6 vrf Net6 match internal
external 1 external 2
```

```
PE5(config)#router bgp 1
PE5(config-router)#address-family ipv4 vrf Net6
PE5(config-router-af)#redistribute ospf 6 vrf Net6 match internal
external 1 external 2
```

Redistribution of MP-IBGP into OSPF

In this step, the BGP VPNv4 routes into OSPF on PE routers, PE1-AS1 and PE2-AS1, are redistributed. It must be ensured that the *subnets* keyword is included when configuring redistribution.

```
PE4(config)#router ospf 6 vrf Net6
PE4(config-router)# redistribute bgp 6 subnets
```

```
PE5(config)#router ospf 6 vrf Net6
PE5(config-router)# redistribute bgp 6 subnets
```

V.4.3.2 Verifying OSPF PE-CE Routing

Verification of OSPF neighbor adjacency

There is an OSPF neighbor relationship and adjacency is formed between the PE and CE routers:

```
PE4#show ip ospf neighbor
```

Verification of route propagation for Customer C:

```
PE5#show ip route vrf Net6 ospf 6
```

Show ip bgp vpn vrf Cust_C on PE2:

```
PE4#show ip bgp vpn vrf Net6
```

Show ip bgp vpnv4 all 172.16.20.0 on PE2:

```
PE5#show ip bgp vpnv4 all 38.1.1.0
```

V.4.4 Overlapping VPNs

MPLS can support overlapping VPNs through the RT concept. An overlapping VPN occurs when at least one CE site needs to be reachable by CEs in different VPNs. Many variations of overlapping VPNs exist.

Regardless of the goal, the RT concept allows an MPLS network to leak routes from multiple VPNs into a particular VRF. BGP supports the addition of multiple Extended Community PAs to each BGP table entry. By doing so, a single prefix can be exported with one RT.

To accomplish these design goals, each PE needs several VRFs, exporting and importing multiple RTs.

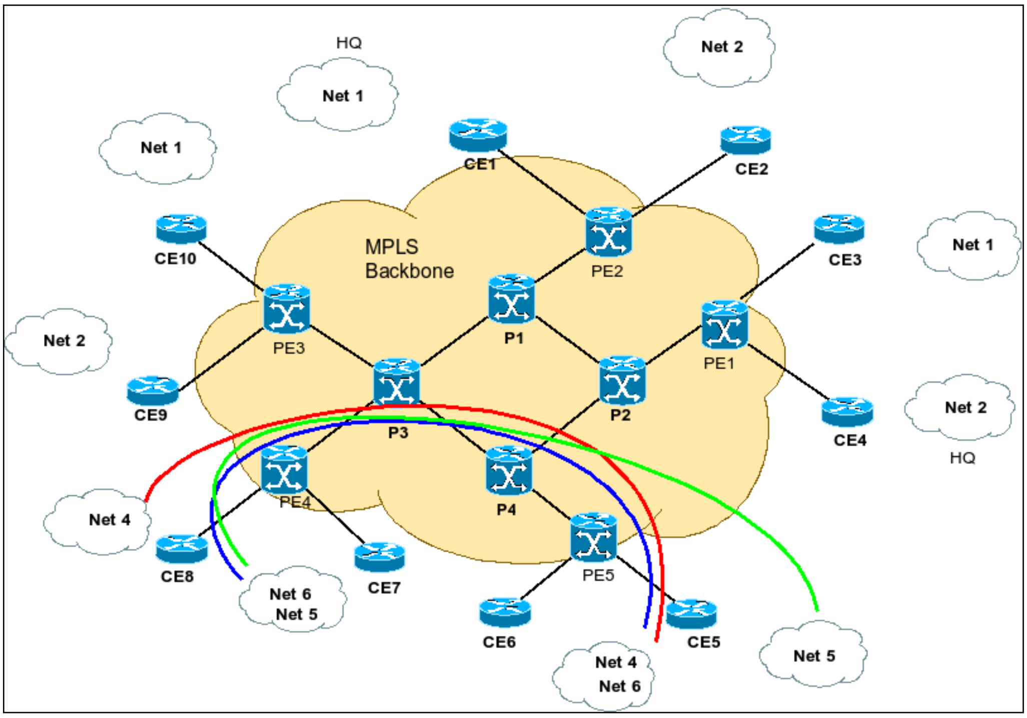


Image V.5: Overlapping VPNs

V.4.5 BGP PE-CE Routing Protocol

BGP version 4 is the current Internet standard for inter-domain (AS) exterior routing. In MPLS-VPN networks, MP-BGP is used and plays a pivotal role in the transportation of VPNv4 prefixes across the service provider network. In traditional environments, customer networks prefer to use BGP in their networks and, therefore, use BGP as a PE-CE routing protocol when migrating from a non-MPLS based to an MPLS-VPN based network. This helps the customer establish a consistent end-to-end routing policy. In an MPLS-VPN network, BGP attributes for a VPN site are transparently transported across the service provider backbone to another site in the same VPN. Because there is a single routing protocol used across the VPN between service provider core and customer sites, the concept of redistribution does not apply.

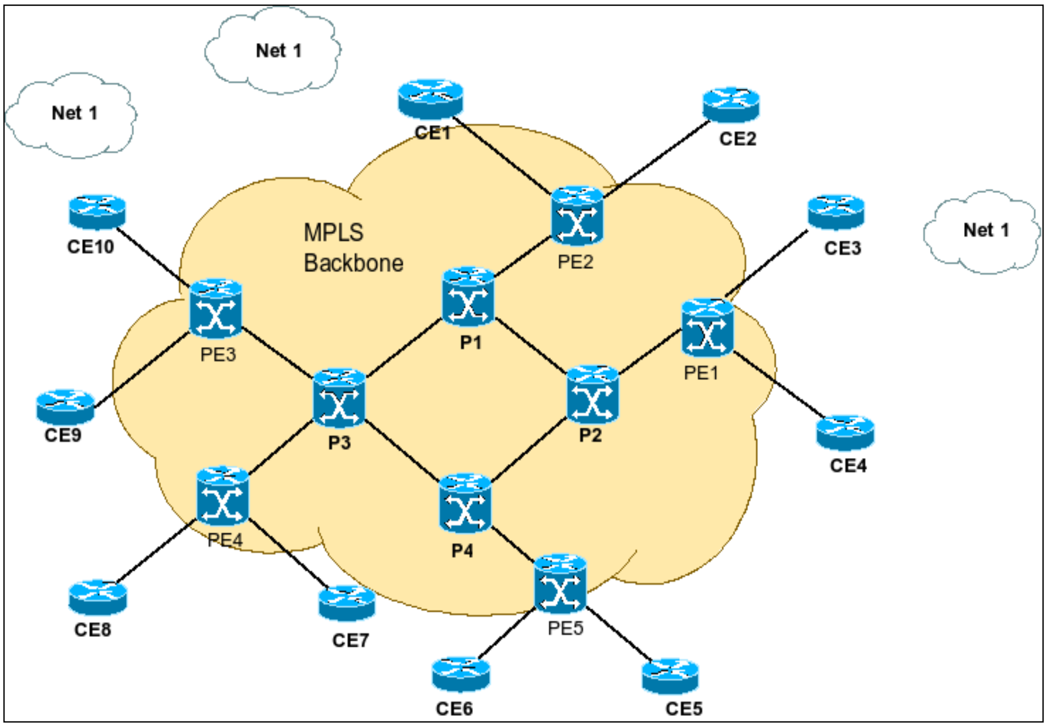


Image V.6: BGP PE-CE Routing

In this MPLS-VPN network, the service provider provisions BGP PE-CE routing protocol for Customer D. CE3 and CE10 are CE devices belonging to Net1 and to the same VPN. These CE devices are connected to the devices PE1 and PE3 in the service provider network.

V.4.5.1 Implementing BGP PE-CE Routing for VPN Sites

Definition of VRF Net1 on PE Router PE1

```
PE1(config)#
ip vrf Net1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
```

Configuration of per VRF BGP routing contexts for Net1 under the BGP routing process on PE1 - this configuration uses a router reflector P1.

```
PE1(config)#
router bgp 111
 no synchronization
 bgp log-neighbor-changes
 neighbor 1.1.1.1 remote-as 111
 neighbor 1.1.1.1 update-source Loopback0
 no auto-summary
!
 address-family vpnv4
  neighbor 1.1.1.1 activate
  neighbor 1.1.1.1 send-community both
 exit-address-family

address-family ipv4 vrf Net1
  redistribute connected
  neighbor 1.1.1.1 remote-as 111
  neighbor 1.1.1.1 activate
  maximum-paths eibgp 2 import 2
  no synchronization
 exit-address-family
```

CE13 configuration:

```
CE10(config)#
interface FastEthernet1/0
 ip address 30.1.1.1 255.255.255.0
 description Connected to PE3 f1/1
 duplex auto
 speed auto
!
router bgp 111
```

```
no synchronization
bgp log-neighbor-changes
network 0.0.0.0
network 192.168.1.128 mask 255.255.255.64
neighbor 30.1.1.2 remote-as 111
maximum-paths 2
no auto-summary
```

V.4.5.2 Verifying BGP PE-CE Routing for VPN Sites

Verification of BGP neighbor relationship between PE-CE routers:

```
PE1#show ip bgp vpnv4 all summary
```

Verification of the VRF routing table on PE1 in order to see if CE networks are received by the connected PE routers:

```
PE1#show ip route vrf Cust_A bgp
```

Verification of BGP VPNv4 routing table on CE networks on the PE routers:

```
PE1#show ip bgp vpnv4 all
```

Verification of end-to-end connectivity between CE10 and CE3

This is performed by issuing a ping from CE10 to network 192.168.1.0/25 on CE3 and vice versa.

```
CE3#ping 192.168.1.1
```

V.4.6 Implementing Route-Reflectors in MPLS-VPN Networks

BGP route-reflectors (RRs) are considered a scalability tool that allows network designers to steer away from BGP full mesh requirements. Classical iBGP split horizon rules mandate that updates received on eBGP sessions should be forwarded on all iBGP and eBGP sessions, but updates received on an iBGP session should be forwarded only on all eBGP sessions. This requires the BGP edge or boundary router (ASBR) to send updates to all other BGP-enabled routers in its own AS directly through individual iBGP sessions to each BGP router. RRs modify the iBGP split horizon rule and allow a specific router, under certain conditions, to forward all incoming iBGP updates to an outgoing iBGP session.

In this scenario, global BGP routing is globally enabled, being P1 the reflector. The following commands correspond to the configuration procedure that is necessary to enable global BGP routing, define the BGP relationship with PE routers, activate them for VPNv4 route-exchange, and configure the PE routers as clients for the route-reflection process.

```
P1(config)#
router bgp 111
 no synchronization
 bgp log-neighbor-changes
 neighbor 11.11.11.11 remote-as 111
 neighbor 11.11.11.11 update-source Loopback0
 neighbor 11.11.11.11 route-reflector-client
 neighbor 22.22.22.22 remote-as 111
 neighbor 22.22.22.22 update-source Loopback0
 neighbor 22.22.22.22 route-reflector-client
 neighbor 33.33.33.33 remote-as 111
 neighbor 33.33.33.33 update-source Loopback0
 neighbor 33.33.33.33 route-reflector-client
 no auto-summary
 !
 address-family vpnv4
  neighbor 11.11.11.11 activate
  neighbor 11.11.11.11 send-community extended
  neighbor 11.11.11.11 route-reflector-client
  neighbor 22.22.22.22 activate
  neighbor 22.22.22.22 send-community extended
  neighbor 22.22.22.22 route-reflector-client
  neighbor 133.33.33.33 activate
  neighbor 33.33.33.33 send-community extended
  neighbor 33.33.33.33 route-reflector-client
```

V.5 Advanced MPLS VPN Topologies

V.5.1 Hub-and-spoke Topology

This feature prevents local connectivity between routers at the spoke PE router and ensures that a hub site provides the connectivity. Any sites that connect to the same PE router must forward inter-site traffic using the hub site. This ensures that the routing done at the spoke site moves from the access-side interface to the network-side interface or from the network-side interface to the access-side interface, but never from the access-side interface to the access-side interface. Subscribers are prevented from directly connecting to each other, requiring the traffic always to go through the hub.

By removing the requirement of one VRF per spoke, configurations become easier. When many spokes are connected to the same PE router, configuration of VRFs for each spoke becomes complex and greatly increases memory usage, for example in large-scale environments that support high-density remote access.

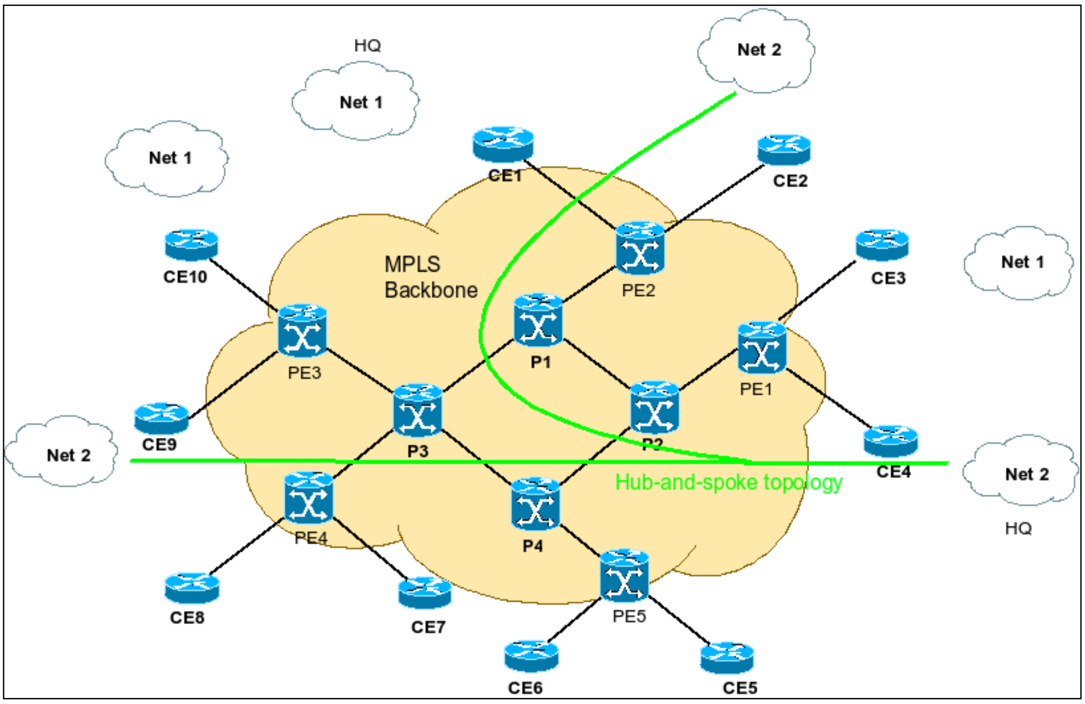


Image V.7: Hub and Spoke

V.5.1.1 Implementation

CE4 configuration

```
ip cef
interface Loopback0
 ip address 192.168.2.1 255.255.255.128
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no shutdown
 ip address 34.1.1.1 255.255.255.0
 description Connected to PE1 f1/1
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
router eigrp 222
 network 34.1.1.0 0.0.0.255
 network 192.168.2.0 0.0.0.127
 network 0.0.0.0
 distribute-list prefix LocalOnly out
 no auto-summary
```

PE1 configuration:

Configuring the Upstream and Downstream VRFs on the PE Router or the Spoke PE Router:

```
PE1(config)#
ip vrf Net2
 rd 200:10
 route-target export 200:0
 route-target import 200:1
 route-target import 200:2
!
```

Configuring OSPF enabled globally:

```
PE1(config)#
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 11.11.11.11
 log-adjacency-changes
 network 0.0.0.0 0.255.255.255 area 0
!
```

Configuring BGP responsible for redistributing the routes:

```
PE1(config)#
router bgp 111
 no synchronization
 bgp log-neighbor-changes
 neighbor 1.1.1.1 remote-as 111
 neighbor 1.1.1.1 update-source Loopback0
 no auto-summary
!
 address-family vpnv4
  neighbor 1.1.1.1 activate
  neighbor 1.1.1.1 send-community both
 exit-address-family
 !
 address-family ipv4 vrf Net2
  redistribute connected
  neighbor 1.1.1.1 remote-as 111
  neighbor 1.1.1.1 activate
  maximum-paths eibgp 2 import 2
  no synchronization
 exit-address-family
 !
```

V.5.1.2 Verifying configuration:

To verify the configuration, there are some commands available:

```
PE1#show ip vrf brief Net2
```

This command displays information about all the VRFs configured on the router.

```
PE1#show ip route vrf Net2
```

This command displays the IP routing table for the specified VRF, and information about the per-user static routes installed in the downstream VRF.

V.5.2 Full-mesh logical topology

A full mesh topology is one in which every CE connects to every other CE. Let us now present an example of such a configuration.

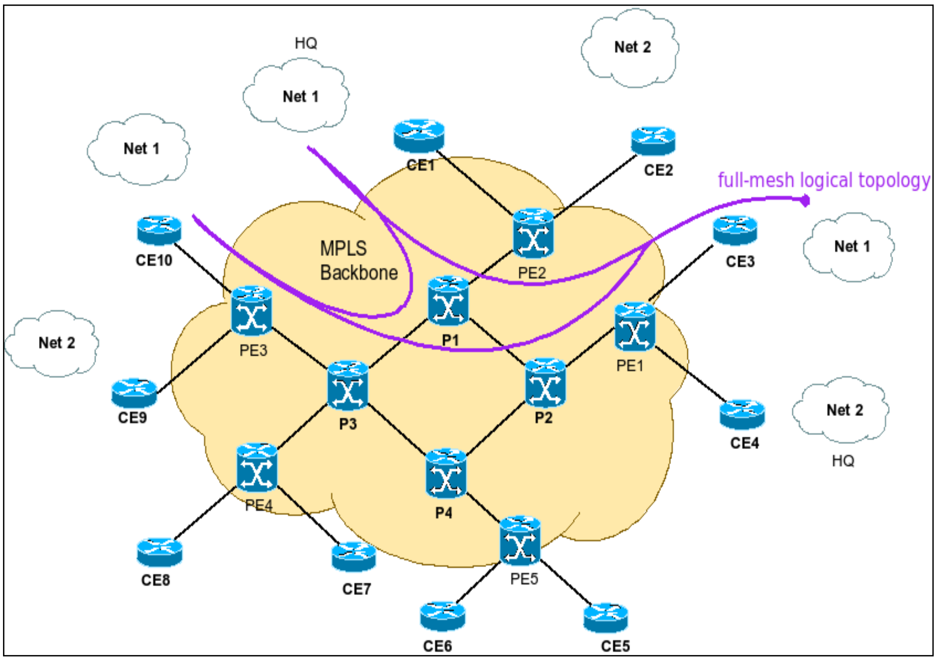


Image V.8: Full Mesh

CE1 configuration

```
ip cef
interface Loopback0
 ip address 192.168.1.1 255.255.255.128
interface Loopback1
 description Internet
 ip address 1.2.3.4 255.255.255.255
interface Fa1/0
 ip address 172.16.1.1 255.255.255.254
 half-duplex
router bgp 111
 no synchronization
 bgp log-neighbor-changes
 network 0.0.0.0
 network 192.168.1.0 mask 255.255.255.128
 neighbor 172.16.1.0 remote-as 100
 neighbor 172.16.1.3 remote-as 100
 maximum-paths 2
 no auto-summary
```

PE2 configuration

```
ip cef
ip vrf Cust1
 rd 100:10
 route-target export 100:1
 route-target import 100:1
mpls label protocol ldp
mpls traffic-eng tunnels
interface Loopback0
 ip address 10.0.2.2 255.255.255.255
interface Fa1/0
 ip vrf forwarding Cust1
 ip address 172.16.1.0 255.255.255.254
 half-duplex
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 10.0.2.2
 log-adjacency-changes
 network 10.0.0.0 0.255.255.255 area 0
router bgp 100
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.0.1.3 remote-as 100
 neighbor 10.0.1.3 update-source Loopback0
 no auto-summary
 address-family vpnv4
  neighbor 10.0.1.3 activate
  neighbor 10.0.1.3 send-community both
 exit-address-family
 address-family ipv4 vrf Cust1
  redistribute connected
  neighbor 10.0.1.3 remote-as 100
  neighbor 10.0.1.3 activate
  neighbor 172.16.1.1 remote-as 111
  neighbor 172.16.1.1 activate
  neighbor 172.16.1.1 route-map Cust1-SOO in
  maximum-paths eibgp 2 import 2
  no synchronization
 exit-address-family
```

VI. Scenario 3: MPLS - IPv6

VI.1 Introduction

Service providers (SP) will have to increasingly start supporting IPv6 traffic across their Networks. The obvious solution is to have IPv6 running on their routers. However, this approach has two disadvantages.

1. The SP needs to enable a new protocol (IPv6) and deploy it on all its routers.
2. IPv4 is not going away soon or transition to IPv6 from day to night. Therefore, IPv4 and IPv6 have to run in parallel in a dual-stack router for a long time into the future.

The solution is to use the MPLS ability to transport traffic independently of its type. More than just IPv4 as a payload, the labeled packets might be IPv6 packets, without the need for the P routers to run IPv6. The solutions 6PE and 6VPE are based on this.

All these solutions have the advantage that the P routers in the MPLS backbone do not need to run IPv6 because the P routers switch only labeled packets. As such, these solutions are more popular than directly running IPv6 across the backbone.

Another possibility to carry IPv6 over an MPLS backbone is to use the MPLS-VPN solution. In the case of MPLS-VPN, IPv4 is carried inside VPNs over the MPLS backbone. To carry IPv6 traffic over IPv4, the CE routers need tunnels between them. This means that the CE routers need to be dual-stack routers. These routers are the only routers that are running IPv6, because the PE routers only recognize IPv4 packets coming from the CE routers. In short, the advantage is that MPLS -VPN is already deployed in many of the service provider networks, and the PE (and P) routers do not need to run IPv6. The disadvantage is that the CE routers need to have tunnels configured, and an extra IPv4 header adds additional overhead.

VI.2 Objectives

This chapter aims to cover the following aspects:

- IPv6 PE (6PE)
- IPv6 VPNs (6VPE)
- Inter-AS Networks for 6PE and v6VPNs

- Carrier Supporting Carriers with BGP with 6PE and v6VPNs.

VI.3 IPv6 PE (6PE)

Cisco 6PE (IPv6 over MPLS) enabled backbones allow IPv6 domains to communicate with each other over an MPLS IPv4 core network. This implementation requires no backbone infrastructure upgrades and no reconfiguration of the core routers. The method relies on BGP extensions in the IPv4 network Provider Edge Routers (6PE) to exchange IPv6 reachability information along with an MPLS label for each IPv6 address prefix announced. 6PE routers are dual-stack (IPv4 and IPv6).

Because forwarding is based on labels rather than on the IP header itself, IPv6 packets can be tunneled into IPv4 frames that can be transported across the MPLS network.

The edge router peers with the customer routers with IPv6 and IPv4 separately. Both IPv4 and IPv6 addresses are exchanged between the edge router and customer routers using a routing protocol. The CE devices establish regular routing with the PE routers, either using static or dynamic routing protocols. The PEs peer with each other directly or by using route reflectors for the exchange of IPv6 routes in addition to IPv4 routes. The core routers learn only IPv4 addresses that are advertised in the IGP. The core routers inside the MPLS clouds are not aware that they are switching IPv6 packets, as they only use MPLS forwarding for binding IPv4 labels and an IPv4 IGP to establish internal reachability inside the MPLS cloud.

To distribute IPv6 address information between PEs, BGP is extended to carry IPv6 addresses. The address family identifier is used to distribute the IPv6 prefixes within BGP. A label associated with that IPv6 network prefix is also advertised as part of BGP extended communities, similar to the IPv4 VPN extensions. The PEs then exchange this IPv6 prefix and label information via the iBGP session between them. Just as the PEs advertise the IPv4 reachability, they also advertise IPv6 reachability with the ability to resolve the IPv6 routes to an IPv4 next hop.

The advantages of 6PE are:

- The P router does not need to run IPv6 or even be IPv6 capable. Therefore, the 6PE solution can be quickly deployed over an existing MPLS backbone.
- The IPv6 packets are directly labeled without an extra header. Because this solution involves no VPNs, no VRF interfaces exist for IPv6 on the PE routers; as such, the PE and CE routers can use any routing protocol for IPv6 in the 6PE solution. The IPv6 routing protocol does not have to be aware of VRF.

VI.3.1.1 Configuration of 6PE

The 6PE solution is simple and straightforward to configure. Configure the iBGP neighbor under the IPv6 address family of BGP and add one extra keyword (iBGP neighbor command with the send-label keyword). Of course, configuration of an IGP for IPv6 on the link between the PE and CE router is needed.

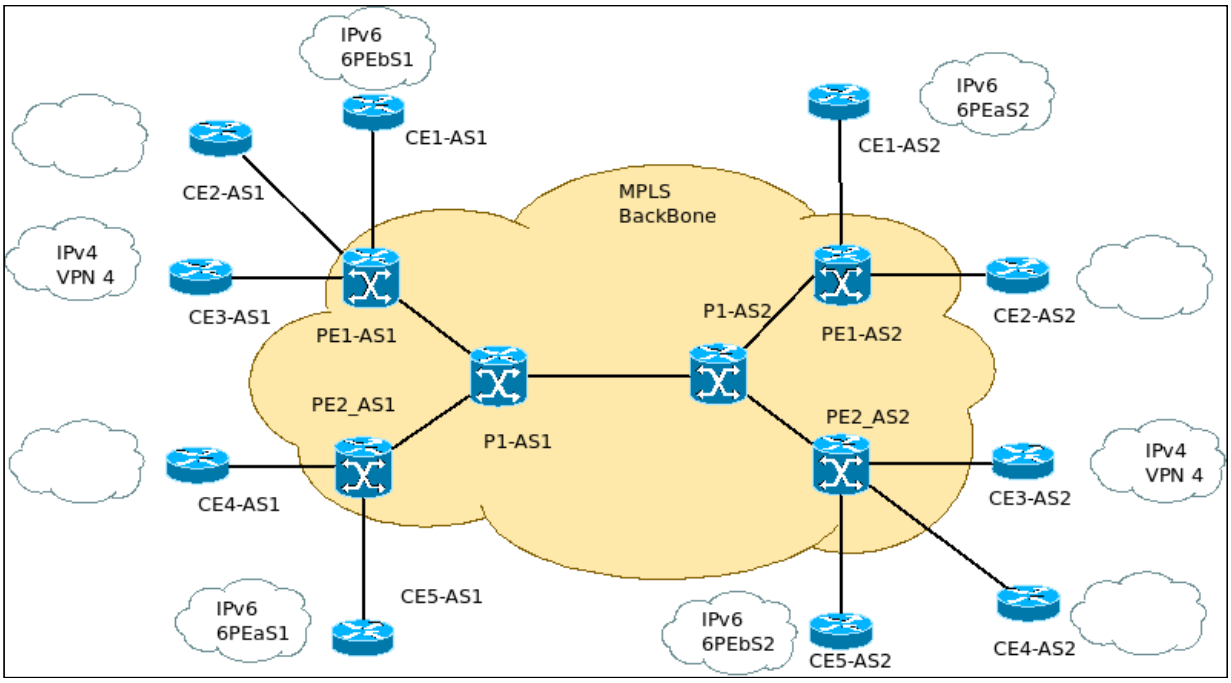


Image VI.1: IPv6 6PE

Configuration of the P Routers

There are no special configurations in the core routers, besides the usual MPLS enabled features and the fact that they will be running OSPF area0 as the IGP.

Configuration of the PE Router

Here, OSPF is configured for IGP routing, and then BGP:

```
PE1-AS2#(config)
ip cef
!
ipv6 cef

!
ipv6 unicast-routing

!
mpls ipv6 source-interface Loopback0

tag-switching tdp router-id Loopback0

!

interface Loopback0
```

```
 ip address 11.11.11.11 255.255.255.255
 ipv6 address 2001:1::1/64
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 11.11.11.11
 log-adjacency-changes
 network 0.0.0.0 0.255.255.255 area 0
router bgp 100
!
 no synchronization
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 22.22.22.22 remote-as 100
 neighbor 22.22.22.22 description to_PE2-AS2
  neighbor 22.22.22.22 update-source Loopback0
 neighbor 33.33.33.33 remote-as 100
  neighbor 33.33.33.33 description to_PE1-AS1
  neighbor 33.33.33.33 update-source Loopback0
  neighbor 44.44.44.44 remote-as 100
 neighbor 44.44.44.44 description to_PE2-AS1
 neighbor 44.44.44.44 update-source Loopback0
 !
 address-family ipv6
 neighbor 22.22.22.22  activate
 neighbor 22.22.22.22 send-label
 neighbor 33.33.33.33 activate
 neighbor 33.33.33.33 send-label
  neighbor 44.44.44.44 activate
  neighbor 44.44.44.44 send-label
 network 2001:21::/64
 exit-address-family
!
ipv6 route 2001:21::/64 interface FastEthernet0/0
```

Configuration of the CE Router

```
CE1-AS1#(config)
ipv6 unicast-routing
ipv6 cef
interface Loopback0
```

```
ip address 192.168.3.1 255.255.255.255
ipv6 address 2001:DB8:1:1::1/128
ipv6 rip cust-one enable
interface Fa1/0
no ip address
ipv6 address 2001:DB8:2::1/64
ipv6 rip cust-one enable
ipv6 router rip cust-one
```

VI.3.1.2 Verifying 6PE Operation

There are no specific commands dedicated to show 6PE functionality, but the output of the following components displays relevant information when 6PE is enabled. Monitoring the behavior of the following components when running 6PE

- MP-BGP
- MPLS
- IPv6 routing table

is performed with the following commands:

```
PE1-AS2#show bgp ipv6 2001:33::/64
```

This command displays the details of Cisco 6PE BGP routes, which the ingress PE1-AS2 learns from the egress 6PE PE2-AS1.

```
PE1-AS2#sh bgp ipv6 neighbors 192.168.0.1
```

This command displays, among other information, the capabilities of the BGP peer. Since a new capability "IPv6+label" is added, it must appear in the list of capabilities returned by this command.

```
PE1-AS2#show mpls forwarding-table
```

This command displays correspondence between labels and prefix. In the case of 6PE, labels are "aggregate label". This indicates that there are several prefixes for one local label. The command now shows "Aggregate", rather than target prefix.

```
PE1-AS2#show ipv6 route 2001:DB8:1:2::1/128
```

This command specifies the Cisco 6PE origin of the route with the "IPv6-mpls" keyword.

VI.4 IPv6 VPNs (6VPE)

The 6VPE solution smoothly introduces IPv6 VPN service in a scalable way, without any IPv6 addressing restrictions. It does not jeopardize a well-controlled service provider IPv4 backbone or any customer networks. For IPv4 VPN customers, IPv6 VPN service is exactly the same as MPLS VPN for IPv4.

The IPv6 MPLS VPN service model is similar to that of IPv4 MPLS VPNs. SPs who have already deployed MPLS IPv4 VPN services over an IPv4 backbone can deploy IPv6 MPLS VPN services over the same IPv4 backbone by upgrading the PE router IOS version and dual-stack configuration without any change on the core routers. IPv4 services can be provided in parallel with IPv6 services. A PE-CE link can be an IPv4 link, an IPv6 link, or a combination of an IPv4 and an IPv6 link. 6VPE offers the same architectural features as MPLS VPN for IPv4. It offers IPv6 VPN and uses the same components.

Route Tagging allows a PE router to assign a tag (community of interest) to any route as part of a BGP attribute. The tag is called a community. By assigning route tags, upon import of routes, a PE can then filter the routes to exclude those that do not belong to its community of interest. This prevents routes from being populated in the PEs where these routes are not needed; it also prevents network reachability.

Virtual Routing and Forwading with IPv6, another model for IPv6 VPNs, consists of using the similar virtual routing and forwarding (VRF) structures, used in IPv4, and separate the IPv6 routes on a per-VRF basis. This separation of routes is similar to that of the IPv4 model.

CEs are connected to the PEs and are placed in a VRF. The PEs have a separate routing and forwarding table per VRF/VPN, and IPv6 routes from the CE are populated in these VRFs. The peering model is similar to VPNv4, in which PEs exchange IPv6 routes in iBGP using the IPv6 address family extension to the MP-BGP. Labels are then distributed in the same way as the IPv4 prefixes are. All the functionality to separate one v6 address from another using route distinguishers and route targets apply here in the same manner. Route targets can be imported or exported in the same manner as IPv4 routes to create intranets and extranets.

IPv4 L3VPNs can coexist with IPv6 L3VPNs on the same PE and in the same VRF table. A single VRF table can be used to store IPv6 and IPv4 addresses belonging to the VPN. The CEs advertise IPv4 and IPv6 addresses, while the PE independently advertises IPv4 and IPv6 addresses with route targets. Allowing the coexistence of IPv4 and IPv6 L3VPNs provides a smooth migration of networks from IPv4 to IPv6. It also enables the coexistence of services regardless of the IP protocol version.

6VPE has the following key features:

- It has an MPLS core network running an IPv4 routing protocol (IGP) and a label distribution protocol (LDP or RSVP for TE).
- The edge LSRs or PE routers are capable of running IPv6.
- The edge LSRs or PE routers have VRFs that designate the VPNs toward the customer or CE routers.
- A full mesh of MP-iBGP sessions exists between the edge LSRs or PE routers and serves to distribute the IPv6 VPN prefixes and their associated labels. (These IPv6 VPN prefixes are called vpnv6 prefixes.) This is also known as vpnv6 + label. The label encoding is done according to RFC 3107.
- The IPv6 packets are transported across the MPLS network with two labels: an IGP label as the top label and a BGP (or VPN) label as the bottom label.
- The PE and CE routers have an IPv6 routing protocol between them.

VI.4.1.1 Configuration of 6VPE

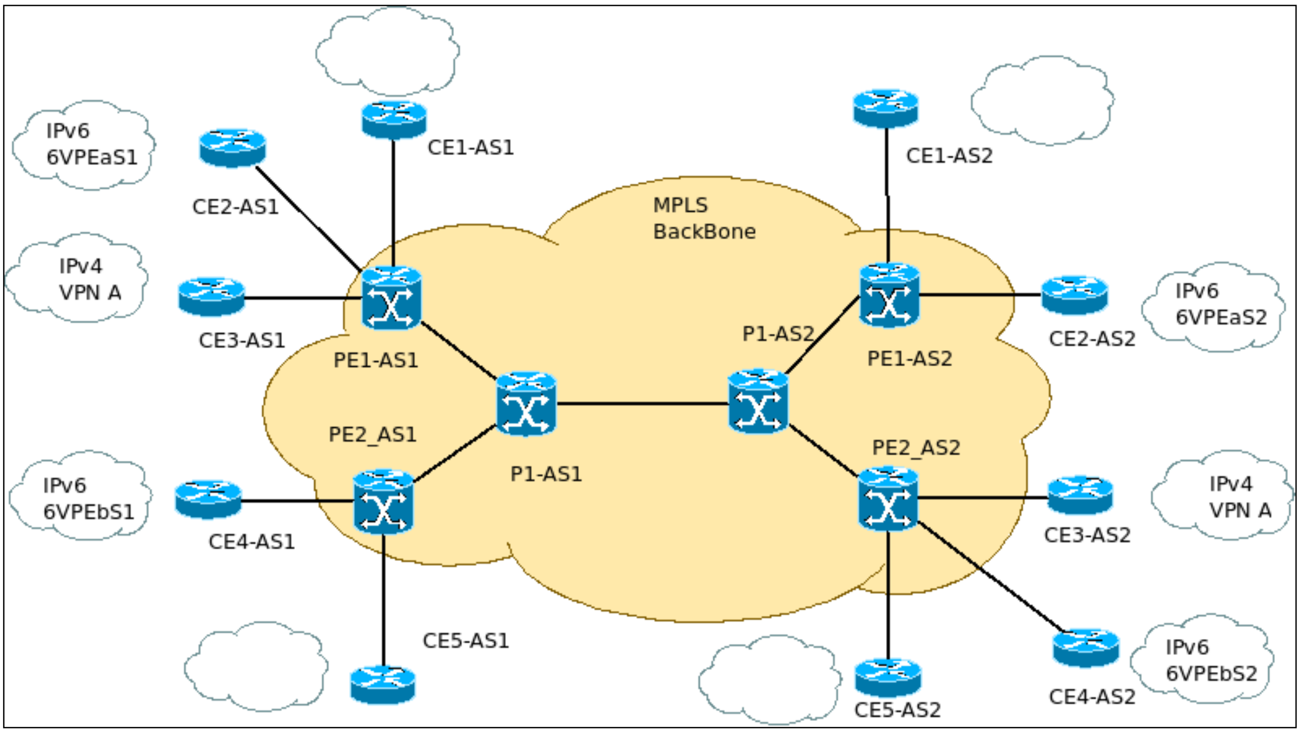


Image VI.2: IPv6 6VPE

6VPE is similar in operation to MPLS-VPN for IPv4, so similar commands are used, but now for IPv6 instead of IPv4. These are the steps needed to configure 6VPE:

- **Step 1:** As in previous examples, the Backbone must be running MPLS with IGP IPv4 unicast routing protocol and a label distribution protocol, as happened for MPLS VPN for IPv4.
- **Step 2:** PE routers must be running dual stack Ipv4 and IPv6 and forwarding (VRF) instance (with route-target import and export policies).
- **Step 3:** IPv6 VRFs must be associated with the respective interfaces on the PE router.
- **Step 4:** Address families must be configured for the BGP.
- **Step 5:** IPv6 must be enabled and configured between PE and CE routers.

Configuration of the PE

```
PE1-AS2#(config)
ipv6unicast-routing
ipv6 cef
vrf definition 6VPE-S1
rd 1:100
address-family ipv6
route-target export 1:100
route-target import 1:100
exit-address-family
mpls ldp router-id Loopback0 force
mpls label protocol ldp
interface Loopback0
ip address 33.33.33.33 255.255.255.255

interface Fa1/1
vrf forwarding 6VPE-S1
ipv6 address 2001:DB8:2::2/64
ipv6 enable

router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 33.33.33.33
 log-adjacency-changes
 network 0.0.0.0 0.255.255.255 area 0
router bgp 100

router bgp 100
!
 no synchronization
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 22.22.22.22 remote-as 100
 neighbor 22.22.22.22 description to_PE2-AS2
  neighbor 22.22.22.22 update-source Loopback0
 neighbor 33.33.33.33 remote-as 100
  neighbor 33.33.33.33 description to_PE1-AS1
  neighbor 33.33.33.33 update-source Loopback0
  neighbor 44.44.44.44 remote-as 100
 neighbor 44.44.44.44 description to_PE2-AS1
 neighbor 44.44.44.44 update-source Loopback0
 !
 address-family ipv6
 neighbor 22.22.22.22  activate
 neighbor 22.22.22.22 send-label
 neighbor 33.33.33.33 activate
 neighbor 33.33.33.33 send-label
  neighbor 44.44.44.44 activate
  neighbor 44.44.44.44 send-label
 network 2001:21::/64
 exit-address-family

address-family ipv4
no neighbor 22.22.22.22 activate
exit-address-family

address-family vpnv6
neighbor 22.22.22.22 activate
neighbor 22.22.22.22 send-community both
exit-address-family

address-family ipv6 vrf 6VPE-S1
neighbor 2001:DB8:2::1 remote-as 10000
neighbor 2001:DB8:2::1 activate
neighbor 2001:DB8:2::1 as-override
redistribute connected
no synchronization
exit-address-family
```

Configuration of the CE

```
ipv6 unicast-routing
ipv6 cef
interface Loopback0
ip address 192.168.2.1 255.255.255.255
```

```
ipv6 address 2001:DB8:1:1::1/128
interface Fa1/1
ip address 30.1.1.1 255.255.255.0
ipv6 address 2001:DB8:2::1/64
router bgp 10000
bgp log-neighbor-changes
neighbor 2001:DB8:2::2 remote-as 10000
address-family ipv4
no neighbor 2001:DB8:2::2 activate
exit-address-family
address-family ipv6
neighbor 2001:DB8:2::2 activate
network 2001:DB8:1:1::1/128
exit-address-family
```

VI.4.1.2 Verifying 6VPE Operation

Verifying IPv6 VRFs show how to check which IPv6 VRFs are running on the PE router.

```
show vrf ipv6 6VPE-S1
show vrf ipv6 detail
show vrf ipv6 interfaces
```

Verifying MP-BGP for VPNv6 shows how to check whether BGP is running the capability of VPNv6 table.

```
show ip bgp neighbors
```

Verifying VPNv6 prefixes shows how to check if, on the remote PE, the IPv6 VRF prefix made it into the BGP table.

```
show bgp vpnv6 unicast vrf 6VPE-S1 2001:DB8:1:2::1/128
```

Verifying VPNv6 Prefixes shows how to verify the routing entries on the CE and PE routers.

```
show bgp vpnv6 unicast vrf 6VPE-S1 2001:DB8:1:2::1/128
```

Verifying Routing Entries on the CE and PE Routers shows how to check if, on the remote PE, the IPv6 VRF prefix made it into the BGP table.

```
show ipv6 route 2001:DB8:1:2::1/128

show ipv6 route vrf 6VPE-S1 2001:DB8:1:2::1/128
show ipv6 route vrf 6VPE-S1 2001:DB8:1:2::1/128
```

```
Traceroute and Ping
traceroute ipv6 2001:DB8:1:2::1
```

```
ping vrf 6VPE-S1 ipv6 2001:DB8:1:2::1
traceroute vrf 6VPE-S1 ipv6 2001:DB8:1:2::1
```

Verifying Labels shows how to verify the (two) labels for the IPv6 VRF prefix on the ingress PE router.

```
show bgp vpnv6 unicast vrf 6VPE-S1 labels
show ipv6 cef vrf 6VPE-S1 2001:DB8:1:2::1/128
```

Verifying the LFIB on P Routers shows how to verify the label for the next hop (22.22.22.22) VRF prefix on the P routers.

```
show mpls forwarding-table 22.22.22.22 detail
```

Verifying the LFIB on the Egress PE Router shows how to verify the incoming VPN label of the IPv6 VRF prefix in the LFIB on the egress PE router.

```
show mpls forwarding-table vrf 6VPE-S1 2001:DB8:1:2::1/128 detail
```

VI.5 Inter-AS Networks for 6PE and v6VPNs

Inter-AS capability is important because provider networks are seldom single autonomous system (AS) networks. The inter-autonomous system for MPLS VPNs feature allows an MPLS VPN to span service providers and autonomous systems. IPv6 information must be carried across ASes between PEs. The peering model at the Autonomous Systems Boundary Routers (ASBR) is no different from IPv4 peering, but both ASBRs must be configured with IPv4 and IPv6 address families.

- Back-to-back peering - VPNv6 routes can be exchanged within the VRFs or the IPv6 global routing table with logical interfaces between ASBRs. No special arrangement is needed to span the IPv6 services across the ASs.
- VPNv6 peering - ASBRs peer with VPNv6 route exchange in addition to IPv4 route exchange. Labels are also exchanged, along with VPNv6 routes, between ASBRs. The packet path in this case is the same as in the case of IPv4, where the ASBR swaps the VPN label to a new label in each AS. Note that both ASBRs must be IPv6-capable.
- IPv4 peering - In this model, only IPv4 peering is done between ASBRs. The IPv4 reachability is available between ASs either through leaking PE loopbacks or via redistribution. The IPv6 route exchange happens only between route reflectors (RR) in each AS. In this model, the ASBRs do not need to be dual stack. This model is great to start an IPv6 service in an existing IPv4 network, irrespective to geography and ASes, by simply

enabling two PEs with IPv6 address families and configuring the RRs to exchange IPv6 information only.

VI.5.1.1 Inter-Provider VPN Using Back-to-Back VRF

The VRF-to-VRF approach is the simplest method for allowing MPLS-VPN providers to exchange VPN routing information for CE sites in different MPLS domains.

VRFs are configured on the ASBRs to collect VPN client routes. Each sub-interface or interface connected between the ASBRs is dedicated to a single client VRF. The single client VRF can run eBGP, RIPv2, EIGRP, OSPF, or static routing to distribute the VPN routes to its adjacent peer. The use of eBGP is, however, the most common in back-to-back VRF method because eBGP scales best to this type of application, retaining the type of the route and offering better policy, scalability, and security mechanisms. In this method, the LSP paths in adjacent MPLS-VPN AS are interconnected using the IP forwarding mechanism between the AS border routers.

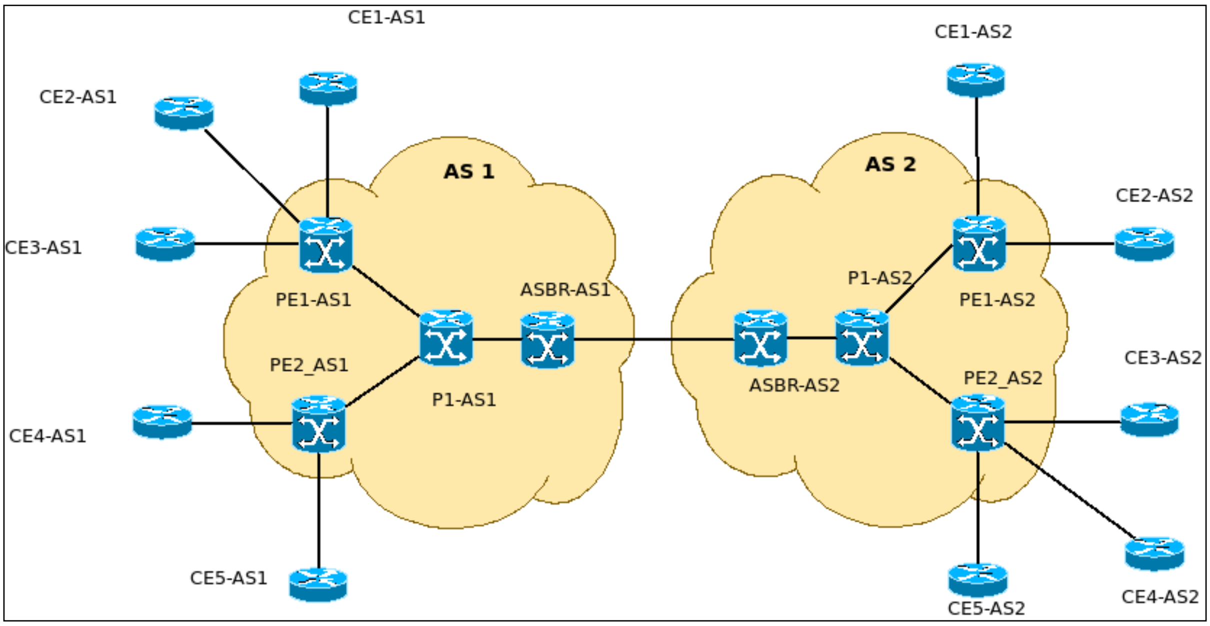


Image VI.3: Inter Autonomous Systems

Configuration of VRF on the PE ASBR routers

VRF and its parameters on the PE ASB Routers ASBR1-AS1 and ASBR2-AS2 have to be configured:

```
ASBR1-AS1(config)#
ip vrf 6VPE-S1
rd 1:100
```

```
route-target export 1:100
route-target import 1:100
interface gi2/0
description connected to 6VPE ASBR1-AS2
ip vrf forwarding 6VPE-S1
ip address 192.168.2.1 255.255.255.0

ip vrf 6VPE-S2
rd 2:100
route-target export 2:100
route-target import 2:100
interface  gi2/0
description connected to 6VPE ASBR1-AS2
ip vrf forwarding 6VPE-S2
ip address 192.168.4.1 255.255.255.0
```

Enabling per VRF PE-CE routing protocol

In this step, per VRF routing protocol on ASBR routers has to be enabled. In this case, eBGP PE-CE routing on the PE and ASBR routers will be used.

```
ASBR1-AS1(config)#router bgp 100
ASBR1-AS1(config-router)# address-family ipv4 vrf 6VPE-S1
ASBR1-AS1(config-router-af)# neighbor 192.168.2.2 remote-as 200
ASBR1-AS1(config-router-af)# neighbor 192.168.2.2 activate
ASBR1-AS1(config-router-af)# no auto-summary
ASBR1-AS1(config-router-af)# no synchronization
ASBR1-AS1(config-router-af)# exit-address-family
ASBR1-AS1(config-router)#address-family ipv4 vrf 6VPE-S2
ASBR1-AS1(config-router-af)# neighbor 192.168.4.2 remote-as 200
ASBR1-AS1(config-router-af)# neighbor 192.168.4.2 activate
ASBR1-AS1(config-router-af)# no auto-summary
ASBR1-AS1(config-router-af)# no synchronization
ASBR1-AS1(config-router-af)# exit-address-family
ASBR2-AS2(config)# router bgp 200
ASBR2-AS2(config-router)# address-family ipv4 vrf 6VPE-S1
ASBR2-AS2(config-router-af)# neighbor 192.168.2.1 remote-as 100
ASBR2-AS2(config-router-af)# neighbor 192.168.2.1 activate
ABR2-AS2(config-router-af)# no auto-summary
ASBR2-AS2(config-router-af)# no synchronization
ASBR2-AS2(config-router-af)# exit-address-family
ASBR2-AS2(config-router)#address-family ipv4 vrf 6VPE-S2
ASBR2-AS2(config-router-af)# neighbor 192.168.4.1 remote-as 100
```

```
ASBR2-AS2(config-router-af)# neighbor 192.168.4.1 activate
ASBR2-AS2(config-router-af)# no auto-summary
ASBR2-AS2(config-router-af)# no synchronization
ASBR2-AS2(config-router-af)# exit-address-family
```

VI.5.1.2 Inter-Provider VPNs Using ASBR-to-ASBR

ASBRs use MP-eBGP to peer with each other to transport VPNv4 routes between autonomous systems (AS). This is called the ASBR-to-ASBR approach, also known as MP-eBGP for VPNv4 exchange, allowing VPNv4 prefixes to be transported across multiple providers. To allow the transportation of VPNv4 prefixes, the link between the ASs must support the exchange of MPLS packets because the VPNv4 updates are encapsulated in MPLS packets when they traverse an AS and therefore need to be encapsulated when going across (between) the AS.

There are some important characteristics to keep in mind when using the ASBR-to-ASBR approach:

- There is no requirement of TDP/LDP or any IGP to be enabled on the link connecting the two ASBRs. The MP-eBGP session between directly connected interfaces on the ASBRs enables the interfaces to forward labeled packets.
- no BGP default route-target filter needs to be configured on an ASBR that does not have any VRFs configured or is functioning as a RR. The command ensures that the ASBR accepts the BGP VPNv4 prefixes from other PE routers inside the AS. The default behavior is to deny incoming VPNv4 prefixes that are not otherwise imported into any local VRF.

VI.6 Carrier Supporting Carriers with BGP

Carrier supporting carriers (CSC) is implemented in circumstances where one service provider needs to use the transport services provided by another service provider. The service provider providing the transport is called the backbone carrier and the service provider using the services provided by the backbone carrier is called a customer carrier. The customer carrier can be either an ISP provider or an MPLS VPN service provider.

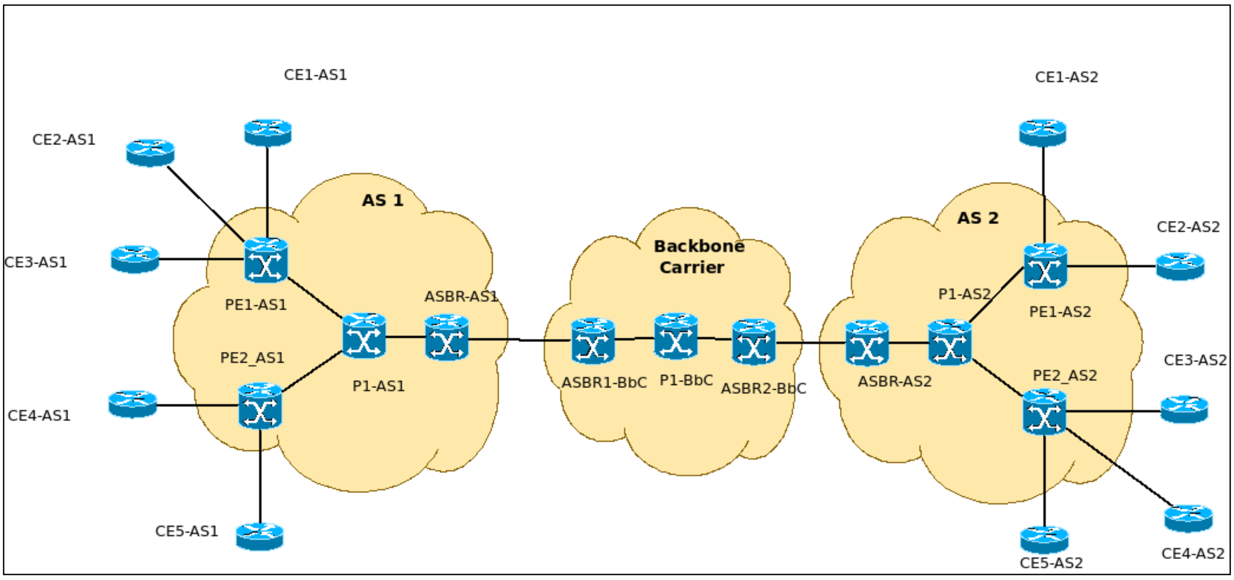


Image VI.4: Carrier Supporting Carrier

VI.6.1.1 Configuring Carrier Supporting Carrier

The above figure shows a carrier supporting carrier network configuration, where the customer carrier is an MPLS VPN provider. The customer carrier has two sites, CE3-AS1 and CE3-AS2. The backbone carrier and the customer carrier use MPLS. The IBGP sessions exchange the external routing information of the ISP. The configurations are the following:

Configuring the CE3-AS1:

```
CE3-AS1(config)#
ip cef
!
interface Loopback0
 ip address 192.168.3.1 255.255.255.255
 no ip directed-broadcast
!
interface Ethernet01/0
 ip address 39.1.1.1 255.255.255.0
 no ip directed-broadcast
!
router ospf 300
 log-adjacency-changes
 redistribute bgp 300 subnets
 passive-interface Ethernet1/1
 network 192.168.3.0 0.0.0.0 area 300
!
router bgp 300
```

```
 no synchronization
 bgp log-neighbor-changes
 timers bgp 10 30
 redistribute connected
 redistribute ospf 300 match internal external 1 external 2
 neighbor 39.1.1.2 remote-as 200
 neighbor 39.1.1.2 advertisement-interval 5
 no auto-summary
ip vrf vpnIPv4
rd 100:1
route target both 100:1
```

Configuring PE1-AS1:

```
PE1-AS1(config)#
ip cef
!
ip vrf vpn2
 rd 200:1
 route-target export 200:1
 route-target import 200:1
mpls label protocol ldp
!
interface Loopback0
 ip address 13.13.13.13 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
router ospf 200
 log-adjacency-changes
 redistribute connected subnets
 passive-interface Ethernet3/0
 network 13.13.13.13 0.0.0.0 area 200
 network 38.0.0.0 0.255.255.255 area 200
!
router bgp 200
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor 15.15.15.15 remote-as 200
 neighbor 15.15.15.15 update-source Loopback0
!
```

```
 address-family ipv4
 neighbor 15.15.15.15 activate
 neighbor 15.15.15.15 send-community extended
 no synchronization
 exit-address-family
!
 address-family vpnv4
 neighbor 15.15.15.15 activate
 neighbor 15.15.15.15 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf vpnIPv4
 neighbor 37.0.0.2 remote-as 300
 neighbor 37.0.0.2 activate
 neighbor 37.0.0.2 as-override
 neighbor 37.0.0.2 advertisement-interval 5
 no auto-summary
 no synchronization
 exit-address-family
```

Configuring ASBR-AS1:

```
mpls label protocol ldp
!
interface Loopback0
 ip address 14.14.14.14 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
router ospf 200
 log-adjacency-changes
 redistribute connected subnets
 network 14.14.14.14 0.0.0.0 area 200
 network 38.0.0.0 0.255.255.255 area 200
 network 46.0.0.0 0.255.255.255 area 200
```

Configuring ASBR-BbC:

```
ip cef distributed
!
ip vrf vpnIPv4
 rd 100:0
 route-target export 100:0
 route-target import 100:0
mpls label protocol ldp
no mpls aggregate-statistics
!
interface Loopback0
 ip address 11.11.11.11 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface Loopback100
 ip vrf forwarding vpn1
 ip address 19.19.19.19 255.255.255.255
 no ip directed-broadcast
!
router ospf 100
 log-adjacency-changes
 passive-interface ATM3/0/0.1
 passive-interface Loopback100
 network 11.11.11.11 0.0.0.0 area 100
 network 33.0.0.0 0.255.255.255 area 100
!
router ospf 200 vrf vpnIPv4
 log-adjacency-changes
 redistribute bgp 100 metric-type 1 subnets
 network 19.19.19.19 0.0.0.0 area 200
 network 46.0.0.0 0.255.255.255 area 200
!
router bgp 100
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor 12.12.12.12 remote-as 100
 neighbor 12.12.12.12 update-source Loopback0
!
 address-family ipv4
 neighbor 12.12.12.12 activate
 neighbor 12.12.12.12 send-community extended
 no synchronization
 exit-address-family
!
 address-family vpnv4
 neighbor 12.12.12.12 activate
 neighbor 12.12.12.12 send-community extended
 exit-address-family
!
 address-family ipv4 vrf vpnIPv4
 redistribute ospf 200 match internal external 1 external 2
 no auto-summary
 no synchronization
 exit-address-family
```

VII. Scenario 4: MPLS – Quality of Service

VII.1 Introduction

Quality of service (QoS) should be an inherent characteristic of large networks in order to enable service differentiation as well as to determine the priorities that should be assigned to an increasing variety of traffic classes. Few networks have unlimited bandwidth, so congestion is always a possibility in the network. QoS is a means to prioritize important traffic over less important traffic and make sure it is delivered.

QoS is used to differentiate diverse classes of traffic based on predefined criteria and assign priorities based on traffic variables that affect the treatment on each router of the network. It might not be implemented in networks with big pipes, however, it is a recommended practice to implement QoS where different classes of traffic are to be transported across the SP network.

The IETF has designated two architectures to implement QoS in an IP network: IntServ and DiffServ. MPLS support for DiffServ uses the Experimental bits in the MPLS header, also referred to as class of service (CoS) bits, for the few classes that DiffServ uses to map the corresponding LSPs. The routers look at these bits to mark, queue, shape, and set the drop precedence of the packet. The big advantage of DiffServ over IntServ is that the DiffServ model needs no signaling protocol. The IntServ model uses a signaling protocol that must run on the hosts and routers. If the network has many thousands of flows, the routers must keep state information for each flow passing through it. This is a serious scalability issue that justifies why IntServ has not proven to be popular. Here, we will only address DiffServ.

The first and foremost step in the deployment of QoS is to identify the various traffic classes that need to be supported by the network. Traffic can be classified based on the type of traffic (voice, applications, data, etc.) and on the properties of the traffic pattern. After traffic has been classified into different classes, the next step is to identify what QoS operations will be performed on each of these classes on the local router. But, although QoS is an end-to-end implementation, it is traditionally configured on all routers in the path from one end to the other. Various portions of the network can be configured with different QoS schemes to handle different types of traffic. The process of defining the QoS operations for a certain traffic class is also called Service Policy definition. Finally, after the policies are defined, they are applied on the device interface.

VII.2 Objectives

This chapter addresses the following topics:

- DiffServ Tunneling
    - Uniform mode
    - Pipe mode
- MPLS QoS on the CE Router
- MPLS QoS for Layer 2 VPN Implementations

VII.3 DiffServ Tunneling

Diffserv Tunneling Modes introduce a new Per-Hop-Behavior (PHB), which allows differentiated QoS in a providers network. The tunneling mode is defined at the edge of the network, normally in the PE label switch routers (LSRs) (both ingress and egress).

VII.3.1 Uniform mode

DiffServ Tunneling Uniform Mode has only one layer of QoS, reaching an end-to-end functionality. The ingress PE router (PE1) copies the DSCP from the incoming IP packet into the MPLS EXP bits of the imposed labels. As the EXP bits travel through the core, they may or may not be modified by intermediate P routers. The usefulness of the Uniform model is that there is only one DiffServ information for a packet. This is the DiffServ information encoded in the top label. Whether this is different from the underlying DiffServ information is not important, because the top DiffServ information is propagated down at the egress LSR of the LSP.

The following rules apply:

- The LSP DiffServ information must be derived from the Tunneled DiffServ information on the ingress LSR.
- On an intermediate LSR (a P router), the LSP DiffServ information of the outgoing label is derived from the LSP DiffServ information of the incoming label.
- On the egress LSR, the LSP DiffServ information must be propagated to the Tunneled DiffServ information.

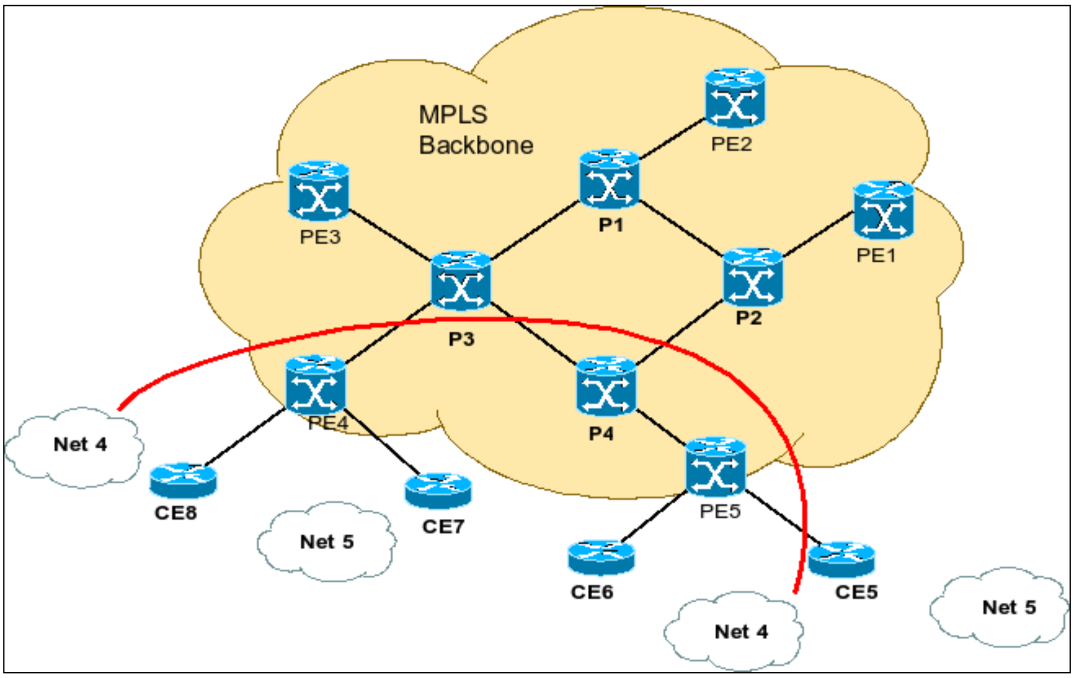


Image VII.1: QoS – Uniform Mode

Uniform mode configurations:

Ingress PE4:

This configuration maps the IP Precedence of the incoming IP packets to the MPLS EXP bits.

```
PE5(config)#
class-map match-all IP-AF11
  match ip precedence 0
!
class-map match-all IP-AF12
  match ip precedence 1
!
class-map match-all IP-AF21
  match ip precedence 2
!
class-map match-all MPLS-AF11
  match mpls experimental topmost 0
!
class-map match-all MPLS-AF12
  match mpls experimental topmost 1
!
class-map match-all MPLS-AF21
  match mpls experimental topmost 2
!
policy-map set-MPLS-PHB
class IP-AF11
```

```
  police 8000 conform-action set-mpls-exp-imposition-transmit 0 exceed-action drop
 class IP-AF12
!
  police 10000 conform-action set-mpls-exp-imposition-transmit 1 exceed-action set-mpls-exp-imposition-transmit 0
 class IP-AF21
!
  police 12000 conform-action set-mpls-exp-imposition-transmit 2 exceed-action set-mpls-exp-imposition-transmit 1
!
policy-map output-qos
 class MPLS-AF11
  bandwidth percent 5
  random-detect
 class MPLS-AF12
  bandwidth percent 10
  random-detect
 class MPLS-AF21
!
interface serial2/0
 ip address 23.2.2.2 255.255.255.0
 max-reserved-bandwidth 90
 service-policy output output-qos
 tag-switching ip
!
interface Fa1/0
 ip vrf forwarding Net4
 ip address 38.1.1.2 255.255.255.0
 service-policy input set-MPLS-PHB
!
```

VII.3.2 Pipe mode

The Pipe mode works in a similar way to the Uniform mode, except that in the MPLS2IP condition the EXP value of the top-most label is not copied as the IP Precedence value of the IP packet. This mode is used if the QoS implemented by the SP is required to be independent of the customer's QoS policy. In Pipe mode, the IP Precedence of the underlying IP packet is unchanged.

The advantage for Pipe model is that the original Tunneled DiffServ information is preserved when the packet leaves the MPLS network. This means that the IP DiffServ information or the tunneled MPLS DiffServ information remains unchanged. When customers connect to the MPLS network, their QoS information is tunneled transparently through the MPLS network. Furthermore, if customers have their own rules for QoS, the MPLS service provider, who imposes his own rules on the packets at the ingress LSR without changing the original QoS of the packet, can ignore the customers' rules. This is a lot more scalable than catering for the QoS of each customer.

Diffserv Tunneling Pipe Mode uses two layers of QoS:

- An underlying QoS for the data, which remains unchanged when traversing the core.
- A per-core QoS, which is separate from that of the underlying IP packets. This per-core QoS PHB remains transparent to end users.

When a packet reaches the edge of the MPLS core, the egress PE router (PE2) classifies the newly exposed IP packets for outbound queuing based on the MPLS PHB from the EXP bits of the recently removed label.

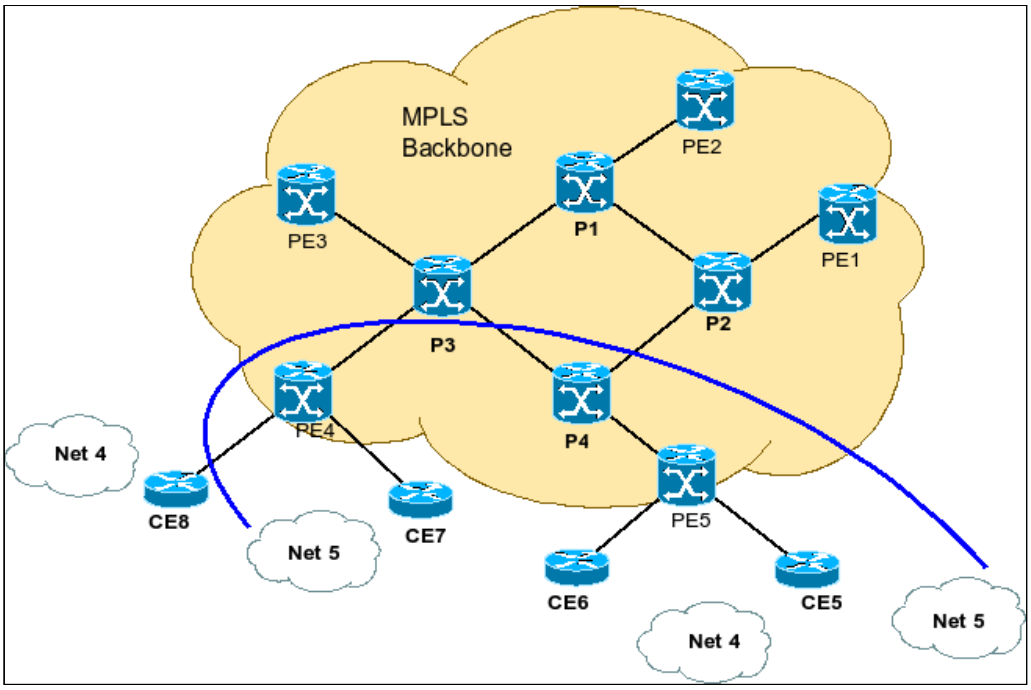


Image VII.1: QoS – Pipe Mode

On the PE routers, packets are received with different IP Precedence values from the attached CE routers. Therefore, the PE routers can map or assign class based on ingress packet IP Precedence. On label imposition and when leaving the router, the MPLS EXP bits are used to classify the traffic into three traffic classes: gold, silver, and bronze (the topmost).

PE4 MPLS Condition Configuration:

```
PE4(config)#
class-map match-all IP-AF22
 match ip precedence 0 1
```

```
class-map match-all IP-AF31
  match ip precedence 2 3
class-map match-all IP-AF32
  match ip precedence 4 5
!
class-map match-all bronze
  match mpls experimental topmost 2 3
class-map match-all silver
  match mpls experimental topmost 4
class-map match-all gold
  match mpls experimental topmost 5
!
policy-map set-MPLS-PHB
  class IP-AF22
   police 12000 conform-action set-mpls-exp-imposition-transmit
             3 exceed-action set-mpls-exp-imposition-transmit 2
  class IP-AF31
   police 12000 conform-action set-mpls-exp-imposition-transmit
             4 exceed-action set-mpls-exp-imposition-transmit 3
  class IP-AF32
   police 12000 conform-action set-mpls-exp-imposition-transmit
             5 exceed-action set-mpls-exp-imposition-transmit 4
!
policy-map output-qos
  class gold
   bandwidth 40
   random-detect
  class silver
   bandwidth 30
   random-detect
  class bronze
   bandwidth 20
   random-detect
!
interface Serial2/0
 ip address 23.1.1.2 255.255.255.0
 max-reserved-bandwidth 90
 service-policy output output-qos
 tag-switching ip
!
interface Fa1/1
 ip vrf forwarding Net5
 ip address 37.1.1.2 255.255.255.0
 service-policy input set-MPLS-PHB
!
```

PE5 router configuration:

```
PE4(config)#
class-map match-all MPLS-AF11
  match mpls experimental topmost 0
class-map match-all MPLS-AF12
  match mpls experimental topmost 1
class-map match-all MPLS-AF21
  match mpls experimental topmost 2
class-map match-all MPLS-AF22
  match mpls experimental topmost 3
class-map match-all MPLS-AF31
  match mpls experimental topmost 4
class-map match-all MPLS-AF32
  match mpls experimental topmost 5
!
class-map match-all gold
  match qos-group 5
class-map match-all silver
  match qos-group 4
class-map match-all bronze
  match qos-group 3
  match qos-group 2
!
policy-map qos-group-in
  class MPLS-AF11
   set qos-group mpls experimental topmost
   set discard-class 0
  class MPLS-AF12
   set qos-group mpls experimental topmost
   set discard-class 1
  class MPLS-AF21
   set qos-group mpls experimental topmost
   set discard-class 2
  class MPLS-AF22
   set qos-group mpls experimental topmost
   set discard-class 3
  class MPLS-AF31
```

```
   set qos-group mpls experimental topmost
   set discard-class 4
  class MPLS-AF32
   set qos-group mpls experimental topmost
   set discard-class 5
!
policy-map qos-group-out
  class gold
   bandwidth 40
   random-detect discard-class-based
  class silver
   bandwidth 30
   random-detect discard-class-based
  class bronze
   bandwidth 20
   random-detect discard-class-based
!
interface Fa1/0
 ip vrf forwarding Net5
 ip address 35.1.1.2 255.255.255.0
 max-reserved-bandwidth 90
 service-policy output qos-group-out
!
interface Serial2/0
 ip address 24.1.1.2 255.255.255.0
 service-policy input qos-group-in
 tag-switching ip
!
```

VIII. Conclusions

MPLS is the “De facto” technology operated by the Service Providers in their Backbone infrastructures. Its study is important not just by the extension of its deployment but also due to its wide potential to solve diverse networking challenges.

MPLS is a switching method used to get packets from one place to another by going through a series of hops. Which hops a packet goes through can be determined by IGP routing or by MPLS TE.

In the course of this work, a progressive approach to the MPLS functioning was made, starting with an overview of its diverse components, and moving to scenarios where constructs were unveiled incrementally. Like the protocols themselves, the complexity layers were consecutively stacked in order to achieve more complex/elaborated functional stages.

The MPLS main features were divided and covered in four functional chapters:

1) Traffic engineering

Shortest-path routing protocols send traffic down the shortest path, without considering other network parameters, such as utilization and traffic demands, rendering some links at the Backbone congested while others remain under-utilized. Furthermore, in the event of failure (router link or path), conversion times are at this level unacceptable. TE aims to address this and further aspects, as it was demonstrated in this chapter. TE also plays an important role in most deployments as a supporting mechanism, as was demonstrated in subsequent chapters.

2) MPLS-VPN

MPLS VPN is the most widely used mechanism in MLPS. It does not only addresses the most profitable market needs, the implementation of virtual circuits that replace the aging leased lines, but serves as a technological layer that allows the implementation of more complex and specific features. This chapter intended to clarify the core functionalities of MPLS VPNs, while building a framework to be used in subsequent chapters.

3) MPLS-IPv6

Following the current pace, the IPv4 address space is scheduled to be exhausted during the next year, and IPv6 is a becoming reality, more than a curiosity. MPLS provides the perfect environment to its increasing adoption, allowing its coexistence with current IPv4 during the transition period. Different approaches that allow the integration of this emerging technology and the necessary protocols that have to be deployed in SP networks where demonstrated.

4) MPLS-QoS

QoS is an umbrella term that covers network performance characteristics, most often used to describe a set of techniques used to manage packet loss, latency, and jitter. DiffServ has proven to be quite scalable, using edge classification, per-hop queuing and discard behaviors in the core, which means that most of the work is done at the edge. Some prominent aspects were lightly focused, as these functionalities are very hard to simulate in a lab environment.

The innovative relevant aspects of this work are related not specifically to the MPLS technologies that were simulated, since there are many works in this field, but in the way simulation was carried out. The cloud computing deployment at the Amazon servers demonstrated that more elaborated scenarios can be studied, scaling to whatever is required, because when dealing with backbone network technologies size matters.

Using cloud computing is a very significant step in emulation/simulation, because this emerging technology can serve as a platform to all sort of simulation environments and can constitute a basis for most of the current science studies in this area. Its scalability characteristics allow very appealing cost effective deployments, since outsourcing is a Universal trend that extends from the industry to the research world.

This work could be enriched in both directions, by either extending the cloud computing concept to other areas of simulation, or in the MPLS domain where a lot of relevant work still remains to be done. Learning by doing is the most effective and fast way to acquire knowledge; so, the more scenarios are developed and made available, the better are the chances for all networking enthusiasts to further develop their skills and know how.

References

[1] Cox RV, Haskell BG, Lecun Y, Shahraray B, Rabiner L. Scanning the technology - On the Applications of Multimedia Processing to Communications . Proceedings of the IEEE, Vol. 86, No. 5, 1998.

[2] http://www.mplstutorial.com/, accessed at 3/09/2010

[3] Alwayn V. Advanced MPLS design and implementation. Cisco Press, 2001.

[4] Multiprotocol Label Switching (MPLS) on Cisco Routers. Cisco Systems. http://www.cisco.com/en/US/docs/ios/12_0s/feature/guide/fs_rtr22.html

[5] Ruela J, Ricardo M. MPLS-Multiprotocol Label Switching. http://paginas.fe.up.pt/~mricardo/doc/ieeehandbook/ieeehandbook.pdf

[6] Pepelnjak I, Guichard J. MPLS and VPN architectures, Band 1. Cisco Press, 2002.

[7] Hardy D, Malléus G, Méreur JN. Networks: internet, telephony, multimedia: convergences and complementaries. Springer, 2002.

[8] Morrow MJ, Sayeed A. MPLS and Next-Generation Networks: Foundation for NGN and Enterprise Virtualization. Pearson Education, 2008.

[9] Tan NK. Building VPNs: with Ipsec and MPLS. McGraw-Hill Professional, 2003.

[10] deGhein L. MPLS Fundamentals. Cisco Press 2006.

[11] Lobo L, Lakshman U. MPLS Configuration on Cisco IOS Software. Cisco Press, 2005.

[12] Guichard J, LeFaucheur F, Vasseur JP. Definitive MPLS Network Designs. Cisco Press, 2005.

[13] Pise RN, Kulkarni SA, Pawar RV. Packet Forwarding with Multiprotocol Label Switching . World Academy of Science, Engineering and Technology 12, 2005 .

[14] Halimi A, Statovci-Halimi B. Overview on MPLS Virtual Private Networks. Photonic Network Communications, Vol. 4, No. 2, 115-131.

[15] Redford R. Enabling Business IP Services with Multiprotocol Label Switching. White Paper. Multiservice Switching Business Unit, Cisco Systems.

[16] MPLS Architecture. http://www.mplsinfo.org/architecture.html, accessed at 4/09/2010

[17] Black U. MPLS and label switching networks (2nd Ed.). Prentice Hall – Gale, 2002.

[18] MPLS Core Protocols; Chapter 5.

http://www8.cs.umu.se/kurser/TDBD03/VT01/slides/MPLS.pdf

[19] to Gonzales F, Chang CH, Chen LW, Lin CK. Using Multiprotocol Label Switching (MPLS) improve IP Network Traffic Engeneering. http://ece.ut.ac.ir/classpages/F83/Advanced%20Computer%20Networks/PAPERS/MPLS/using%20Networks/PAPERS/MPLS/using-multiprotocol-label-switching.pdf.

[20] JUNOSe™ Software for E-series™ Routing Platforms; BGP and MPLS Configuration Guide. Release 9.3.x http://www.juniper.net/techpubs/software/erx/junose93/swconfig-bgp-mpls/book-swconfig-bgp-mpls-TOC.html

[21] http://www.ietf.org/rfc/rfc3209.txt, accessed at 4/09/2010

[22] Ohba Y. Issues on loop prevention in MPLS networks. Communications Magazine, IEEE. Vol. 13, Issue 12, 1999.

[23] Wang N, Kon Ho H, Pawlou G, Howarth M. An overview of routing optimization for Internet Traffic Engeneering. IEEE Communications Surveys, Vol. 12, No. 1, 1st quarter 2008.

[24] Gasson A. MPLS-TE: Fundamentals and Advanced Fast Reroute. 2001, Cisco Systems.

[25] Osborne E, Simha A. Traffic Engeneering with MPLS. Cisco Press, 2002.

[26] Smith E. Comparison of Traffic Engineering Techniques in Connectionless and Connection-Oriented Networks. The Journal of The Communications Network; Vol. 6 Part 2, 2007 .

[27] http://tools.ietf.org/html/rfc4804, accessed at 8/09/2010

[28] Teare D. Implementing Cisco IP Routing (ROUTE) Foundation Learning Guide: Foundation learning for the ROUTE 642-902 Exam. Cisco Press 2010.

[29] Advanced topics in MPLS-TE Deployment. White Paper. Cisco Systems.

[30] Francois P, Bonaventure O. An evaluation of IP-based Fast Reroute Techniques. CoNEXT'05, October 24–27, 2005, Toulouse, France.

[31] MPLS Traffic Engineering–Autotunnel Primary and Backup. Cisco Systems. First Published: January 26, 2004 , Last Updated: November 10, 2010.

[32] Guichard J, Pepelnjak I, Apcar J. MPLS and VPN architecture. Cisco Press, 2003.

[33] Introduction to Cisco MPLS VPN Technology. Cisco VPN Solutions Center: MPLS Solution User Guide. Cisco Systems.

[34] MPLS VPN-Route Target Rewrite. Cisco Systems. First Published: August 26, 2003 , Last Updated: May 31, 2007

[35] IPv6- A Service Provider View in Advancing MPLS Networks. http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_8-2/ipv6.html

[36] http://tools.ietf.org/html/rfc4659, accessed at 14/09/2010

[37] IPv6 over MPLS (Cisco 6PE). Cisco Systems. Application Note http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_data_sheet09186a008052edd3.html

[38] Popoviciu C, Levy-Abegnoli E, Grossetete P. Deploying IPv6 Networks. Cisco Press, 2006.

[39] Cisco IOS Configuration Guide. Release 12.2SX. Cisco Systems. http://www.cisco.com/en/US/docs/ios/netflow/configuration/guide/12_2sx/nf_12_2sx_book.html

[40] MPLS VPN-VRF CLI IPv4 and IPv6 VPNs. Cisco Systems. http://www.cisco.com/en/US/docs/ios/12_2sr/12_2srb/feature/guide/sr_mpvrf.html

[41] Castelli M. Network consultants handbook. Cisco Press, 2002.

[42] Li Q, Jinmei T, Shima K. IPv6 Core Protocols Implementation. The Morgan Kaufmann Series in Networking. 2006.

[43] Fineberg V. QoS Support in MPLS Networks. MPLS/Frame Relay Alliance White Paper. MPLS Forum. May 2003

[44] Deploying Guaranteed-Bandwidth Services with MPLS. White Paper. Cisco Systems. http://www.cisco.com/en/US/tech/tk436/tk428/technologies_white_paper09186a00800a3e69.shtml

[45] Pereira A, Monteiro E . Dynamic Mapping between the Controlled-Load IntServ Service and the Assured Forward DiffServ PHB . High-Speed Networks and Multimedia Communications. Lecture Notes in Computer Science, 2003, Volume 2720/2003, 1-10 .

[46] Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.2. http://www.cisco.com/en/US/docs/ios/12_2/qos/configuration/guide/fqos_c.htm

[47] Alvarez S. QoS for IP/MPLS networks. Cisco Press, 2006.

[48] http://tools.ietf.org/html/rfc2474, accessed at 17/09/2010

[49] http://www.ietf.org/rfc/rfc2475, accessed at 17/09/2010

[50] DiffServ -The Scalable End-to-End QoS Model. White Paper, Cisco Systems. http://www.cisco.com/en/US/technologies/tk543/tk766/technologies_white_paper09186a00800a3e2f.html

[51] Traffic Policing. Cisco IOS Software Releases 12.2 T. Cisco systems. http://www.cisco.com/en/US/docs/ios/12_2t/12_2t2/feature/guide/ftpoli.html

[52] QoS Policing: Comparing Traffic Policing and Traffic Shaping for Bandwidth Limiting. Document ID: 19645. Cisco Systems. http://www.cisco.com/en/US/tech/tk543/tk545/technologies_tech_note09186a00800a3a25.shtml

[53] Szigeti T, Hattingh C. End-to-end QoS network design. Cisco Press. 2005.

[54] Davie BS, Farrel A. MPLS: next steps. Morgan Kaufmann, 2008.

[55] Multiprotocol Label Switching Quality of Service: Cisco IOS MPLS Quality of Service. White Paper. Cisco Systems. http://www.cisco.com/en/US/products/ps6607/products_white_paper09186a00800a4455.shtml

[56] Luo W, Pignatoro C, Bokotey D, Chan A. Layer 2 VPN Architecture. Cisco Press, 2005.

[57] Lewis M. Comparing, Designing and Deploying VPNs. Cisco Press, 2006.

[58] Cisco Any Transport over MPLS. White Paper. Cisco Systems. http://www.cisco.com/warp/public/cc/so/neso/vpn/unvpnst/atomf_ov.htm#wp34608

[59] Harrison J. VPN Technologies – a comparison. 2003. http://ftp.utcluj.ro/pub/users/dadarlat/retele_master/mpls-vpn/MPLS_VPN/vpntechwp.pdf

[60] Importance of Virtual Private Lan Services. http://e-articles.info/e/a/title/Importance-of-Virtual-Private-LAN-services/

[61] http://www.semsim.com/

[62] http://www.boson.com/AboutNetSim.html

[63] http://www.certexams.com/

[64] http://www.routersim.com/

[65] http://mimic-virtual-lab-ccna.software.informer.com

[66] http://www.techexams.net/

[67] http://www.ccna4u.org/2010/05/packet-tracer-version-53-software.html

[68] http://networksims.com/

[69] http://www.dynagen.org/

[70] Burdette B. Cisco Router Simulator Comparison. April 2010.

[71] http://www.dynagen.org/tutorial.htm, accessed at 10/2010 and 11/2010

[72] GNS3 Dynamips for Cisco Emulation Step by Step Tutorial
http://www.docstoc.com/docs/32277462/GNS3-Dynamips-for-Cisco-Emulation-Step-by-Step-Tutorial

[73] Dynamips/Dynagen Setting Manual : Connection with a real network.
http://netguide.sakura.ne.jp/dynamips/english/real-network.html

[74] http://7200emu.hacki.at/

[75] http://aws.amazon.com/ec2/

Appendix

I. Scenario 1: MPLS – Traffic Engineering

I.1 Router interfaces IP list

Router	Interface	IP	connected to	Interface	IP
P1	s1/0	1.3.1.1	P2	s1/0	1.3.1.2
	s1/1	1.1.2.1	PE1	s1/1	1.1.2.2
	s1/2	1.2.1.2	P4	s1/1	1.2.1.1
	s1/3	1.2.2.1	P5	s1/0	1.2.2.2
	Loopback0	1.1.1.1			
P2	s1/0	1.3.1.2	P1	s1/0	1.3.1.1
	s1/1	1.3.2.1	P3	s1/0	1.3.2.2
	Loopback0	2.2.2.2			
P3	s1/0	1.3.2.2	P2	s1/1	1.3.2.1
	s1/1	1.1.7.1	PE3	s1/0	1.1.7.2
	s1/2	1.3.3.2	P6	s1/1	1.3.3.1
	Loopback0	3.3.3.3			
P4	s1/0	1.1.4.1	PE1	s1/2	1.1.4.2
	s1/1	1.2.1.1	P1	s1/2	1.2.1.2
	Loopback0	4.4.4.4			
P5	s1/0	1.2.2.2	P1	S1/3	1.2.2.1
	s1/1	1.2.3.1	P6	s1/0	1.2.3.2
	Loopback0	5.5.5.5			
P6	s1/0	1.2.3.2	P5	s1/1	1.2.3.1
	s1/1	1.3.3.1	P3	s1/2	1.3.3.2
	Loopback0	6.6.6.6			
P7	s1/0	2.1.1.1	P8	s1/1	2.1.1.2
	s1/1	1.1.3.1	PE1	s1/0	1.1.3.2
	Loopback0	7.7.7.7			
P8	s1/0	2.1.2.1	P9	s1/1	2.1.2.2
	s1/1	2.1.1.2	P7	s1/0	2.1.1.1
	Loopback0	8.8.8.8			
P9	s1/0	1.1.6.1	PE2	s1/1	1.1.6.2
	s1/1	2.1.2.2	P8	s1/0	2.1.1.2
	Loopback0	9.9.9.9			
P10	s1/0	1.1.3.10	PE1	s1/1	1.1.3.9
	s1/1	1.1.2.2	P1	s1/1	1.1.2.1
	s1/2	1.1.4.2	P4	s1/0	1.1.4.1
	Loopback0	10.10.10.10			

Router	Interface	IP	connected to	Interface	IP
PE1	f2/0	10.1.1.2	CE1	f1/0	10.1.1.1
	s1/0	1.1.3.2	P7	s1/1	1.1.3.1
	s1/1	1.1.3.9	P10	s1/1	1.1.3.10
	Loopback0	11.11.11.11			
PE2	f2/0	13.1.1.2	CE2	f1/1	13.1.1.2
	s1/0	1.1.5.2	PE3	s1/1	1.1.5.1
	s1/1	1.1.6.2	P9	s1/0	1.1.6.1
	Loopback0	22.22.22.22			
PE3	f2/0	14.1.1.2	CE3	f1/0	14.1.1.1
	s1/0	1.1.7.2	P3	s1/1	1.1.7.1
	s1/1	1.1.5.1	PE2	s1/0	1.1.5.2
	Loopback0	33.33.33.33			
CE1	f1/0	10.1.1.1	PE1	f2/0	10.1.1.2
	f1/1	11.1.1.1	Network A	f0/0	11.1.1.2
	Loopback0	70.70.70.70			
CE2	f1/0	12.1.1.1	Network C	f0/0	12.1.1.2
	f1/1	13.1.1.1	PE2	f2/0	13.1.1.2
	Loopback0	20.20.20.20			
CE3	f1/0	14.1.1.1	PE3	F2/0	14.1.1.2
	f1/1	15.1.1.1	Network B	f0/0	15.1.1.2
	Loopback0	30.30.30.30			

I.2 Dynagen's .net file

```
autostart = False

################################################################################################
##

[localhost:7203]
    workingdir = /tmp
    udp = 10030
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER CE1]]
        cnfg = /labs/te/config/CE1.cfg
        console = 2011
        slot1 = PA-2FE-TX
        f1/0 = PE1 f2/0

        x = -638.0
        y = -111.0

[localhost:7204]
    workingdir = /tmp
    udp = 10040
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
[[ROUTER CE2]]
        cnfg = /labs/te/config/CE2.cfg
        console = 2012
        slot1 = PA-2FE-TX
        f1/1 = PE2 f2/0
        x = 437.0
        y = -269.0
[localhost:7205]
    workingdir = /tmp
    udp = 10050
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER CE3]]
        cnfg = /labs/te/config/CE3.cfg
        console = 2013
        slot1 = PA-2FE-TX
        f1/0 = PE3 f2/0

        x = 396.0
        y = 101.0

#################################################################################################
##########
[localhost:7206]
    workingdir = /tmp
    udp = 10060
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
 [[ROUTER PE1]]
        cnfg = /labs/te/config/PE1.cfg
        console = 2021
        slot1 = PA-4T

        slot2 = PA-2FE-TX
        f2/0 = CE1 f1/0
        x = -486.0
        y = -96.0

[localhost:7207]
    workingdir = /tmp
    udp = 10070
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER PE2]]
        cnfg = /labs/te/config/PE2.cfg
        console = 2022
        slot1 = PA-4T
        s1/0 = PE3 s1/1

        slot2 = PA-2FE-TX
        f2/0 = CE2 f1/1
        x = 253.0
        y = -250.0

[localhost:7208]
    workingdir = /tmp
    udp = 10080
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER PE3]]
        cnfg = /labs/te/config/PE3.cfg
        console = 2023
        slot1 = PA-4T

        slot2 = PA-2FE-TX
        f2/0 = CE3 f1/0
        x = 235.0
        y = -62.0


#################################################################################################
##
[localhost:7209]
    workingdir = /tmp
    udp = 10090
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
[[ROUTER P1]]
        cnfg = /labs/te/config/P1.cfg
        console = 2031
        slot1 = PA-4T
        s1/0 = P2 s1/0
        s1/1 = P10 s1/0
        s1/2 = P4 s1/1
        s1/3 = P5 s1/0
        x = -298.0
        y = -81.0

[localhost:7210]
    workingdir = /tmp
    udp = 10100
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER P2]]
        cnfg = /labs/te/config/P2.cfg
        console = 2032
        slot1 = PA-4T
        s1/1 = P3 s1/0
        x = -147.0
        y = -65.0


[localhost:7211]
    workingdir = /tmp
    udp = 10110
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER P3]]
        cnfg = /labs/te/config/P3.cfg
        console = 2033
        slot1 = PA-4T
        s1/1 = PE3 s1/0
        s1/2 = P6 s1/1
            s1/3 = PE2 s1/3
        x = 32.0
        y = -53.0

[localhost:7212]
    workingdir = /tmp
    udp = 10120
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True

    [[ROUTER P4]]
        cnfg = /labs/te/config/P4.cfg
        console = 2034
        slot1 = PA-4T
        s1/0 = P10 s1/2

        x = -318.0
        y = 148.0

[localhost:7213]
    workingdir = /tmp
    udp = 10130
```

```
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER P5]]
        cnfg = /labs/te/config/P5.cfg
        console = 2035
        slot1 = PA-4T

        s1/1 = P6 s1/0
        x = -110.0
        y = 141.0

[localhost:7214]
    workingdir = /tmp
    udp = 10140
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER P6]]
        cnfg = /labs/te/config/P6.cfg
        console = 2036
        slot1 = PA-4T

        s1/1 = P3 s1/2
        x = 55.0
        y = 133.0

[localhost:7215]
    workingdir = /tmp
    udp = 10150
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER P7]]
        cnfg = /labs/te/config/P7.cfg
        console = 2037
        slot1 = PA-4T
        s1/0 = P8 s1/1
        s1/1 = PE1 s1/0
        x = -299.0
        y = -251.0

[localhost:7216]
    workingdir = /tmp
    udp = 10160
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER P8]]
        cnfg = /labs/te/config/P8.cfg
        console = 2038
        slot1 = PA-4T
        s1/0 = P9 s1/1

        x = -121.0
        y = -263.0

[localhost:7217]
    workingdir = /tmp
    udp = 10170
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True

    [[ROUTER P9]]
        cnfg = /labs/te/config/P9.cfg
        console = 2039
        slot1 = PA-4T
        s1/0 = PE2 s1/1

        x = 25.0
        y = -242.0

[localhost:7218]
    workingdir = /tmp
    udp = 10180
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True

    [[ROUTER P10]]
        cnfg = /labs/te/config/P10.cfg
        console = 2040
        slot1 = PA-4T
        s1/1 = PE1 s1/1
```

I.3 Dynamips Start Script

```
#!/bin/bash

sudo dynamips -H 7203 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7204 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7205 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7206 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7207 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7208 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7209 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7210 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7211 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7212 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7213 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7214 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7215 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7216 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7217 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7218 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
```

I.4 Router configurations

CE1

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname CE1
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 70.70.70.70 255.255.255.255
!
interface FastEthernet1/0
 ip address 10.1.1.1 255.255.255.0
 description Connected to PE1 f2/0
 duplex auto
 speed auto
!
interface FastEthernet1/1
 ip address 11.1.1.1 255.255.255.0
 description Connected to Network A f0/0
 duplex auto
 speed auto
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

CE2

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname CE2
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 20.20.20.20 255.255.255.255
!
interface FastEthernet1/0
 ip address 12.1.1.1 255.255.255.0
 description Connected to Network C f0/0
 duplex auto
 speed auto
!
interface FastEthernet1/1
 ip address 13.1.1.1 255.255.255.0
 description Connected to PE2 f2/0
 duplex auto
 speed auto
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

CE3

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname CE3
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 30.30.30.30 255.255.255.255
!
interface FastEthernet1/0
 ip address 14.1.1.1 255.255.255.0
 description Connected to PE3 f2/0
 duplex auto
 speed auto
!
```

```
interface FastEthernet1/1
 ip address 15.1.1.1 255.255.255.0
 description Connected to Network C f0/0
 duplex auto
 speed auto
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

P1

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname P1
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
no logging console
!
no aaa new-model
ip source-route
ip cef
!
no ip domain lookup
no ipv6 cef
!
mpls label protocol ldp
mpls traffic-eng tunnels
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 1.1.1.1 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface Serial1/0
 description Connected to P2 s1/0
 ip address 1.3.1.1 255.255.255.252
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp resource-provider none
 ip rsvp bandwidth 1000
 serial restart-delay 0
!
interface Serial1/1
 description Connected to PE1 s1/1
 ip address 1.1.2.1 255.255.255.252
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp resource-provider none
 ip rsvp bandwidth 1000
 serial restart-delay 0
!
interface Serial1/2
 description Connected to P4 s1/1
 ip address 1.2.1.2 255.255.255.252
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp resource-provider none
 ip rsvp bandwidth 1000
 serial restart-delay 0
!
interface Serial1/3
 description Connected to P5 s1/0
 ip address 1.2.2.1 255.255.255.252
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp resource-provider none
 ip rsvp bandwidth 1000
 serial restart-delay 0
!
router ospf 100
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id Loopback0
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
```

```
!
line con 0
 exec-timeout 0 0
 logging synchronous
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

P2

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname P2
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
no logging console
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
mpls label protocol ldp
mpls traffic-eng tunnels
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 2.2.2.2 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface Serial1/0
 description Connected to P1 s1/0
 ip address 1.3.1.2 255.255.255.252
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp resource-provider none
 ip rsvp bandwidth 1000
 serial restart-delay 0
!
interface Serial1/1
```

```
 description Connected to P3 s1/0
 ip address 1.3.2.1 255.255.255.252
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp resource-provider none
 ip rsvp bandwidth 1000
 serial restart-delay 0
!
interface Serial1/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
!
router ospf 100
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id Loopback0
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

P3

```
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname P3
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
no logging console
```

```
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
mpls label protocol ldp
mpls traffic-eng tunnels
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 3.3.3.3 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface Serial1/0
 ip address 1.3.2.2 255.255.255.252
 description Connected to P2 s1/1
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp resource-provider none
 ip rsvp bandwidth 1000
 serial restart-delay 0
!
interface Serial1/1
 ip address 1.1.7.1 255.255.255.252
 description Connected to PE3 s1/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp resource-provider none
 ip rsvp bandwidth 1000
 serial restart-delay 0
!
interface Serial1/2
 ip address 1.3.3.2 255.255.255.252
 description Connected to P6 s1/1
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp resource-provider none
 ip rsvp bandwidth 1000
 serial restart-delay 0
!
interface Serial1/3
 ip address 3.3.1.3 255.255.255.252
 description Connected to PE2 s1/1
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp resource-provider none
 ip rsvp bandwidth 1000
 serial restart-delay 0
!
router ospf 100
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id Loopback0
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 exec-timeout 0 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

P4

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname P4
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
no logging console
!
no aaa new-model
ip source-route
ip cef
!
no ip domain-lookup
!
no ipv6 cef
!
mpls label protocol ldp
mpls traffic-eng tunnels
!
archive
 log config
  hidekeys
```

```
!
interface Loopback0
 ip address 4.4.4.4 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface Serial1/0
 description Connected to PE1 s1/2
 ip address 1.1.4.1 255.255.255.252
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp resource-provider none
 ip rsvp bandwidth 1000
 serial restart-delay 0
!
interface Serial1/1
 description Connected to P1 s1/2
 ip address 1.2.1.1 255.255.255.252
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp resource-provider none
 ip rsvp bandwidth 1000
 serial restart-delay 0
!
interface Serial1/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
!
router ospf 100
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id Loopback0
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
```

```
 stopbits 1
line vty 0 4
 login
!
end
```

P5

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname P5
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
no logging console
!
no aaa new-model
ip source-route
ip cef
!
no ip domain-lookup
!
no ip domain lookup
no ipv6 cef
!
mpls label protocol ldp
mpls traffic-eng tunnels
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 5.5.5.5 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface Serial1/0
 description Connected to P1 s1/3
 ip address 1.2.2.2 255.255.255.252
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp resource-provider none
 ip rsvp bandwidth 1000
 serial restart-delay 0
!
interface Serial1/1
 description Connected to P6 s1/0
 ip address 1.2.3.1 255.255.255.252
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp resource-provider none
 ip rsvp bandwidth 1000
 serial restart-delay 0
!
interface Serial1/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
!
router ospf 100
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id Loopback0
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

P6

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname P6
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
no logging console
!
no aaa new-model
```

```
ip source-route
!
no ip domain-lookup
!
no ip domain lookup
no ipv6 cef
!
mpls label protocol ldp
mpls traffic-eng tunnels
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 6.6.6.6 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface Serial1/0
 description Connected to P5 s1/1
 ip address 1.2.3.2 255.255.255.252
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp resource-provider none
 ip rsvp bandwidth 1000
 serial restart-delay 0
!
interface Serial1/1
 description Connected to P3 s1/2
 ip address 1.3.3.1 255.255.255.252
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp resource-provider none
 ip rsvp bandwidth 1000
 serial restart-delay 0
!
interface Serial1/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
!
router ospf 100
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id Loopback0
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0
!
ip forward-protocol nd
```

```
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

P7

```
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname P7
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
no logging console
!
no aaa new-model
ip source-route
ip cef
!
no ip domain-lookup
!
no ip domain lookup
no ipv6 cef
!
mpls label protocol ldp
mpls traffic-eng tunnels
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 7.7.7.7 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface Serial1/0
 ip address 2.1.1.1 255.255.255.252
 description Connected to P8 s1/1
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp resource-provider none
 ip rsvp bandwidth 1000
 serial restart-delay 0
!
interface Serial1/1
 ip address 1.1.3.1 255.255.255.252
 description Connected to PE1 s1/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp resource-provider none
 ip rsvp bandwidth 1000
 serial restart-delay 0
!
interface Serial1/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
!
router ospf 100
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id Loopback0
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

P8

```
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
```

```
no service password-encryption
!
hostname P8
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
no logging console
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
no ip domain lookup
no ipv6 cef
!
mpls label protocol ldp
mpls traffic-eng tunnels
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 8.8.8.8 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface Serial1/0
 ip address 2.1.2.1 255.255.255.252
 description Connected to P9 s1/1
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp resource-provider none
 ip rsvp bandwidth 1000
 serial restart-delay 0
!
interface Serial1/1
 ip address 2.1.1.2 255.255.255.252
 description Connected to P7 s1/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp resource-provider none
 ip rsvp bandwidth 1000
 serial restart-delay 0
!
interface Serial1/2
 no ip address
 shutdown
 serial restart-delay 0
!
```

```
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
!
router ospf 100
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id Loopback0
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

P9

```
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname P9
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
no logging console
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
no ip domain lookup
no ipv6 cef
!
mpls label protocol ldp
mpls traffic-eng tunnels
multilink bundle-name authenticated
!
```

```
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 9.9.9.9 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface Serial1/0
 ip address 1.1.6.1 255.255.255.252
 description Connected to PE2 s1/1
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp resource-provider none
 ip rsvp bandwidth 1000
 serial restart-delay 0
!
interface Serial1/1
 ip address 2.1.2.2 255.255.255.252
 description Connected to P8 s1/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp resource-provider none
 ip rsvp bandwidth 1000
 serial restart-delay 0
!
interface Serial1/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
!
router ospf 100
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id Loopback0
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
```

```
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

P10

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname P10
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
mpls label protocol ldp
mpls traffic-eng tunnels
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 10.10.10.10 255.255.255.255
!
interface Tunnel0
 ip unnumbered Loopback0
 tunnel destination 1.1.1.1
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 1 1
 tunnel mpls traffic-eng bandwidth  500
 tunnel mpls traffic-eng path-option 1 dynamic
!
interface Tunnel10
 ip unnumbered Loopback0
 tunnel destination 1.1.1.1
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 2 2
 tunnel mpls traffic-eng bandwidth  500
 tunnel mpls traffic-eng path-option 1 explicit name NHOP
!
interface Serial1/0
 description Connected to PE1 s1/1
 ip address 1.1.3.10 255.255.255.252
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp resource-provider none
 ip rsvp bandwidth 1000
 serial restart-delay 0
!
interface Serial1/1
 description Connected to P1 s1/1
 ip address 1.1.2.2 255.255.255.252
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp resource-provider none
 ip rsvp bandwidth 1000
 serial restart-delay 0
!
interface Serial1/2
 description Connected to P4 s1/0
 ip address 1.1.4.2 255.255.255.252
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp resource-provider none
 ip rsvp bandwidth 1000
 serial restart-delay 0
!
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
!
router ospf 100
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 10.10.10.10
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0
!
ip explicit-path name NHOP enable
 next-address 10.10.10.10
 next-address 4.4.4.4
 next-address 1.1.1.1
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

PE1

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE1
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
mpls label protocol ldp
mpls traffic-eng tunnels
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 11.11.11.11 255.255.255.255
!
interface Tunnel12
 ip unnumbered Loopback0
 tunnel destination 33.33.33.33
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 7 7
 tunnel mpls traffic-eng bandwidth  100
 tunnel mpls traffic-eng path-option 10 explicit name P10123
 tunnel mpls traffic-eng path-option 20 explicit name P7892
 tunnel mpls traffic-eng path-option 50 dynamic
!
interface Tunnel123
 ip unnumbered Loopback0
 no ip directed-broadcast
 tunnel destination 33.33.33.33
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng bandwidth 10
 tunnel mpls traffic-eng path-option 10 dynamic
!
interface Tunne456
 ip unnumbered Loopback0
 no ip directed-broadcast
 tunnel destination 33.33.33.33
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng bandwidth 30
 tunnel mpls traffic-eng path-option 40 dynamic
!
interface Tunnel7892
```

```
 ip unnumbered Loopback0
 no ip directed-broadcast
 tunnel destination 33.33.33.33
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng bandwidth 20
 tunnel mpls traffic-eng path-option 50 dynamic
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface Serial1/0
 description Connected to P7 s1/1
 ip address 1.1.3.2 255.255.255.252
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp resource-provider none
 ip rsvp bandwidth 1000
 serial restart-delay 0
!
interface Serial1/1
 description Connected to P10 s1/1
 ip address 1.1.3.9 255.255.255.252
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp resource-provider none
 ip rsvp bandwidth 1000
 serial restart-delay 0
!
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
!
interface FastEthernet2/0
 description Connected to CE1 f1/0
 ip address 10.1.1.2 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet2/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
router ospf 100
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 11.11.11.11
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0
!
ip route 14.1.1.0 255.255.255.0 Tunnel8
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
ip explicit-path name P10123 enable
 next-address 10.10.10.10
 next-address 1.1.1.1
 next-address 2.2.2.2
 next-address 3.3.3.3
 next-address 33.33.33.33
!
ip explicit-path name P7892 enable
 next-address 7.7.7.7
 next-address 8.8.8.8
 next-address 9.9.9.9
 next-address 22.22.22.22
 next-address 33.33.33.33
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

PE2

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE2
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
mpls label protocol ldp
mpls traffic-eng tunnels
!
archive
 log config
  hidekeys
!
```

```
interface Loopback0
 ip address 22.22.22.22 255.255.255.255
!
interface Tunnel332
 ip unnumbered Loopback0
 tunnel destination 11.11.11.11
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 7 7
 tunnel mpls traffic-eng bandwidth  500
 tunnel mpls traffic-eng path-option 1 explicit name 332
 tunnel mpls traffic-eng path-option 50 dynamic
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface Serial1/0
 description Connected to PE3 s1/1
 ip address 1.1.5.2 255.255.255.252
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp resource-provider none
 ip rsvp bandwidth 1000
 serial restart-delay 0
!
interface Serial1/1
 description Connected to P9 s1/0
 ip address 1.1.6.2 255.255.255.252
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp resource-provider none
 ip rsvp bandwidth 1000
 serial restart-delay 0
!
interface Serial1/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/3
 description Connected to P3 s1/0
 ip address 3.3.1.2 255.255.255.252
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp resource-provider none
 ip rsvp bandwidth 1000
 serial restart-delay 0
!
interface FastEthernet2/0
 description Connected to CE2 f1/1
 ip address 13.1.1.2 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet2/1
```

```
 no ip address
 shutdown
 duplex auto
 speed auto
!
router ospf 100
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 22.22.22.22
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0
!
ip route 10.1.1.0 255.255.255.0 Tunnel8
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
ip explicit-path name 332 enable
 next-address 3.3.3.3
 next-address 33.33.33.33
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

PE3

```
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE3
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
mpls label protocol ldp
mpls traffic-eng tunnels
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 33.33.33.33 255.255.255.255
!
interface Tunnel12
 ip unnumbered Loopback0
 tunnel destination 11.11.11.11
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 7 7
 tunnel mpls traffic-eng bandwidth  100
 tunnel mpls traffic-eng path-option 10 explicit name P10123
 tunnel mpls traffic-eng path-option 20 explicit name P7892
 tunnel mpls traffic-eng path-option 50 dynamic
!
interface Tunnel123
 ip unnumbered Loopback0
 no ip directed-broadcast
 tunnel destination 11.11.11.11
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng bandwidth 10
 tunnel mpls traffic-eng path-option 10 dynamic
!
interface Tunne456
 ip unnumbered Loopback0
 no ip directed-broadcast
 tunnel destination 11.11.11.11
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng bandwidth 30
 tunnel mpls traffic-eng path-option 40 dynamic
!
interface Tunnel7892
 ip unnumbered Loopback0
 no ip directed-broadcast
 tunnel destination 11.11.11.11
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng bandwidth 20
 tunnel mpls traffic-eng path-option 50 dynamic
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet2/0
 ip address 14.1.1.2 255.255.255.0
 description Connected to CE3 f1/0
 duplex auto
 speed auto
!
interface Serial1/0
 ip address 1.1.7.2 255.255.255.252
 description Connected to P3 s1/1
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp resource-provider none
 ip rsvp bandwidth 1000
 serial restart-delay 0
!
interface Serial1/1
 ip address 1.1.5.1 255.255.255.252
 description Connected to PE2 s1/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp resource-provider none
 ip rsvp bandwidth 1000
 serial restart-delay 0
!
interface Serial1/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
!
ip forward-protocol nd
!
router ospf 100
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 33.33.33.33
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0
!
no ip http server
no ip http secure-server
!
!!
ip explicit-path name P10123 enable
 next-address 3.3.3.3
 next-address 2.2.2.2
 next-address 1.1.1.1
 next-address 10.10.10.10
 next-address 11.11.11.11
!
ip explicit-path name P7892 enable
 next-address 22.22.22.22
 next-address 9.9.9.9
 next-address 8.8.8.8
 next-address 7.7.7.7
 next-address 11.11.11.11
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
```

```
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

II. Scenario 2: MPLS – Virtual Private Network

II.1 Router interfaces IP list

Router	Interface	IP	connected to	Interface	IP
CE1	f1/0	31.1.1.1	PE2	F1/0	31.1.1.2
	f1/1		CustE site1	f0/0	
	Loopback0	192.168.1.193			
CE2	f1/0	32.1.1.1	PE2	f1/1	32.1.1.2
	f1/1		CustC site2	f0/0	
	Loopback0	192.168.2.129			
CE3	f1/0	33.1.1.1	PE1	f1/0	33.1.1.2
	f1/1		CustD site1	f0/0	
	Loopback0	192.168.1.1			
CE4	f1/0	34.1.1.1	PE1	f1/1	34.1.1.2
	f1/1		CustC site1	f0/0	
	Loopback0	192.168.2.1			
CE5	f1/0	35.1.1.1	PE5	f1/0	35.1.1.2
	f1/1		CustE site2	f0/0	
	Loopback0	192.168.5.1			
CE6	f1/0		CustB site2	f0/0	
	f1/1	36.1.1.1	PE5	f1/1	36.1.1.2
	Loopback0	192.168.6.1			
CE7	f1/0		CustB site1	f0/0	
	f1/1	37.1.1.1	PE4	f1/1	37.1.1.2
	Loopback0	192.168.7.1			
CE8	f1/0	38.1.1.1	PE4	f1/0	38.1.1.2
	f1/1		CustA site1	F0/0	
	Loopback0	192.168.8.1			
CE9	f1/0	39.1.1.1	PE3	f1/0	39.1.1.2
	f1/1		CustD site2	f0/0	
	Loopback0	192.168.2.193			
CE10	f1/0	30.1.1.1	PE3	f1/1	30.1.1.2
	f1/1		CustA site 2	f0/0	
	Loopback0	192.168.1.129			
PE1	f1/0	33.1.1.2	CE3	f1/0	33.1.1.1
	f1/1	34.1.1.2	CE4	f1/0	34.1.1.1
	s2/0	22.1.1.2	P2	s1/0	22.1.1.1
	Loopback0	11.11.11.11			

Router	Interface	IP	connected to	Interface	IP
PE2	f1/0	31.1.1.2	CE1	f1/0	31.1.1.1
	f1/1	32.1.1.2	CE2	f1/0	32.1.1.1
	s2/0	21.1.1.2	P1	s1/0	21.1.1.1
	Loopback0	22.22.22.22			
PE3	f1/0	39.1.1.2	CE9	F1/0	39.1.1.1
	f1/1	30.1.1.2	CE10	F1/0	30.1.1.1
	s2/0	23.2.2.2	P3	s1/1	23.2.2.1
	Loopback0	33.33.33.33			
PE4	f1/0	38.1.1.2	CE8	f1/0	38.1.1.1
	f1/1	37.1.1.2	CE7	f1/1	37.1.1.1
	s2/0	23.1.1.2	P3	s1/0	23.1.1.1
	Loopback0	44.44.44.44			
PE5	f1/0	35.1.1.2	CE5	f1/0	35.1.1.1
	f1/1	36.1.1.2	CE6	f1/1	36.1.1.1
	s2/0	24.1.1.2	P4	s1/0	24.1.1.1
	Loopback0	55.55.55.55			
P1	s1/0	21.1.1.1	PE2	s2/0	21.1.1.2
	g2/0	10.1.1.1	P3	g2/0	10.1.1.2
	g3/0	10.1.1.9	P2	g2/0	10.1.1.10
	Loopback0	1.1.1.1			
P2	s1/0	22.1.1.1	PE1	s2/0	22.1.1.2
	g2/0	10.1.1.10	P1	g3/0	10.1.1.9
	g3/0	10.1.1.5	P4	g2/0	10.1.1.6
	Loopback0	2.2.2.2			
P3	s1/0	23.1.1.1	PE4	s2/0	23.1.1.2
	s1/1	23.2.2.1	PE3	s2/0	23.2.2.2
	g2/0	10.1.1.2	P1	g2/0	10.1.1.1
	g3/0	10.1.1.13	P4	g3/0	10.1.1.14
	Loopback0	3.3.3.3			
P4	s1/0	24.1.1.1	PE5	s2/0	24.1.1.2
	g2/0	10.1.1.6	P2	g3/0	10.1.1.5
	g3/0	10.1.1.14	P3	g3/0	10.1.1.13
	Loopback0	4.4.4.4			

II.2 Dynagen's .net file

```
autostart = False

[localhost:7200]
    workingdir = /tmp
    udp = 10200
      [[7200]]
            image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
            idlepc = 0x607a48c0
            npe = npe-400
            ram = 256
            ghostios = True
            sparsemem = True
    [[ROUTER P1]]
            cnfg = /labs/vpn/config/P1.cfg
        console = 2020
        slot1 = PA-8T
        s1/0 = PE2 s2/0
        slot2 = PA-GE
        g2/0 = P3 g2/0
        slot3 = PA-GE
        g3/0 = P2 g2/0

[localhost:7201]
    workingdir = /tmp
    udp = 10210
      [[7200]]
                image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER P2]]
            cnfg = /labs/vpn/config/P2.cfg
        console = 2021
        slot1 = PA-8T
        s1/0 = PE1 s2/0
        slot2 = PA-GE
        g2/0 = P1 g3/0
        slot3 = PA-GE
        g3/0 = P4 g2/0

[localhost:7202]
    workingdir = /tmp
    udp = 10220
      [[7200]]
                image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER P3]]
            cnfg = /labs/vpn/config/P3.cfg
        console = 2022
        slot1 = PA-8T
        s1/0 = PE4 s2/0
        s1/1 = PE3 s2/0
        slot2 = PA-GE
        g2/0 = P1 g2/0
        slot3 = PA-GE
        g3/0 = P4 g3/0

[localhost:7203]
    workingdir = /tmp
    udp = 10230
      [[7200]]
                image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER P4]]
            cnfg = /labs/vpn/config/P4.cfg
        console = 2023
        slot1 = PA-8T
        s1/0 = PE5 s2/0
        slot2 = PA-GE
        g2/0 = P2 g3/0
        slot3 = PA-GE
        g3/0 = P3 g3/0

####################################################################################################

[localhost:7204]
    workingdir = /tmp
    udp = 10250
      [[7200]]
                image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER PE1]]
            cnfg = /labs/vpn/config/PE1.cfg
        console = 2025
        slot1 = PA-2FE-TX
        f1/0 = CE3 f1/0
        f1/1 = CE4 f1/0
        slot2 = PA-8T
        s2/0 = P2 s1/0

[localhost:7205]
    workingdir = /tmp
    udp = 10260
      [[7200]]
                image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER PE2]]
            cnfg = /labs/vpn/config/PE2.cfg
        console = 2026
        slot1 = PA-2FE-TX
        f1/0 = CE1 f1/0
        f1/1 = CE2 f1/0
        slot2 = PA-8T
        s2/0 = P1 s1/0

[localhost:7206]
    workingdir = /tmp
    udp = 10270
      [[7200]]
                image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER PE3]]
            cnfg = /labs/vpn/config/PE3.cfg
        console = 2027
        slot1 = PA-2FE-TX
        f1/0 = CE9 f1/0
        f1/1 = CE10 f1/0
        slot2 = PA-8T
        s2/0 = P3 s1/1

[localhost:7207]
    workingdir = /tmp
    udp = 10280
      [[7200]]
                image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER PE4]]
            cnfg = /labs/vpn/config/PE4.cfg
        console = 2028
        slot1 = PA-2FE-TX
        f1/0 = CE8 f1/0
        f1/1 = CE7 f1/1
        slot2 = PA-8T
        s2/0 = P3 s1/0

[localhost:7208]
    workingdir = /tmp
    udp = 10290
      [[7200]]
                image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER PE5]]
            cnfg = /labs/vpn/config/PE5.cfg
        console = 2029
        slot1 = PA-2FE-TX
        f1/0 = CE5 f1/0
        f1/1 = CE6 f1/1
        slot2 = PA-8T
        s2/0 = P4 s1/0

##########################################################################################
#############
[localhost:7209]
    workingdir = /tmp
    udp = 10300
      [[7200]]
                image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER CE1]]
            cnfg = /labs/vpn/config/CE1.cfg
        console = 2030
        slot1 = PA-2FE-TX

        slot2 = PA-2FE-TX

[localhost:7210]
    workingdir = /tmp
    udp = 10310
      [[7200]]
                image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True

    [[ROUTER CE2]]
            cnfg = /labs/vpn/config/CE2.cfg
        console = 2031
        slot1 = PA-2FE-TX
        f1/0 = PE2 f1/1

[localhost:7211]
    workingdir = /tmp
    udp = 10320
      [[7200]]
                image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER CE3]]
            cnfg = /labs/vpn/config/CE3.cfg
        console = 2032
        slot1 = PA-2FE-TX
        f1/0 = PE1 f1/0

[localhost:7212]
    workingdir = /tmp
    udp = 10330
      [[7200]]
                image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER CE4]]
            cnfg = /labs/vpn/config/CE4.cfg
        console = 2033
        slot1 = PA-2FE-TX
        f1/0 = PE1 f1/1

[localhost:7013]
    workingdir = /tmp
    udp = 10340
      [[7200]]
```

```
            image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
            idlepc = 0x607a48c0
            npe = npe-400
            ram = 256
            ghostios = True
            sparsemem = True
    [[ROUTER CE5]]
          cnfg = /labs/vpn/config/CE5.cfg
        console = 2034
        slot1 = PA-2FE-TX
        f1/0 = PE5 f1/0


[localhost:7214]
    workingdir = /tmp
    udp = 10350
      [[7200]]
            image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
            idlepc = 0x607a48c0
            npe = npe-400
            ram = 256
            ghostios = True
            sparsemem = True
    [[ROUTER CE6]]
          cnfg = /labs/vpn/config/CE6.cfg
        console = 2035
        slot1 = PA-2FE-TX

        f1/1 = PE5 f1/1
        slot2 = PA-2FE-TX

[localhost:7215]
    workingdir = /tmp
    udp = 10360
      [[7200]]
            image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
            idlepc = 0x607a48c0
            npe = npe-400
            ram = 256
            ghostios = True
            sparsemem = True
    [[ROUTER CE7]]
          cnfg = /labs/vpn/config/CE7.cfg
        console = 2036
        slot1 = PA-2FE-TX

        f1/1 = PE4 f1/1

[localhost:7216]
    workingdir = /tmp
    udp = 10370
      [[7200]]
            image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
            idlepc = 0x607a48c0
            npe = npe-400
            ram = 256
            ghostios = True
            sparsemem = True
    [[ROUTER CE8]]
          cnfg = /labs/vpn/config/CE8.cfg
        console = 2037
        slot1 = PA-2FE-TX
        f1/0 = PE4 f1/0

[localhost:7217]
    workingdir = /tmp
    udp = 10380
      [[7200]]
                image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER CE9]]
            cnfg = /labs/vpn/config/CE9.cfg
        console = 2038
        slot1 = PA-2FE-TX
        f1/0 = PE3 f1/0

[localhost:7218]
    workingdir = /tmp
    udp = 10390
      [[7200]]
                image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER CE10]]
            cnfg = /labs/vpn/config/CE10.cfg
        console = 2039
        slot1 = PA-2FE-TX
        f1/0 = PE3 f1/1
```

II.3 Dynamips Start Script

```
#!/bin/bash
sudo dynamips -H 7200 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7201 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7202 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7203 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7204 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7205 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7206 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7207 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7208 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7209 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7210 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7211 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7212 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7213 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7214 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7215 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7216 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7217 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7218 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked &
sudo dynamips -H 7219 /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
```

II.4 Router configurations

CE1

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 192.168.1.193 255.255.255.64
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no shutdown
 ip address 31.1.1.1 255.255.255.0
 description Connected to PE2 f1/0
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet2/0
 no ip address
 shutdown
 duplex auto
 speed auto
```

```
!
interface FastEthernet2/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
router bgp 100
 no synchronization
 bgp log-neighbor-changes
 network 0.0.0.0
 network 192.168.1.128 mask 255.255.255.64
 neighbor 31.1.1.2 remote-as 111
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

CE2

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
```

```
  hidekeys
!
interface Loopback0
 ip address 192.168.2.129 255.255.255.64
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no shutdown
 ip address 32.1.1.1 255.255.255.0
 description Connected to PE2 f1/1
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

CE3

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 192.168.1.1 255.255.255.128
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no shutdown
 ip address 33.1.1.1 255.255.255.0
 description Connected to PE1 f1/0
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
router bgp 100
 no synchronization
 bgp log-neighbor-changes
 network 0.0.0.0
 network 192.168.1.0 mask 255.255.255.128
 neighbor 33.1.1.2 remote-as 111
 maximum-paths 2
 no auto-summary
!
ip http server
no ip http secure-server
ip route 0.0.0.0 0.0.0.0 Null0
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

CE4

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 192.168.2.1 255.255.255.128
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no shutdown
 ip address 34.1.1.1 255.255.255.0
 description Connected to PE1 f1/1
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
router eigrp 222
 network 34.1.1.0 0.0.0.255
 network 192.168.2.0 0.0.0.127
 network 0.0.0.0
 distribute-list prefix LocalOnly out
 no auto-summary
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
```

```
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

CE5

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no shutdown
 ip address 35.1.1.1 255.255.255.0
 description Connected to PE5 f1/0
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no ip address
 shutdown
 duplex auto
 speed auto
```

```
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

CE6

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet1/1
```

```
 no shutdown
 ip address 36.1.1.1 255.255.255.0
 description Connected to PE5 f1/1
 duplex auto
 speed auto
!
interface FastEthernet2/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet2/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

CE7

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
```

```
archive
 log config
  hidekeys
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no shutdown
 ip address 37.1.1.1 255.255.255.0
 description Connected to PE4 f1/1
 duplex auto
 speed auto
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

CE8

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
```

```
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no shutdown
 ip address 38.1.1.1 255.255.255.0
 description Connected to PE4 f1/0
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no shutdown
 no ip address
 shutdown
 duplex auto
 speed auto
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

CE9

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
```

```
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 192.168.2.193 255.255.255.64
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no shutdown
 ip address 39.1.1.1 255.255.255.0
 description Connected to PE3 f1/0
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no ip address
 shutdown
 duplex auto
 speed auto
!

ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

CE10

```
!
upgrade fpd auto
```

```
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 192.168.1.129 255.255.255.64
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no shutdown
 ip address 30.1.1.1 255.255.255.0
 description Connected to PE3 f1/1
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
router bgp 100
 no synchronization
 bgp log-neighbor-changes
 network 0.0.0.0
 network 192.168.1.128 mask 255.255.255.64
 neighbor 30.1.1.2 remote-as 111
!
 maximum-paths 2
 no auto-summary
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
```

```
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

P1

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname P1
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
no logging console
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
mpls label protocol ldp
mpls traffic-eng tunnels
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 1.1.1.1 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface Serial1/0
 no shutdown
 ip address 21.1.1.1 255.255.255.252
 description Connected to PE2 s2/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 serial restart-delay 0
!
interface Serial1/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
!
interface GigabitEthernet2/0
 no shutdown
 ip address 10.1.1.1 255.255.255.252
 description Connected to P3 g2/0
 half-duplex
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 negotiation auto
!
interface GigabitEthernet3/0
 no shutdown
 ip address 10.1.1.9 255.255.255.252
 description Connected to P2 g2/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 negotiation auto
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 1.1.1.1
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0
!
!
router bgp 111
 no synchronization
 bgp log-neighbor-changes
 neighbor 11.11.11.11 remote-as 111
 neighbor 11.11.11.11 update-source Loopback0
 neighbor 11.11.11.11 route-reflector-client
 neighbor 22.22.22.22 remote-as 111
 neighbor 22.22.22.22 update-source Loopback0
 neighbor 22.22.22.22 route-reflector-client
 neighbor 33.33.33.33 remote-as 111
 neighbor 33.33.33.33 update-source Loopback0
 neighbor 33.33.33.33 route-reflector-client
 neighbor 44.44.44.44 remote-as 111
 neighbor 44.44.44.44 update-source Loopback0
 neighbor 44.44.44.44 route-reflector-client
 neighbor 55.55.55.55 remote-as 111
 neighbor 55.55.55.55 update-source Loopback0
 neighbor 55.55.55.55 route-reflector-client
 no auto-summary
 !
 address-family vpnv4
  neighbor 11.11.11.11 activate
  neighbor 11.11.11.11 send-community extended
  neighbor 11.11.11.11 route-reflector-client
  neighbor 22.22.22.22 activate
  neighbor 22.22.22.22 send-community extended
  neighbor 22.22.22.22 route-reflector-client
  neighbor 133.33.33.33 activate
  neighbor 33.33.33.33 send-community extended
  neighbor 33.33.33.33 route-reflector-client
  neighbor 44.44.44.44 activate
  neighbor 44.44.44.44 send-community extended
  neighbor 44.44.44.44 route-reflector-client
  neighbor 55.55.55.55 activate
  neighbor 55.55.55.55 send-community extended
  neighbor 55.55.55.55 route-reflector-client
 exit-address-family
!
logging alarm informational
!
mpls ldp router-id Loopback0 force
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

P2

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname P2
```

```
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
no logging console
!
no aaa new-model
ip source-route
ip cef
!
mpls label protocol ldp
mpls traffic-eng tunnels
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 2.2.2.2 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface Serial1/0
 no shutdown
 ip address 22.1.1.1 255.255.255.252
 description Connected to PE1 s2/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 serial restart-delay 0
!
interface Serial1/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
!
interface GigabitEthernet2/0
 no shutdown
 ip address 10.1.1.10 255.255.255.252
 description Connected to P1 g3/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 negotiation auto
!
interface GigabitEthernet3/0
 no shutdown
 ip address 10.1.1.5 255.255.255.252
 description Connected to P4 g2/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 negotiation auto
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 2.2.2.2
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0
!
logging alarm informational
!
mpls ldp router-id Loopback0 force
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

P3

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname P3
!
boot-start-marker
boot-end-marker
!
```

```
logging message-counter syslog
no logging console
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
mpls label protocol ldp
mpls traffic-eng tunnels
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 3.3.3.3 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface Serial1/0
 no shutdown
 ip address 23.1.1.1 255.255.255.252
 description Connected to PE4 s2/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 serial restart-delay 0
!
interface Serial1/1
 no shutdown
 ip address 23.2.2.1 255.255.255.252
 description Connected to PE3 s2/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 serial restart-delay 0
!
interface Serial1/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
!
interface GigabitEthernet2/0
 no shutdown
 ip address 10.1.1.2 255.255.255.252
 description Connected to P1 g2/0
 half-duplex
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 negotiation auto
!
interface GigabitEthernet3/0
 no shutdown
 ip address 10.1.1.13 255.255.255.252
 description Connected to P4 g3/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 negotiation auto
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 3.3.3.3
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0
!
logging alarm informational
!
mpls ldp router-id Loopback0 force
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

P4

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname P4
!
```

```
boot-start-marker
boot-end-marker
!
logging message-counter syslog
no logging console
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
mpls label protocol ldp
mpls traffic-eng tunnels
!
multilink bundle-name authenticated
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
mpls label protocol ldp
mpls traffic-eng tunnels
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 4.4.4.4 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface Serial1/0
 no shutdown
 ip address 24.1.1.1 255.255.255.252
 description Connected to PE5 s2/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 serial restart-delay 0
!
interface Serial1/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/3
```

```
 no ip address
 shutdown
 serial restart-delay 0
!
interface GigabitEthernet2/0
 no shutdown
 ip address 10.1.1.6 255.255.255.252
 description Connected to P2 g3/0
 half-duplex
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 negotiation auto
!
interface GigabitEthernet3/0
 no shutdown
 ip address 10.1.1.14 255.255.255.252
 description Connected to P3 g3/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 negotiation auto
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 4.4.4.4
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0
!
router bgp 100
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.0.2.1 remote-as 100
 neighbor 10.0.2.1 update-source Loopback0
 neighbor 10.0.2.1 route-reflector-client
 neighbor 10.0.2.2 remote-as 100
 neighbor 10.0.2.2 update-source Loopback0
 neighbor 10.0.2.2 route-reflector-client
 neighbor 10.0.2.3 remote-as 100
 neighbor 10.0.2.3 update-source Loopback0
 neighbor 10.0.2.3 route-reflector-client
 neighbor 10.0.2.4 remote-as 100
 neighbor 10.0.2.4 update-source Loopback0
 neighbor 10.0.2.4 route-reflector-client
 no auto-summary
 !
 address-family vpnv4
  neighbor 10.0.2.1 activate
  neighbor 10.0.2.1 send-community extended
  neighbor 10.0.2.1 route-reflector-client
  neighbor 10.0.2.2 activate
  neighbor 10.0.2.2 send-community extended
  neighbor 10.0.2.2 route-reflector-client
  neighbor 10.0.2.3 activate
  neighbor 10.0.2.3 send-community extended
  neighbor 10.0.2.3 route-reflector-client
  neighbor 10.0.2.4 activate
  neighbor 10.0.2.4 send-community extended
  neighbor 10.0.2.4 route-reflector-client
 exit-address-family
!
ip http server
no ip http secure-server
!
mpls ldp router-id Loopback0 force
!
control-plane
!
line con 0
 exec-timeout 35791 0
line aux 0
line vty 0 4
 login
!
end
```

PE1

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE3
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
no logging console
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
ip vrf Net1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
!
ip vrf Net2
 rd 200:10
 route-target export 200:0
 route-target import 200:1
 route-target import 200:2
!
mpls label protocol ldp
mpls traffic-eng tunnels
```

```
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 11.11.11.11 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no shutdown
 ip address 33.1.1.2 255.255.255.0
 description Connected to CE3 f1/0
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no shutdown
 ip address 34.1.1.2 255.255.255.0
 description Connected to CE4 f1/0
 duplex auto
 speed auto
!
interface Serial2/0
 no shutdown
 ip address 22.1.1.2 255.255.255.252
 description Connected to P2 s1/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 serial restart-delay 0
!
interface Serial2/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/3
 no ip address
 shutdown
 serial restart-delay 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
router ospf 1
```

```
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 11.11.11.11
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0
!
router bgp 111
 no synchronization
 bgp log-neighbor-changes
 neighbor 1.1.1.1 remote-as 111
 neighbor 1.1.1.1 update-source Loopback0
 no auto-summary
!
 address-family vpnv4
  neighbor 1.1.1.1 activate
  neighbor 1.1.1.1 send-community both
 exit-address-family
 !
 address-family ipv4 vrf Net1
  redistribute connected
  neighbor 1.1.1.1 remote-as 111
  neighbor 1.1.1.1 activate
  maximum-paths eibgp 2 import 2
  no synchronization
 exit-address-family
!
 !
 address-family ipv4 vrf Net2
  redistribute connected
  neighbor 1.1.1.1 remote-as 111
  neighbor 1.1.1.1 activate
  maximum-paths eibgp 2 import 2
  no synchronization
 exit-address-family
 !
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

PE2

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Pe2
```

```
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
no logging console
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
ip vrf Net1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
!
ip vrf Net2
 rd 200:1
 route-target export 200:1
 route-target import 200:0
!
mpls label protocol ldp
mpls traffic-eng tunnels
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 22.22.22.22 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no shutdown
 ip address 31.1.1.2 255.255.255.0
 description Connected to CE1 f1/0
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no shutdown
 ip address 32.1.1.2 255.255.255.0
 description Connected to CE2 f1/0
 duplex auto
 speed auto
!
interface Serial2/0
 no shutdown
 ip address 21.1.1.2 255.255.255.252
 description Connected to P1 s1/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 serial restart-delay 00
!
interface Serial2/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/3
 no ip address
 shutdown
 serial restart-delay 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 22.22.22.22
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0
!
router bgp 111
 no synchronization
 bgp log-neighbor-changes
 neighbor 1.1.1.1 remote-as 111
 neighbor 1.1.1.1 update-source Loopback0
 no auto-summary
!
 address-family vpnv4
  neighbor 1.1.1.1 activate
  neighbor 1.1.1.1 send-community both
 exit-address-family
 !
 address-family ipv4 vrf Net1
  redistribute connected
  neighbor 1.1.1.1 remote-as 111
  neighbor 1.1.1.1 activate
  maximum-paths eibgp 2 import 2
  no synchronization
 exit-address-family
!
 address-family ipv4 vrf Net2
  redistribute connected
  neighbor 1.1.1.1 remote-as 111
  neighbor 1.1.1.1 activate
  maximum-paths eibgp 2 import 2
  no synchronization
 exit-address-family
 !
logging alarm informational
```

```
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

PE3

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE1
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
no logging console
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
ip vrf Net1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
!
ip vrf Net2
 rd 200:2
 route-target export 200:2
 route-target import 200:0
!
mpls label protocol ldp
mpls traffic-eng tunnels
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 33.33.33.33 255.255.255.255
!
interface FastEthernet0/0
```

```
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no shutdown
 ip address 39.1.1.2 255.255.255.0
 description Connected to CE9 f1/0
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no shutdown
 ip address 30.1.1.2 255.255.255.0
 description Connected to CE10 f1/0
 duplex auto
 speed auto
!
interface Serial2/0
 no shutdown
 ip address 23.2.2.2 255.255.255.252
 description Connected to P3 s1/1
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 serial restart-delay 0
!
interface Serial2/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/3
 no ip address
 shutdown
 serial restart-delay 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 33.33.33.33
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0
!
router bgp 111
 no synchronization
 bgp log-neighbor-changes
 neighbor 1.1.1.1 remote-as 111
 neighbor 1.1.1.1 update-source Loopback0
 no auto-summary
!
 address-family vpnv4
  neighbor 1.1.1.1 activate
  neighbor 1.1.1.1 send-community both
 exit-address-family
 !
 address-family ipv4 vrf Net1
  redistribute connected
  neighbor 1.1.1.1 remote-as 111
  neighbor 1.1.1.1 activate
  maximum-paths eibgp 2 import 2
  no synchronization
 exit-address-family
!
 address-family ipv4 vrf Net2
  redistribute connected
  neighbor 1.1.1.1 remote-as 111
  neighbor 1.1.1.1 activate
  maximum-paths eibgp 2 import 2
  no synchronization
 exit-address-family
 !
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

PE4

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE4
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
```

```
ip vrf Net4
 rd 100:140
 route-target export 1:400
 route-target import 1:400
!
ip vrf Net5
 rd 100:150
 route-target export 1:500
 route-target import 1:500
!
ip vrf Net6
 rd 100:160
 route-target export 1:600
 route-target import 1:600
!
mpls label protocol ldp
mpls traffic-eng tunnels
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 44.44.44.44 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no shutdown
 ip vrf forwarding Net4
 ip address 38.1.1.2 255.255.255.0
 description Connected to CE8 f1/0
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no shutdown
 ip vrf forwarding Net5
 ip address 37.1.1.2 255.255.255.0
 description Connected to CE7 f1/1
 duplex auto
 speed auto
!
interface Serial2/0
 no shutdown
 ip address 23.1.1.2 255.255.255.252
 description Connected to P3 s1/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 serial restart-delay 00
!
interface Serial2/1
```

```
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/3
 no ip address
 shutdown
 serial restart-delay 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
ip route vrf Net4  192.168.8.0 255.255.255.0  38.1.1.2
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 44.44.44.44
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0
!
router ospf 6 vrf Net6
router¬≠id 44.44.44.44
network 35.1.10 0.255.255.255 area 0
network 38.1.10 0.255.255.255 area 0
redistribute bgp 6 subnets
!
router rip
version 2
address¬≠family ipv4 vrf Net5
redistribute bgp 5 metric transparent
network 35.1.1.0
network 37.1.1.0
no auto¬≠summary
exit¬≠address¬≠family
!
router bgp 4
 bgp log-neighbor-changes
 address¬≠family ipv4 vrf Net4
 neighbor 55.55.55.55 remote-as 4
 neighbor 55.55.55.55 update-source Loopback0
 no auto-summary
 redistribute static
 redistribute connected
!
router bgp 5
 address¬≠family ipv4 vrf Net5
 bgp log-neighbor-changes
 neighbor 55.55.55.55 remote-as 5
 neighbor 55.55.55.55 update-source Loopback0
 no auto-summary
 redistribute rip
!
router bgp 6
```

```
address¬≠family ipv4 vrf Net6
redistribute ospf 6 vrf Net6 match internal external 1 external 2
!
logging alarm informational
!
control-plane
!
 address-family vpnv4
 neighbor 55.55.55.55 activate
 neighbor 55.55.55.55 send-community both
 exit-address-family
!
address-family ipv4 vrf Net4
 redistribute connected
 no auto-summary
 no synchronization
 exit-address-family
 !
address-family ipv4 vrf Net6
 redistribute connected
 no auto-summary
 no synchronization
 exit-address-family
 !
ip classless
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

PE5

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE5
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
ip vrf Net4
 rd 100:140
 route-target export 1:400
 route-target import 1:400
!
ip vrf Net5
 rd 100:150
 route-target export 1:500
 route-target import 1:500
!
ip vrf Net6
 rd 100:160
 route-target export 1:600
 route-target import 1:600
!
no ipv6 cef
!
mpls label protocol ldp
mpls traffic-eng tunnels
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 55.55.55.55 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no shutdown
 ip vrf forwarding Net5
 ip address 35.1.1.2 255.255.255.0
 description Connected to CE5 f1/0
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no shutdown
 ip vrf forwarding Net4
 ip address 36.1.1.2 255.255.255.0
 description Connected to CE6 f1/1
 duplex auto
 speed auto
!
interface Serial2/0
 no shutdown
 ip address 24.1.1.2 255.255.255.252
 description Connected to P4 s1/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 serial restart-delay 0
!
interface Serial2/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/3
 no ip address
 shutdown
 serial restart-delay 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
ip route vrf Net4 192.168.8.0 255.255.255.0  36.1.1.2
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 55.55.55.55
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0
!
router ospf 6 vrf Net6
router¬≠id 55.55.55.55
network 35.1.10 0.255.255.255 area 0
network 38.1.10 0.255.255.255 area 0
redistribute bgp 6 subnets
!
router rip
version 2
address¬≠family ipv4 vrf Net5
network 35.1.1.0
network 37.1.1.0
redistribute bgp 5 metric transparent
no auto¬≠summary
exit¬≠address¬≠family
!
router bgp 4
 bgp log-neighbor-changes
 neighbor 44.44.44.44 remote-as 4
 neighbor 44.44.44.44 update-source Loopback0
 no auto-summary
 redistribute static
 redistribute connected
!
router bgp 5
 address¬≠family ipv4 vrf Net5
 bgp log-neighbor-changes
 neighbor 44.44.44.44 remote-as 5
 neighbor 44.44.44.44 update-source Loopback0
 no auto-summary
 redistribute rip
!
router bgp 6
address¬≠family ipv4 vrf Net6
redistribute ospf 6 vrf Net6 match internal external 1 external 2
```

```
!
logging alarm informational
!
control-plane
!
 address-family vpnv4
 neighbor 44.44.44.44 activate
 neighbor 44.44.44.44 send-community both
 exit-address-family
!
address-family ipv4 vrf Net4
 redistribute connected
 no auto-summary
 no synchronization
 exit-address-family
 !
address-family ipv4 vrf Net5
 redistribute connected
 no auto-summary
 no synchronization
 exit-address-family
 !
address-family ipv4 vrf Net6
 redistribute connected
 no auto-summary
 no synchronization
 exit-address-family
 !
ip classless
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

III. Scenario 3: MPLS – IPv6

III.1 Router interfaces IP list

III.1.1 IPv6

Router	Interface	IP	connected to	Interface	IP
CE1-AS2	f1/0	31.1.1.1	PE1-AS2	f1/0	31.1.1.2
	Loopback0	192.168.1.193			
CE2-AS2	f1/0	32.1.1.1	PE1-AS2	f1/1	32.1.1.2
	Loopback0	192.168.2.129			
CE3-AS2	f1/0	33.1.1.1	PE2-AS2	f1/0	33.1.1.2
	Loopback0	192.168.1.1			
CE4-AS2	f1/0	34.1.1.1	PE2-AS2	f1/1	34.1.1.2
	Loopback0	192.168.2.1			
CE5-AS2	f1/1	36.1.1.1	PE2-AS2	f0/0	36.1.1.2
	Loopback0	192.168.4.1			
CE5-AS1	f1/1	37.1.1.1	PE2-AS1	f1/1	37.1.1.2
	Loopback0	192.168.4.129			
CE4-AS1	f1/0	38.1.1.1	PE2-AS1	f1/0	38.1.1.2
	Loopback0	192.168.3.129			
CE3-AS1	f1/0	39.1.1.1	PE1-AS1	f1/0	39.1.1.2
	Loopback0	192.168.2.193			
CE2-AS1	f1/0	30.1.1.1	PE1-AS1	f1/1	30.1.1.2
	Loopback0	192.168.1.129			
CE1-AS1	f1/0	35.1.1.1	PE1-AS1	f0/0	35.1.1.2
	Loopback0	192.168.3.1			
PE2-AS2	f1/0	33.1.1.2	CE3-AS2	f1/0	33.1.1.1
	f1/1	34.1.1.2	CE4-AS2	f1/0	34.1.1.1
	f0/0	36.1.1.2	CE5-AS2	f1/0	36.1.1.1
	s2/0	22.1.1.2	P1-AS2	s1/0	22.1.1.1
	Loopback0	11.11.11.11			
PE1-AS2	f1/0	31.1.1.2	CE1-AS2	f1/0	31.1.1.1
	f1/1	32.1.1.2	CE2-AS2	f1/0	32.1.1.1
	s2/0	21.1.1.2	P1-AS2	s1/0	21.1.1.1
	Loopback0	22.22.22.22			
PE1-AS1	f1/0	39.1.1.2	CE3-AS1	f1/0	39.1.1.1
	f1/1	30.1.1.2	CE2-AS1	f1/0	30.1.1.1
	f0/0	35.1.1.2	CE1-AS1	f1/0	35.1.1.1
	s2/0	23.2.2.2	P1-AS1	s1/1	23.2.2.1
	Loopback0	33.33.33.33			

PE2-AS1	f1/0	38.1.1.2	CE4-AS1	f1/0	38.1.1.1
	f1/1	37.1.1.2	CE5-AS1	f1/1	37.1.1.1
	s2/0	23.1.1.2	P1-AS1	s1/0	23.1.1.1
	Loopback0	44.44.44.44			
P1-AS2	s1/0	22.1.1.1	PE2-AS2	s2/0	22.1.1.2
	s1/1	21.1.1.1	PE1-AS2	s2/0	21.1.1.2
	g2/0	10.1.1.1	P1-AS1	g2/0	10.1.1.2
	Loopback0	2.2.2.2			
P1-AS1	s1/0	23.1.1.1	PE2-AS1	s2/0	23.1.1.2
	s1/1	23.2.2.1	PE1-AS1	s2/0	23.2.2.2
	g2/0	10.1.1.2	P1-AS2	g2/0	10.1.1.1
	Loopback0	3.3.3.3			

III.1.2 IAS

Router	Interface	IP	connected to	Interface	IP
CE1-AS2	f1/0	31.1.1.1	PE1-AS2	f1/0	31.1.1.2
	Loopback0	192.168.1.193			
CE2-AS2	f1/0	32.1.1.1	PE1-AS2	f1/1	32.1.1.2
	Loopback0	192.168.2.129			
CE3-AS2	f1/0	33.1.1.1	PE2-AS2	f1/0	33.1.1.2
	Loopback0	192.168.1.1			
CE4-AS2	f1/0	34.1.1.1	PE2-AS2	f1/1	34.1.1.2
	Loopback0	192.168.2.1			
CE5-AS2	f1/1	36.1.1.1	PE2-AS2	f0/0	36.1.1.2
	Loopback0	192.168.4.1			
CE5-AS1	f1/1	37.1.1.1	PE2-AS1	f1/1	37.1.1.2
	Loopback0	192.168.4.129			
CE4-AS1	f1/0	38.1.1.1	PE2-AS1	f1/0	38.1.1.2
	Loopback0	192.168.3.129			
CE3-AS1	f1/0	39.1.1.1	PE1-AS1	f1/0	39.1.1.2
	Loopback0	192.168.2.193			
CE2-AS1	f1/0	30.1.1.1	PE1-AS1	f1/1	30.1.1.2
	Loopback0	192.168.1.129			
CE1-AS1	f1/0	35.1.1.1	PE1-AS1	f0/0	35.1.1.2
	Loopback0	192.168.3.1			
PE2-AS2	f1/0	33.1.1.2	CE3-AS2	f1/0	33.1.1.1
	f1/1	34.1.1.2	CE4-AS2	f1/0	34.1.1.1
	f0/0	36.1.1.2	CE5-AS2	f1/0	36.1.1.1
	s2/0	22.1.1.2	P1-AS2	s1/0	22.1.1.1
	Loopback0	11.11.11.11			
PE1-AS2	f1/0	31.1.1.2	CE1-AS2	f1/0	31.1.1.1
	f1/1	32.1.1.2	CE2-AS2	f1/0	32.1.1.1

	s2/0	21.1.1.2	P1-AS2	s1/0	21.1.1.1
	Loopback0	22.22.22.22			
PE1-AS1	f1/0	39.1.1.2	CE3-AS1	f1/0	39.1.1.1
	f1/1	30.1.1.2	CE2-AS1	f1/0	30.1.1.1
	f0/0	35.1.1.2	CE1-AS1	f1/0	35.1.1.1
	s2/0	23.2.2.2	P1-AS1	s1/1	23.2.2.1
	Loopback0	33.33.33.33			
PE2-AS1	f1/0	38.1.1.2	CE4-AS1	f1/0	38.1.1.1
	f1/1	37.1.1.2	CE5-AS1	f1/1	37.1.1.1
	s2/0	23.1.1.2	P1-AS1	s1/0	23.1.1.1
	Loopback0	44.44.44.44			
P1-AS2	s1/0	22.1.1.1	PE2-AS2	s2/0	22.1.1.2
	s1/1	21.1.1.1	PE1-AS2	s2/0	21.1.1.2
	g2/0	10.1.1.1	ASBR-AS2	g2/0	10.1.1.2
	Loopback0	2.2.2.2			
P1-AS1	s1/0	23.1.1.1	PE2-AS1	s2/0	23.1.1.2
	s1/1	23.2.2.1	PE1-AS1	s2/0	23.2.2.2
	g2/0	10.2.1.1	ASBR-AS1	g2/0	10.2.1.2
	Loopback0	3.3.3.3			
ASBR-AS1	g2/0	10.2.1.2	P1-AS1	g2/0	10.2.1.1
	g3/0	10.1.2.2	ASBR-AS2	g3/0	10.1.2.1
	Loopback0	4.4.4.4			
ASBR-AS2	g2/0	10.1.1.2	P1-AS2	g2/0	10.1.1.1
	g3/0	10.1.2.1	ASBR-AS1	g3/0	10.1.2.2
	Loopback0	5.5.5.5			

III.1.2 CsC

Router	Interface	IP	connected to	Interface	IP
CE1-AS2	f1/0	31.1.1.1	PE1-AS2	f1/0	31.1.1.2
	Loopback0	192.168.1.193			
CE2-AS2	f1/0	32.1.1.1	PE1-AS2	f1/1	32.1.1.2
	Loopback0	192.168.2.129			
CE3-AS2	f1/0	33.1.1.1	PE2-AS2	f1/0	33.1.1.2
	Loopback0	192.168.1.1			
CE4-AS2	f1/0	34.1.1.1	PE2-AS2	f1/1	34.1.1.2
	Loopback0	192.168.2.1			
CE5-AS2	f1/1	36.1.1.1	PE2-AS2	f0/0	36.1.1.2
	Loopback0	192.168.4.1			
CE5-AS1	f1/1	37.1.1.1	PE2-AS1	f1/1	37.1.1.2
	Loopback0	192.168.4.129			
CE4-AS1	f1/0	38.1.1.1	PE2-AS1	f1/0	38.1.1.2

	Loopback0	192.168.3.129			
CE3-AS1	f1/0	39.1.1.1	PE1-AS1	f1/0	39.1.1.2
	Loopback0	192.168.2.193			
CE2-AS1	f1/0	30.1.1.1	PE1-AS1	f1/1	30.1.1.2
	Loopback0	192.168.1.129			
CE1-AS1	f1/0	35.1.1.1	PE1-AS1	f0/0	35.1.1.2
	Loopback0	192.168.3.1			
PE2-AS2	f1/0	33.1.1.2	CE3-AS2	f1/0	33.1.1.1
	f1/1	34.1.1.2	CE4-AS2	f1/0	34.1.1.1
	f0/0	36.1.1.2	CE5-AS2	f1/0	36.1.1.1
	s2/0	22.1.1.2	P1-AS2	s1/0	22.1.1.1
	Loopback0	11.11.11.11			
PE1-AS2	f1/0	31.1.1.2	CE1-AS2	f1/0	31.1.1.1
	f1/1	32.1.1.2	CE2-AS2	f1/0	32.1.1.1
	s2/0	21.1.1.2	P1-AS2	s1/0	21.1.1.1
	Loopback0	22.22.22.22			
PE1-AS1	f1/0	39.1.1.2	CE3-AS1	f1/0	39.1.1.1
	f1/1	30.1.1.2	CE2-AS1	f1/0	30.1.1.1
	f0/0	35.1.1.2	CE1-AS1	f1/0	35.1.1.1
	s2/0	23.2.2.2	P1-AS1	s1/1	23.2.2.1
	Loopback0	33.33.33.33			
PE2-AS1	f1/0	38.1.1.2	CE4-AS1	f1/0	38.1.1.1
	f1/1	37.1.1.2	CE5-AS1	f1/1	37.1.1.1
	s2/0	23.1.1.2	P1-AS1	s1/0	23.1.1.1
	Loopback0	44.44.44.44			
P1-AS2	s1/0	22.1.1.1	PE2-AS2	s2/0	22.1.1.2
	s1/1	21.1.1.1	PE1-AS2	s2/0	21.1.1.2
	g2/0	10.1.1.1	ASBR-AS2	g2/0	10.1.1.2
	Loopback0	2.2.2.2			
P1-AS1	s1/0	23.1.1.1	PE2-AS1	s2/0	23.1.1.2
	s1/1	23.2.2.1	PE1-AS1	s2/0	23.2.2.2
	g2/0	10.2.1.1	ASBR-AS1	g2/0	10.2.1.2
	Loopback0	3.3.3.3			
ASBR-AS1	g2/0	10.2.1.2	P1-AS1	g2/0	10.2.1.1
	g3/0	11.1.2.2	ASBR1-BbC	g3/0	11.1.2.1
	Loopback0	4.4.4.4			
ASBR-AS2	g2/0	10.1.1.2	P1-AS2	g2/0	10.1.1.1
	g3/0	10.1.2.1	ASBR2-BbC	g3/0	10.1.2.2
	Loopback0	5.5.5.5			
ASBR1-BbC	g3/0	11.1.2.1	ASBR-AS1	g3/0	11.1.2.2
	g1/0	1.1.1.1	P1-BbC	g1/0	1.1.1.2
	Loopback0	6.6.6.6			

ASBR2-BbC	g3/0	10.1.2.2	ASBR-AS2	g3/0	10.1.2.1
	g1/0	2.1.1.1	P1-BbC	g2/0	2.1.1.2
	Loopback0	7.7.7.7			
P1-BbC	g1/0	1.1.1.2	ASBR1-BbC	g1/0	1.1.1.1
	g2/0	2.1.1.2	ASBR2-BbC	g1/0	2.1.1.1
	Loopback0	8.8.8.8			

III.2 Dynagen's .net file

III.2.1 IPv6

```
autostart = False

 [localhost:7200]
   workingdir = /tmp
   udp = 10210
     [[7200]]
            image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
            idlepc = 0x607a48c0
            npe = npe-400
            ram = 256
            ghostios = True
            sparsemem = True
   [[ROUTER P1-AS2]]
         cnfg = /labs/ipv6/config/P1-AS2.cfg
      console = 2021
      slot1 = PA-8T
      s1/0 = PE2-AS2 s2/0
      s1/1 = PE1-AS2 s2/0
      slot2 = PA-GE
      g2/0 = P1-AS1 g2/0


[localhost:7201]
   workingdir = /tmp
   udp = 10220
     [[7200]]
            image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
            idlepc = 0x607a48c0
            npe = npe-400
            ram = 256
            ghostios = True
            sparsemem = True
   [[ROUTER P1-AS1]]
         cnfg = /labs/ipv6/config/P1-AS1.cfg
      console = 2022
      slot1 = PA-8T
      s1/0 = PE2-AS1 s2/0
      s1/1 = PE1-AS1 s2/0
      slot2 = PA-GE
      g2/0 = P1-AS2 g2/0

####################################################################################################
```

```
[localhost:7202]
  workingdir = /tmp
  udp = 10250
   [[7200]]
        image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
        idlepc = 0x607a48c0
        npe = npe-400
        ram = 256
        ghostios = True
        sparsemem = True
  [[ROUTER PE2-AS2]]
      cnfg = /labs/ipv6/config/PE2-AS2.cfg
    console = 2025
    slot2 = PA-2FE-TX
    slot1 = PA-2FE-TX
    f1/0 = CE3-AS2 f1/0
    f1/1 = CE4-AS2 f1/0
      f0/0 = CE5-AS2 f1/0
    slot3 = PA-8T
    s2/0 = P1-AS2 s1/0


[localhost:7203]
  workingdir = /tmp
  udp = 10260
   [[7200]]
        image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
        idlepc = 0x607a48c0
        npe = npe-400
        ram = 256
        ghostios = True
        sparsemem = True
  [[ROUTER PE1-AS2]]
      cnfg = /labs/ipv6/config/PE1-AS2.cfg
    console = 2026
    slot1 = PA-2FE-TX
    f1/0 = CE1-AS2 f1/0
    f1/1 = CE2-AS2 f1/0
    slot2 = PA-8T
    s2/0 = P1-AS2 s1/0


[localhost:7204]
  workingdir = /tmp
  udp = 10270
   [[7200]]
        image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
        idlepc = 0x607a48c0
        npe = npe-400
        ram = 256
        ghostios = True
        sparsemem = True
  [[ROUTER PE1-AS1]]
      cnfg = /labs/ipv6/config/PE1-AS1.cfg
    console = 2027
    slot1 = PA-2FE-TX
    slot2 = PA-2FE-TX
    f1/0 = CE3-AS1 f1/0
    f1/1 = CE2-AS1 f1/0
```

```
            f0/0 = CE1-AS1 f1/0
        slot3 = PA-8T
            s2/0 = P1-AS1 s1/1

[localhost:7205]
    workingdir = /tmp
    udp = 10280
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER PE2-AS1]]
            cnfg = /labs/ipv6/config/PE2-AS1.cfg
        console = 2028
        slot1 = PA-2FE-TX
        f1/0 = CE4-AS1 f1/0
        f1/1 = CE5-AS1 f1/1
        slot2 = PA-8T
        s2/0 = P1-AS1 s1/1

########################################################################################################
##############
[localhost:7206]
    workingdir = /tmp
    udp = 10300
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER CE1-AS2]]
            cnfg = /labs/ipv6/config/CE1-AS2.cfg
        console = 2030
        slot1 = PA-2FE-TX
        f1/0 = PE1-AS2 f1/0
            slot2 = PA-2FE-TX

[localhost:7215]
    workingdir = /tmp
    udp = 10310
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True


    [[ROUTER CE2-AS2]]
            cnfg = /labs/ipv6/config/CE2-AS2.cfg
        console = 2031
        slot1 = PA-2FE-TX
        f1/0 = PE1-AS2 f1/1
```

```
[localhost:7207]
    workingdir = /tmp
    udp = 10320
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER CE3-AS2]]
            cnfg = /labs/ipv6/config/CE3-AS2.cfg
        console = 2032
        slot1 = PA-2FE-TX
        f1/0 = PE2-AS2 f1/0

[localhost:7208]
    workingdir = /tmp
    udp = 10330
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER CE4-AS2]]
            cnfg = /labs/ipv6/config/CE4-AS2.cfg
        console = 2033
        slot1 = PA-2FE-TX
        f1/0 = PE2-AS2 f1/1


[localhost:7209]
    workingdir = /tmp
    udp = 10340
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER CE1-AS1]]
            cnfg = /labs/ipv6/config/CE1-AS1.cfg
        console = 2034
        slot1 = PA-2FE-TX
            f1/0 = PE1-AS2 f1/0

[localhost:7210]
    workingdir = /tmp
    udp = 10350
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER CE5-AS2]]
```

```
      cnfg = /labs/ipv6/config/CE5-AS2.cfg
    console = 2035
    slot1 = PA-2FE-TX
      f1/1 = PE2-AS2 f0/0

[localhost:7211]
  workingdir = /tmp
  udp = 10360
   [[7200]]
        image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
        idlepc = 0x607a48c0
        npe = npe-400
        ram = 256
        ghostios = True
        sparsemem = True
  [[ROUTER CE5-AS1]]
      cnfg = /labs/ipv6/config/CE5-AS1.cfg
    console = 2036
    slot1 = PA-2FE-TX
    f1/1 = PE2-AS1 f1/1

[localhost:7212]
  workingdir = /tmp
  udp = 10370
   [[7200]]
        image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
        idlepc = 0x607a48c0
        npe = npe-400
        ram = 256
        ghostios = True
        sparsemem = True
  [[ROUTER CE4-AS1]]
      cnfg = /labs/ipv6/config/CE4-AS1.cfg
    console = 2037
    slot1 = PA-2FE-TX
    f1/0 = PE2-AS1 f1/0

[localhost:7213]
  workingdir = /tmp
  udp = 10380
   [[7200]]
        image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
        idlepc = 0x607a48c0
        npe = npe-400
        ram = 256
        ghostios = True
        sparsemem = True
  [[ROUTER CE3-AS1]]
      cnfg = /labs/ipv6/config/CE3-AS1.cfg
    console = 2038
    slot1 = PA-2FE-TX
    f1/0 = PE1-AS1 f1/0

[localhost:7214]
  workingdir = /tmp
  udp = 10390
   [[7200]]
        image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
        idlepc = 0x607a48c0
        npe = npe-400
```

```
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER CE2-AS1]]
            cnfg = /labs/ipv6/config/CE2-AS1.cfg
        console = 2039
        slot1 = PA-2FE-TX
        f1/0 = PE1-AS1 f1/1
```

III.2.2 interAS

```
autostart = False

[localhost:7200]
    workingdir = /tmp
    udp = 10210
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER P1-AS2]]
            cnfg = /labs/ipv6/config/P1-AS2.cfg
        console = 2021
        slot1 = PA-8T
        s1/0 = PE2-AS2 s2/0
        s1/1 = PE1-AS2 s2/0
        slot2 = PA-GE
        g2/0 = ASBR-AS1 g2/0


[localhost:7201]
    workingdir = /tmp
    udp = 10220
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER P1-AS1]]
            cnfg = /labs/ipv6/config/P1-AS1.cfg
        console = 2022
        slot1 = PA-8T
        s1/0 = PE2-AS1 s2/0
        s1/1 = PE1-AS1 s2/0
        slot2 = PA-GE
        g2/0 = ASBR-AS1 g2/0

##########################################################################################


[localhost:7216]
    workingdir = /tmp
    udp = 10400
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER ASBR-AS1]]
            cnfg = /labs/ipv6/config/ASBR-AS1.cfg
        console = 2021
        slot2 = PA-GE
        g2/0 = P1-AS1 g2/0
            g3/0 = ASBR-AS2 g3/0


[localhost:7217]
    workingdir = /tmp
    udp = 10410
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER ASBR-AS2]]
            cnfg = /labs/ipv6/config/ASBR-AS2.cfg
        console = 2021
        slot2 = PA-GE
        g2/0 = P1-AS1 g2/0
            g3/0 = ASBR-AS1 g3/0


#######################################################################################################


[localhost:7202]
    workingdir = /tmp
    udp = 10250
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER PE2-AS2]]
            cnfg = /labs/ipv6/config/PE2-AS2.cfg
        console = 2025
        slot2 = PA-2FE-TX
        slot1 = PA-2FE-TX
        f1/0 = CE3-AS2 f1/0
        f1/1 = CE4-AS2 f1/0
            f0/0 = CE5-AS2 f1/0
        slot3 = PA-8T
        s2/0 = P1-AS2 s1/0


[localhost:7203]
    workingdir = /tmp
    udp = 10260
      [[7200]]
```

```
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER PE1-AS2]]
            cnfg = /labs/ipv6/config/PE1-AS2.cfg
        console = 2026
        slot1 = PA-2FE-TX
        f1/0 = CE1-AS2 f1/0
        f1/1 = CE2-AS2 f1/0
        slot2 = PA-8T
        s2/0 = P1-AS2 s1/0


[localhost:7204]
    workingdir = /tmp
    udp = 10270
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER PE1-AS1]]
            cnfg = /labs/ipv6/config/PE1-AS1.cfg
        console = 2027
        slot1 = PA-2FE-TX
        slot2 = PA-2FE-TX
        f1/0 = CE3-AS1 f1/0
        f1/1 = CE2-AS1 f1/0
            f0/0 = CE1-AS1 f1/0
        slot3 = PA-8T
            s2/0 = P1-AS1 s1/1


[localhost:7205]
    workingdir = /tmp
    udp = 10280
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER PE2-AS1]]
            cnfg = /labs/ipv6/config/PE2-AS1.cfg
        console = 2028
        slot1 = PA-2FE-TX
        f1/0 = CE4-AS1 f1/0
        f1/1 = CE5-AS1 f1/1
        slot2 = PA-8T
        s2/0 = P1-AS1 s1/0


##########################################################################################
```

```
#############
[localhost:7206]
    workingdir = /tmp
    udp = 10300
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER CE1-AS2]]
            cnfg = /labs/ipv6/config/CE1-AS2.cfg
        console = 2030
        slot1 = PA-2FE-TX
        f1/0 = PE1-AS2 f1/0
            slot2 = PA-2FE-TX


[localhost:7215]
    workingdir = /tmp
    udp = 10310
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True

    [[ROUTER CE2-AS2]]
            cnfg = /labs/ipv6/config/CE2-AS2.cfg
        console = 2031
        slot1 = PA-2FE-TX
        f1/0 = PE1-AS2 f1/1


[localhost:7207]
    workingdir = /tmp
    udp = 10320
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER CE3-AS2]]
            cnfg = /labs/ipv6/config/CE3-AS2.cfg
        console = 2032
        slot1 = PA-2FE-TX
        f1/0 = PE2-AS2 f1/0

[localhost:7208]
    workingdir = /tmp
    udp = 10330
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER CE4-AS2]]
            cnfg = /labs/ipv6/config/CE4-AS2.cfg
        console = 2033
        slot1 = PA-2FE-TX
        f1/0 = PE2-AS2 f1/1

[localhost:7209]
    workingdir = /tmp
    udp = 10340
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER CE1-AS1]]
            cnfg = /labs/ipv6/config/CE1-AS1.cfg
        console = 2034
        slot1 = PA-2FE-TX
            f1/0 = PE1-AS1 f0/0


[localhost:7210]
    workingdir = /tmp
    udp = 10350
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER CE5-AS2]]
            cnfg = /labs/ipv6/config/CE5-AS2.cfg
        console = 2035
        slot1 = PA-2FE-TX
            f1/1 = PE2-AS2 f0/0


[localhost:7211]
    workingdir = /tmp
    udp = 10360
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER CE5-AS1]]
            cnfg = /labs/ipv6/config/CE5-AS1.cfg
        console = 2036
        slot1 = PA-2FE-TX
        f1/1 = PE2-AS1 f1/1

 [localhost:7212]
```

```
   workingdir = /tmp
   udp = 10370
    [[7200]]
          image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
          idlepc = 0x607a48c0
          npe = npe-400
          ram = 256
          ghostios = True
          sparsemem = True
   [[ROUTER CE4-AS1]]
        cnfg = /labs/ipv6/config/CE4-AS1.cfg
     console = 2037
     slot1 = PA-2FE-TX
     f1/0 = PE2-AS1 f1/0



[localhost:7213]
   workingdir = /tmp
   udp = 10380
    [[7200]]
          image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
          idlepc = 0x607a48c0
          npe = npe-400
          ram = 256
          ghostios = True
          sparsemem = True
   [[ROUTER CE3-AS1]]
        cnfg = /labs/ipv6/config/CE3-AS1.cfg
     console = 2038
     slot1 = PA-2FE-TX
     f1/0 = PE1-AS1 f1/0

[localhost:7214]
   workingdir = /tmp
   udp = 10390
    [[7200]]
          image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
          idlepc = 0x607a48c0
          npe = npe-400
          ram = 256
          ghostios = True
          sparsemem = True
   [[ROUTER CE2-AS1]]
        cnfg = /labs/ipv6/config/CE2-AS1.cfg
     console = 2039
     slot1 = PA-2FE-TX
     f1/0 = PE1-AS1 f1/1
```

III.2.3 CsC

```
autostart = False

[localhost:7200]
   workingdir = /tmp
   udp = 10210
    [[7200]]
          image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
          idlepc = 0x607a48c0
```

```
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER P1-AS2]]
            cnfg = /labs/ipv6/config/P1-AS2.cfg
        console = 2021
        slot1 = PA-8T
        s1/0 = PE2-AS2 s2/0
        s1/1 = PE1-AS2 s2/0
        slot2 = PA-GE
        g2/0 = ASBR-AS1 g2/0




[localhost:7201]
    workingdir = /tmp
    udp = 10220
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER P1-AS1]]
            cnfg = /labs/ipv6/config/P1-AS1.cfg
        console = 2022
        slot1 = PA-8T
        s1/0 = PE2-AS1 s2/0
        s1/1 = PE1-AS1 s2/0
        slot2 = PA-GE
        g2/0 = ASBR-AS1 g2/0

######################################################################################

[localhost:7216]
    workingdir = /tmp
    udp = 10400
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER ASBR-AS1]]
            cnfg = /labs/ipv6/config/ASBR-AS1.cfg
        console = 2021
        slot1 = PA-8T
        slot2 = PA-GE
        g2/0 = P1-AS1 g2/0
            g3/0 = ASBR1-BbC g3/0

[localhost:7217]
    workingdir = /tmp
    udp = 10410
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER ASBR-AS2]]
            cnfg = /labs/ipv6/config/ASBR-AS2.cfg
        console = 2021
        slot1 = PA-8T
        slot2 = PA-GE
        g2/0 = P1-AS2 g2/0
            g3/0 = ASBR2-BbC g3/0


####################################################################################################


[localhost:7218]
    workingdir = /tmp
    udp = 10420
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER ASBR1-BbC]]
            cnfg = /labs/ipv6/config/ASBR1-BbC.cfg
        console = 2021
        slot1 = PA-8T
        slot2 = PA-GE
        g3/0 = ASBR-AS1 g3/0
            g1/0 = P1-BbC g1/0

[localhost:7219]
    workingdir = /tmp
    udp = 10430
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER ASBR2-BbC]]
            cnfg = /labs/ipv6/config/ASBR2-BbC.cfg
        console = 2021
        slot1 = PA-8T
        slot2 = PA-GE
        g3/0 = ASBR-AS2 g3/0
            g1/0 = P1-BbC g2/0


[localhost:7220]
    workingdir = /tmp
    udp = 10410
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER P1-BbC]]
            cnfg = /labs/ipv6/config/P1-BbC.cfg
        console = 2021
        slot1 = PA-8T
        slot2 = PA-GE
        g1/0 = ASBR1-BbC g1/0
            g2/0 = ASBR2-BbC g1/0


####################################################################################################
#######


[localhost:7202]
    workingdir = /tmp
    udp = 10250
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER PE2-AS2]]
            cnfg = /labs/ipv6/config/PE2-AS2.cfg
        console = 2025
        slot2 = PA-2FE-TX
        slot1 = PA-2FE-TX
        f1/0 = CE3-AS2 f1/0
        f1/1 = CE4-AS2 f1/0
            f2/0 = CE5-AS2 f1/0
        slot3 = PA-8T
        s3/0 = P1-AS2 s1/0

[localhost:7203]
    workingdir = /tmp
    udp = 10260
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER PE1-AS2]]
            cnfg = /labs/ipv6/config/PE1-AS2.cfg
        console = 2026
        slot1 = PA-2FE-TX
        f1/0 = CE1-AS2 f1/0
        f1/1 = CE2-AS2 f1/0
        slot2 = PA-8T
        s2/0 = P1-AS2 s1/0

[localhost:7204]
    workingdir = /tmp
    udp = 10270
```

```
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER PE1-AS1]]
            cnfg = /labs/ipv6/config/PE1-AS1.cfg
        console = 2027
        slot1 = PA-2FE-TX
        slot2 = PA-2FE-TX
        f1/0 = CE3-AS1 f1/0
        f1/1 = CE2-AS1 f1/1
            f2/0 = CE1-AS1 f1/0
        slot3 = PA-8T

[localhost:7205]
    workingdir = /tmp
    udp = 10280
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER PE2-AS1]]
            cnfg = /labs/ipv6/config/PE2-AS1.cfg
        console = 2028
        slot1 = PA-2FE-TX
        f1/0 = CE4-AS1 f1/0
        f1/1 = CE5-AS1 f1/1
        slot2 = PA-8T
        s2/0 = P1-AS1 s1/1

#######################################################################################################
##############
[localhost:7206]
    workingdir = /tmp
    udp = 10300
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER CE1-AS2]]
            cnfg = /labs/ipv6/config/CE1-AS2.cfg
        console = 2030
        slot1 = PA-2FE-TX
        f1/0 = PE1-AS2 f1/0

        slot2 = PA-2FE-TX

[localhost:7215]
    workingdir = /tmp
    udp = 10310
      [[7200]]
```

```
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True


    [[ROUTER CE2-AS2]]
            cnfg = /labs/ipv6/config/CE2-AS2.cfg
        console = 2031
        slot1 = PA-2FE-TX
        f1/0 = PE1-AS2 f1/1


[localhost:7207]
    workingdir = /tmp
    udp = 10320
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER CE3-AS2]]
            cnfg = /labs/ipv6/config/CE3-AS2.cfg
        console = 2032
        slot1 = PA-2FE-TX
        f1/0 = PE2-AS2 f1/0

[localhost:7208]
    workingdir = /tmp
    udp = 10330
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER CE4-AS2]]
            cnfg = /labs/ipv6/config/CE4-AS2.cfg
        console = 2033
        slot1 = PA-2FE-TX
        f1/0 = PE2-AS2 f1/1

[localhost:7209]
    workingdir = /tmp
    udp = 10340
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER CE1-AS1]]
            cnfg = /labs/ipv6/config/CE1-AS1.cfg
        console = 2034
        slot1 = PA-2FE-TX
            f1/0 = PE1-AS1 f0/0


[localhost:7210]
    workingdir = /tmp
    udp = 10350
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER CE5-AS2]]
            cnfg = /labs/ipv6/config/CE5-AS2.cfg
        console = 2035
        slot1 = PA-2FE-TX
            f1/1 = PE2-AS2 f0/0


[localhost:7211]
    workingdir = /tmp
    udp = 10360
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER CE5-AS1]]
            cnfg = /labs/ipv6/config/CE5-AS1.cfg
        console = 2036
        slot1 = PA-2FE-TX
        f1/1 = PE2-AS1 f1/1

[localhost:7212]
    workingdir = /tmp
    udp = 10370
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER CE4-AS1]]
            cnfg = /labs/ipv6/config/CE4-AS1.cfg
        console = 2037
        slot1 = PA-2FE-TX
        f1/0 = PE2-AS1 f1/0


[localhost:7213]
    workingdir = /tmp
    udp = 10380
      [[7200]]
                image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
        npe = npe-400
        ram = 256
        ghostios = True
        sparsemem = True
  [[ROUTER CE3-AS1]]
      cnfg = /labs/ipv6/config/CE3-AS1.cfg
    console = 2038
    slot1 = PA-2FE-TX
    f1/0 = PE1-AS1 f1/0


[localhost:7214]
  workingdir = /tmp
  udp = 10390
   [[7200]]
        image = /labs/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
        idlepc = 0x607a48c0
        npe = npe-400
        ram = 256
        ghostios = True
        sparsemem = True
  [[ROUTER CE2-AS1]]
      cnfg = /labs/ipv6/config/CE2-AS1.cfg
    console = 2039
    slot1 = PA-2FE-TX
    f1/0 = PE1-AS1 f1/1
```

III.3 Router configurations

ASBR-AS1

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname ASBR-AS1.cfg
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
no logging console
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface Loopback0
```

```
 ip address 4.4.4.4 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/1
 no ip address
 shutdown
 duplex half
!
interface Serial2/0
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/3
 no ip address
 shutdown
 serial restart-delay 0
!
interface GigabitEthernet2/0
 ip address 10.2.1.2 255.255.255.252
 description Connected to P1-AS2 g2/0
 half-duplex
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 negotiation auto
!
interface GigabitEthernet3/0
 ip address 11.1.2.2 255.255.255.252
 description Connected to ASBR1-BbC g3/0
 half-duplex
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 negotiation auto
ip forward-protocol nd
no ip http server
no ip http secure-server
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id Loopback0
 log-adjacency-changes
 network 0.0.0.0 0.255.255.255 area 0
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

ASBR-AS2

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname ASBR-AS2.cfg
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
no logging console
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 5.5.5.5 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
```

```
!
interface FastEthernet1/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/1
 no ip address
 shutdown
 duplex half
!
interface Serial2/0
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/3
 no ip address
 shutdown
 serial restart-delay 0
!
interface GigabitEthernet2/0
 ip address 10.1.1.2 255.255.255.252
 description Connected to P1-AS2 g2/0
 half-duplex
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 negotiation auto
!
interface GigabitEthernet3/0
 ip address 10.1.2.1 255.255.255.252
 description Connected to ASBR2-BbC g3/0
 half-duplex
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 negotiation auto
ip forward-protocol nd
no ip http server
no ip http secure-serv
ip forward-protocol nd
no ip http server
no ip http secure-server
!
router ospf 1
```

```
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id Loopback0
 log-adjacency-changes
 network 0.0.0.0 0.255.255.255 area 0
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

ASBR1-BbC

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname ASBR1-BbC.cfg
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
no logging console
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 6.6.6.6 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/1
 no ip address
 shutdown
 duplex half
!
interface Serial2/0
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/3
 no ip address
 shutdown
 serial restart-delay 0
!
interface GigabitEthernet1/0
 ip address 1.1.1.1 255.255.255.252
 description Connected to P1-BbC g1/0
 half-duplex
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 negotiation auto
!
interface GigabitEthernet2/0
no ip address
 shutdown
 duplex half
!
interface GigabitEthernet3/0
 ip address 11.1.2.1 255.255.255.252
 description Connected to ASBR-AS1 g3/0
 half-duplex
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 negotiation auto
ip forward-protocol nd
no ip http server
no ip http secure-serv
ip forward-protocol nd
no ip http server
no ip http secure-server
```

```
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id Loopback0
 log-adjacency-changes
 network 0.0.0.0 0.255.255.255 area 0
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

ASBR2-BbC

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname ASBR2-BbC.cfg
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
no logging console
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 7.7.7.7 255.255.255.255
!
interface FastEthernet0/0
```

```
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 ip address 33.1.1.2 255.255.255.0
 description Connected to CE3 f1/0
 duplex auto
 speed auto
!
interface FastEthernet1/1
 ip address 34.1.1.2 255.255.255.0
 description Connected to CE4 f1/0
 duplex auto
 speed auto
!
interface Serial2/0
 ip address 22.1.1.2 255.255.255.252
 description Connected to P2 s1/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 serial restart-delay 0
!
interface Serial2/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/3
 no ip address
 shutdown
 serial restart-delay 0
!
interface GigabitEthernet1/0
 ip address 1.1.1.2 255.255.255.252
 description Connected to P1-BbC g1/0
 half-duplex
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 negotiation auto
!
interface GigabitEthernet2/0
no ip address
 shutdown
 duplex half
!
interface GigabitEthernet3/0
 ip address 10.1.2.2 255.255.255.252
 description Connected to ASBR-AS2 g3/0
 half-duplex
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 negotiation auto
ip forward-protocol nd
no ip http server
no ip http secure-serv
ip forward-protocol nd
no ip http server
no ip http secure-server
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id Loopback0
 log-adjacency-changes
 network 0.0.0.0 0.255.255.255 area 0
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

CE1-AS1

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname CE1-AS1.cfg
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
```

```
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 ip address 35.1.1.1 255.255.255.0
 description Connected to PE1-AS1 f0/0
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

CE1-AS2

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname CE1-AS2.cfg
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
```

```
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 ip address 31.1.1.1 255.255.255.0
 description Connected to PE1-AS2 f1/0
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet2/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet2/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

CE2-AS1

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname CE2-AS1.cfg
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 ip address 30.1.1.1 255.255.255.0
 description Connected to PE1-AS1 f1/1
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
```

```
line vty 0 4
 login
!
end
```

CE2-AS2

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname CE2-AS2.cfg
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 ip address 32.1.1.1 255.255.255.0
 description Connected to PE1-AS2 f1/1
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
```

```
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

CE3-AS1

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname CE3-AS1.cfg
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 ip address 39.1.1.1 255.255.255.0
 description Connected to PE1-AS1 f1/0
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
```

```
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

CE3-AS2

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname CE3-AS2.cfg
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 ip address 33.1.1.1 255.255.255.0
 description Connected to PE2-AS2 f1/0
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
ip forward-protocol nd
no ip http server
no ip http secure-server
```

```
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

CE4-AS1

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname CE4-AS1.cfg
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 ip address 38.1.1.1 255.255.255.0
 description Connected to PE2-AS1 f1/0
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no ip address
 shutdown
 duplex auto
 speed auto
```

```
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

CE4-AS2

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname CE4-AS2.cfg
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 ip address 34.1.1.1 255.255.255.0
 description Connected to PE2-AS2 f1/1
 duplex auto
 speed auto
!
interface FastEthernet1/1
```

```
 no ip address
 shutdown
 duplex auto
 speed auto
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

CE5-AS1

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname CE5-AS1.cfg
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet1/1
 ip address 37.1.1.1 255.255.255.0
 description Connected to PE2-AS1 f1/1
 duplex auto
 speed auto
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

CE5-AS2

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname CE5-AS2.cfg
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
```

```
!
interface FastEthernet1/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet1/1
 ip address 36.1.1.1 255.255.255.0
 description Connected to PE2-AS2 f0/0
 duplex auto
 speed auto
!
interface FastEthernet2/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet2/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

P1-AS1

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname P1-AS1.cfg
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
no logging console
```

```
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
mpls label protocol ldp
mpls traffic-eng tunnels
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 3.3.3.3 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface Serial1/0
 ip address 23.1.1.1 255.255.255.252
 description Connected to PE2-AS1 s2/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 serial restart-delay 0
!
interface Serial1/1
 ip address 23.2.2.1 255.255.255.252
 description Connected to PE1-AS1 s2/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 serial restart-delay 0
!
interface Serial1/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
!
interface GigabitEthernet2/0
 ip address 10.1.1.2 255.255.255.252
 description Connected to ASBR-AS1 g2/0
 half-duplex
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 negotiation auto
!
interface GigabitEthernet3/0
 no ip address
 shutdown
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 3.3.3.3
 log-adjacency-changes
 network 10.0.0.0 0.255.255.255 area 0
!
logging alarm informational
!
mpls ldp router-id Loopback0 force
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

P1-AS2

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname P1-AS2.cfg
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
no logging console
!
no aaa new-model
ip source-route
ip cef
!
mpls label protocol ldp
mpls traffic-eng tunnels
```

```
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 2.2.2.2 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface Serial1/0
 ip address 22.1.1.1 255.255.255.252
 description Connected to PE2-AS2 s2/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 serial restart-delay 0
!
interface Serial1/1
 ip address 22.1.1.1 255.255.255.252
 description Connected to PE1-AS2 s2/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 serial restart-delay 0
!
interface Serial1/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
!
interface GigabitEthernet2/0
 ip address 10.1.1.4 255.255.255.252
 description Connected to ASBR-AS2 g2/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 negotiation auto
!
interface GigabitEthernet3/0
 no ip address
 shutdown
```

```
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id Loopback0
 log-adjacency-changes
 network 0.0.0.0 0.255.255.255 area 0
!
logging alarm informational
!
mpls ldp router-id Loopback0 force
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

P1-BbC

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname P1-BbC.cfg
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
no logging console
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 8.8.8.8 255.255.255.255
```

```
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/1
 no ip address
 shutdown
 duplex half
!
interface Serial2/0
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/3
 no ip address
 shutdown
 serial restart-delay 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
interface GigabitEthernet1/0
 ip address 10.1.1.2 255.255.255.252
 description Connected to ASBR1-BbC g1/0
 half-duplex
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 negotiation auto
!
interface GigabitEthernet2/0
 ip address 10.1.1.2 255.255.255.252
 description Connected to ASBR2-BbC g1/0
 half-duplex
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 negotiation auto
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id Loopback0
 log-adjacency-changes
 network 0.0.0.0 0.255.255.255 area 0
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

PE1-AS1

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE1-AS1.cfg
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
no logging console
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 33.33.33.33 255.255.255.255
!
interface FastEthernet0/0
 ip address 35.1.1.2 255.255.255.0
 description Connected to CE1-AS1 f1/0
 duplex half
```

```
!
interface FastEthernet1/0
 ip address 39.1.1.2 255.255.255.0
 description Connected to CE4-AS1 f1/0
 duplex auto
 speed auto
!
interface FastEthernet1/1
 ip address 30.1.1.2 255.255.255.0
 description Connected to CE2-AS1 f1/0
 duplex auto
 speed auto
!
interface Serial2/0
 ip address 23.2.2.2 255.255.255.252
 description Connected to P1-AS1 s1/1
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 serial restart-delay 0
!
interface Serial2/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/3
 no ip address
 shutdown
 serial restart-delay 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id Loopback0
 log-adjacency-changes
 network 0.0.0.0 0.255.255.255 area 0
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
```

```
line vty 0 4
 login
!
end
```

PE1-AS2

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE1-AS2.cfg
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
no logging console
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 22.22.22.22 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 ip address 31.1.1.2 255.255.255.0
 description Connected to CE1-AS2 f1/0
 duplex auto
 speed auto
!
interface FastEthernet1/1
 ip address 32.1.1.2 255.255.255.0
 description Connected to CE2-AS2 f1/0
 duplex auto
 speed auto
!
interface Serial2/0
 ip address 21.1.1.2 255.255.255.252
 description Connected to P1 s1/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 serial restart-delay 00
!
interface Serial2/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/3
 no ip address
 shutdown
 serial restart-delay 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id Loopback0
 log-adjacency-changes
 network 0.0.0.0 0.255.255.255 area 0
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

PE2-AS1

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE2-AS1.cfg
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
```

```
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 44.44.44.44 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 ip address 38.1.1.2 255.255.255.0
 description Connected to CE4-AS1 f1/0
 duplex auto
 speed auto
!
interface FastEthernet1/1
 ip address 37.1.1.2 255.255.255.0
 description Connected to CE5-AS1 f1/1
 duplex auto
 speed auto
!
interface Serial2/0
 ip address 23.1.1.2 255.255.255.252
 description Connected to P1-AS1 s1/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 serial restart-delay 00
!
interface Serial2/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/3
 no ip address
 shutdown
 serial restart-delay 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
```

```
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id Loopback0
 log-adjacency-changes
 network 0.0.0.0 0.255.255.255 area 0
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

PE2-AS2

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE2-AS2.cfg
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
no logging console
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 11.11.11.11 255.255.255.255
!
interface FastEthernet0/0
 ip address 36.1.1.2 255.255.255.0
 description Connected to CE5-AS2 f1/0
 duplex half
!
```

```
interface FastEthernet1/0
 ip address 33.1.1.2 255.255.255.0
 description Connected to CE3-AS2 f1/0
 duplex auto
 speed auto
!
interface FastEthernet1/1
 ip address 34.1.1.2 255.255.255.0
 description Connected to CE4-AS2 f1/0
 duplex auto
 speed auto
!
interface Serial2/0
 ip address 22.1.1.2 255.255.255.252
 description Connected to P1-AS2 s1/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 serial restart-delay 0
!
interface Serial2/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/3
 no ip address
 shutdown
 serial restart-delay 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id Loopback0
 log-adjacency-changes
 network 0.0.0.0 0.255.255.255 area 0
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
```

```
 login
!
end
```

IV. Scenario 4: MPLS – QoS

IV.1 Router interfaces IP list

Router	Interface	IP	connected to	Interface	IP
CE1	f1/0	31.1.1.1	PE2	F1/0	31.1.1.2
	f1/1		CustE site1	f0/0	
	Loopback0	192.168.1.193			
CE2	f1/0	32.1.1.1	PE2	f1/1	32.1.1.2
	f1/1		CustC site2	f0/0	
	Loopback0	192.168.2.129			
CE3	f1/0	33.1.1.1	PE1	f1/0	33.1.1.2
	f1/1		CustD site1	f0/0	
	Loopback0	192.168.1.1			
CE4	f1/0	34.1.1.1	PE1	f1/1	34.1.1.2
	f1/1		CustC site1	f0/0	
	Loopback0	192.168.2.1			
CE5	f1/0	35.1.1.1	PE5	f1/0	35.1.1.2
	f1/1		CustE site2	f0/0	
	Loopback0	192.168.5.1			
CE6	f1/0		CustB site2	f0/0	
	f1/1	36.1.1.1	PE5	f1/1	36.1.1.2
	Loopback0	192.168.6.1			
CE7	f1/0		CustB site1	f0/0	
	f1/1	37.1.1.1	PE4	f1/1	37.1.1.2
	Loopback0	192.168.7.1			
CE8	f1/0	38.1.1.1	PE4	f1/0	38.1.1.2
	f1/1		CustA site1	F0/0	
	Loopback0	192.168.8.1			
CE9	f1/0	39.1.1.1	PE3	f1/0	39.1.1.2
	f1/1		CustD site2	f0/0	
	Loopback0	192.168.2.193			
CE10	f1/0	30.1.1.1	PE3	f1/1	30.1.1.2
	f1/1		CustA site 2	f0/0	
	Loopback0	192.168.1.129			
PE1	f1/0	33.1.1.2	CE3	f1/0	33.1.1.1
	f1/1	34.1.1.2	CE4	f1/0	34.1.1.1
	s2/0	22.1.1.2	P2	s1/0	22.1.1.1
	Loopback0	11.11.11.11			

Router	Interface	IP	connected to	Interface	IP
PE2	f1/0	31.1.1.2	CE1	f1/0	31.1.1.1
	f1/1	32.1.1.2	CE2	f1/0	32.1.1.1
	s2/0	21.1.1.2	P1	s1/0	21.1.1.1
	Loopback0	22.22.22.22			
PE3	f1/0	39.1.1.2	CE9	F1/0	39.1.1.1
	f1/1	30.1.1.2	CE10	F1/0	30.1.1.1
	s2/0	23.2.2.2	P3	s1/1	23.2.2.1
	Loopback0	33.33.33.33			
PE4	f1/0	38.1.1.2	CE8	f1/0	38.1.1.1
	f1/1	37.1.1.2	CE7	f1/1	37.1.1.1
	s2/0	23.1.1.2	P3	s1/0	23.1.1.1
	Loopback0	44.44.44.44			
PE5	f1/0	35.1.1.2	CE5	f1/0	35.1.1.1
	f1/1	36.1.1.2	CE6	f1/1	36.1.1.1
	s2/0	24.1.1.2	P4	s1/0	24.1.1.1
	Loopback0	55.55.55.55			
P1	s1/0	21.1.1.1	PE2	s2/0	21.1.1.2
	g2/0	10.1.1.1	P3	g2/0	10.1.1.2
	g3/0	10.1.1.9	P2	g2/0	10.1.1.10
	Loopback0	1.1.1.1			
P2	s1/0	22.1.1.1	PE1	s2/0	22.1.1.2
	g2/0	10.1.1.10	P1	g3/0	10.1.1.9
	g3/0	10.1.1.5	P4	g2/0	10.1.1.6
	Loopback0	2.2.2.2			
P3	s1/0	23.1.1.1	PE4	s2/0	23.1.1.2
	s1/1	23.2.2.1	PE3	s2/0	23.2.2.2
	g2/0	10.1.1.2	P1	g2/0	10.1.1.1
	g3/0	10.1.1.13	P4	g3/0	10.1.1.14
	Loopback0	3.3.3.3			
P4	s1/0	24.1.1.1	PE5	s2/0	24.1.1.2
	g2/0	10.1.1.6	P2	g3/0	10.1.1.5
	g3/0	10.1.1.14	P3	g3/0	10.1.1.13
	Loopback0	4.4.4.4			

IV.2 Dynagen’s .net file

```
autostart = False

[localhost:7200]
    workingdir = /tmp
    udp = 10200
      [[7200]]
              image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
              idlepc = 0x607a48c0
              npe = npe-400
              ram = 256
              ghostios = True
              sparsemem = True
    [[ROUTER P1]]
            cnfg = /labs/vpn/config/P1.cfg
        console = 2020
        slot1 = PA-8T
        s1/0 = PE2 s2/0
        slot2 = PA-GE
        g2/0 = P3 g2/0
        slot3 = PA-GE
        g3/0 = P2 g2/0

[localhost:7201]
    workingdir = /tmp
    udp = 10210
      [[7200]]
                image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER P2]]
            cnfg = /labs/vpn/config/P2.cfg
        console = 2021
        slot1 = PA-8T
        s1/0 = PE1 s2/0
        slot2 = PA-GE
        g2/0 = P1 g3/0
        slot3 = PA-GE
        g3/0 = P4 g2/0

[localhost:7202]
    workingdir = /tmp
    udp = 10220
      [[7200]]
                image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER P3]]
            cnfg = /labs/vpn/config/P3.cfg
        console = 2022
        slot1 = PA-8T
        s1/0 = PE4 s2/0
        s1/1 = PE3 s2/0
        slot2 = PA-GE
        g2/0 = P1 g2/0
        slot3 = PA-GE
        g3/0 = P4 g3/0

[localhost:7203]
    workingdir = /tmp
    udp = 10230
      [[7200]]
                image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER P4]]
            cnfg = /labs/vpn/config/P4.cfg
        console = 2023
        slot1 = PA-8T
        s1/0 = PE5 s2/0
        slot2 = PA-GE
        g2/0 = P2 g3/0
        slot3 = PA-GE
        g3/0 = P3 g3/0

####################################################################################################

[localhost:7204]
    workingdir = /tmp
    udp = 10250
      [[7200]]
                image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER PE1]]
            cnfg = /labs/vpn/config/PE1.cfg
        console = 2025
        slot1 = PA-2FE-TX
        f1/0 = CE3 f1/0
        f1/1 = CE4 f1/0
        slot2 = PA-8T
        s2/0 = P2 s1/0

[localhost:7205]
    workingdir = /tmp
    udp = 10260
      [[7200]]
                image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER PE2]]
            cnfg = /labs/vpn/config/PE2.cfg
        console = 2026
        slot1 = PA-2FE-TX
        f1/0 = CE1 f1/0
        f1/1 = CE2 f1/0
        slot2 = PA-8T
        s2/0 = P1 s1/0

[localhost:7206]
    workingdir = /tmp
    udp = 10270
      [[7200]]
                image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER PE3]]
            cnfg = /labs/vpn/config/PE3.cfg
        console = 2027
        slot1 = PA-2FE-TX
        f1/0 = CE9 f1/0
        f1/1 = CE10 f1/0
        slot2 = PA-8T
        s2/0 = P3 s1/1

[localhost:7207]
    workingdir = /tmp
    udp = 10280
      [[7200]]
                image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER PE4]]
            cnfg = /labs/vpn/config/PE4.cfg
        console = 2028
        slot1 = PA-2FE-TX
        f1/0 = CE8 f1/0
        f1/1 = CE7 f1/1
        slot2 = PA-8T
        s2/0 = P3 s1/0

[localhost:7208]
    workingdir = /tmp
    udp = 10290
      [[7200]]
                image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER PE5]]
            cnfg = /labs/vpn/config/PE5.cfg
        console = 2029
        slot1 = PA-2FE-TX
        f1/0 = CE5 f1/0
        f1/1 = CE6 f1/1
        slot2 = PA-8T
        s2/0 = P4 s1/0

################################################################################################
#############
[localhost:7209]
    workingdir = /tmp
    udp = 10300
      [[7200]]
                image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER CE1]]
            cnfg = /labs/vpn/config/CE1.cfg
        console = 2030
        slot1 = PA-2FE-TX
        slot2 = PA-2FE-TX

[localhost:7210]
    workingdir = /tmp
    udp = 10310
      [[7200]]
                image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True

    [[ROUTER CE2]]
            cnfg = /labs/vpn/config/CE2.cfg
        console = 2031
        slot1 = PA-2FE-TX
        f1/0 = PE2 f1/1

[localhost:7211]
    workingdir = /tmp
    udp = 10320
      [[7200]]
                image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER CE3]]
            cnfg = /labs/vpn/config/CE3.cfg
        console = 2032
        slot1 = PA-2FE-TX
        f1/0 = PE1 f1/0

[localhost:7212]
    workingdir = /tmp
    udp = 10330
      [[7200]]
                image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
                idlepc = 0x607a48c0
                npe = npe-400
                ram = 256
                ghostios = True
                sparsemem = True
    [[ROUTER CE4]]
            cnfg = /labs/vpn/config/CE4.cfg
        console = 2033
        slot1 = PA-2FE-TX
        f1/0 = PE1 f1/1

[localhost:7213]
    workingdir = /tmp
    udp = 10340
      [[7200]]
                image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
            idlepc = 0x607a48c0
            npe = npe-400
            ram = 256
            ghostios = True
            sparsemem = True
    [[ROUTER CE5]]
          cnfg = /labs/vpn/config/CE5.cfg
        console = 2034
        slot1 = PA-2FE-TX
        f1/0 = PE5 f1/0

[localhost:7214]
    workingdir = /tmp
    udp = 10350
      [[7200]]
            image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
            idlepc = 0x607a48c0
            npe = npe-400
            ram = 256
            ghostios = True
            sparsemem = True
    [[ROUTER CE6]]
          cnfg = /labs/vpn/config/CE6.cfg
        console = 2035
        slot1 = PA-2FE-TX
        f1/1 = PE5 f1/1
        slot2 = PA-2FE-TX

[localhost:7215]
    workingdir = /tmp
    udp = 10360
      [[7200]]
            image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
            idlepc = 0x607a48c0
            npe = npe-400
            ram = 256
            ghostios = True
            sparsemem = True
    [[ROUTER CE7]]
          cnfg = /labs/vpn/config/CE7.cfg
        console = 2036
        slot1 = PA-2FE-TX
        f1/1 = PE4 f1/1

[localhost:7216]
    workingdir = /tmp
    udp = 10370
      [[7200]]
            image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
            idlepc = 0x607a48c0
            npe = npe-400
            ram = 256
            ghostios = True
            sparsemem = True
    [[ROUTER CE8]]
          cnfg = /labs/vpn/config/CE8.cfg
        console = 2037
        slot1 = PA-2FE-TX
        f1/0 = PE4 f1/0
```

```
[localhost:7217]
    workingdir = /tmp
    udp = 10380
      [[7200]]
            image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
            idlepc = 0x607a48c0
            npe = npe-400
            ram = 256
            ghostios = True
            sparsemem = True
    [[ROUTER CE9]]
        cnfg = /labs/vpn/config/CE9.cfg
      console = 2038
      slot1 = PA-2FE-TX
      f1/0 = PE3 f1/0

[localhost:7218]
    workingdir = /tmp
    udp = 10390
      [[7200]]
            image = /home/dynagen/ios/c7200-adventerprisek9-mz.124-20.T.bin.unpacked
            idlepc = 0x607a48c0
            npe = npe-400
            ram = 256
            ghostios = True
            sparsemem = True
    [[ROUTER CE10]]
        cnfg = /labs/vpn/config/CE10.cfg
      console = 2039
      slot1 = PA-2FE-TX
      f1/0 = PE3 f1/1
```

IV.4 Router configurations

CE1

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
```

```
  hidekeys
!
interface Loopback0
 ip address 192.168.1.193 255.255.255.64
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no shutdown
 ip address 31.1.1.1 255.255.255.0
 description Connected to PE2 f1/0
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet2/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet2/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
!
router bgp 100
 no synchronization
 bgp log-neighbor-changes
 network 0.0.0.0
 network 192.168.1.128 mask 255.255.255.64
 neighbor 31.1.1.2 remote-as 111
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
```

```
!
end
```

CE2

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 192.168.2.129 255.255.255.64
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no shutdown
 ip address 32.1.1.1 255.255.255.0
 description Connected to PE2 f1/1
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
```

```
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

CE3

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 192.168.1.1 255.255.255.128
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no shutdown
 ip address 33.1.1.1 255.255.255.0
 description Connected to PE1 f1/0
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
router bgp 100
```

```
 no synchronization
 bgp log-neighbor-changes
 network 0.0.0.0
 network 192.168.1.0 mask 255.255.255.128
 neighbor 33.1.1.2 remote-as 111
 maximum-paths 2
 no auto-summary
!
ip http server
no ip http secure-server
ip route 0.0.0.0 0.0.0.0 Null0
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

CE4

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
!
interface Loopback0
 ip address 192.168.2.1 255.255.255.128
!
```

```
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no shutdown
 ip address 34.1.1.1 255.255.255.0
 description Connected to PE1 f1/1
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
!
router eigrp 222
 network 34.1.1.0 0.0.0.255
 network 192.168.2.0 0.0.0.127
 network 0.0.0.0
 distribute-list prefix LocalOnly out
 no auto-summary
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

CE5

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
```

```
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no shutdown
 ip address 35.1.1.1 255.255.255.0
 description Connected to PE5 f1/0
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

CE6

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
```

```
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no shutdown
 ip address 36.1.1.1 255.255.255.0
 description Connected to PE5 f1/1
 duplex auto
 speed auto
!
interface FastEthernet2/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet2/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
```

```
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

CE7

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no shutdown
 ip address 37.1.1.1 255.255.255.0
 description Connected to PE4 f1/1
 duplex auto
 speed auto
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
```

```
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

CE8

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no shutdown
 ip address 38.1.1.1 255.255.255.0
 description Connected to PE4 f1/0
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no shutdown
 no ip address
 shutdown
 duplex auto
 speed auto
!
ip forward-protocol nd
```

```
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

CE9

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 192.168.2.193 255.255.255.64
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no shutdown
 ip address 39.1.1.1 255.255.255.0
 description Connected to PE3 f1/0
 duplex auto
 speed auto
```

```
!
interface FastEthernet1/1
 no ip address
 shutdown
 duplex auto
 speed auto
!

ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

CE10

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 192.168.1.129 255.255.255.64
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no shutdown
 ip address 30.1.1.1 255.255.255.0
 description Connected to PE3 f1/1
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
!
router bgp 100
 no synchronization
 bgp log-neighbor-changes
 network 0.0.0.0
 network 192.168.1.128 mask 255.255.255.64
 neighbor 30.1.1.2 remote-as 111
!
 maximum-paths 2
 no auto-summary
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

P1

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname P1
!
boot-start-marker
boot-end-marker
!
```

```
logging message-counter syslog
no logging console
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
mpls label protocol ldp
mpls traffic-eng tunnels
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 1.1.1.1 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface Serial1/0
 no shutdown
 ip address 21.1.1.1 255.255.255.252
 description Connected to PE2 s2/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 serial restart-delay 0
!
interface Serial1/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
!
interface GigabitEthernet2/0
 no shutdown
 ip address 10.1.1.1 255.255.255.252
 description Connected to P3 g2/0
 half-duplex
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 negotiation auto
!
interface GigabitEthernet3/0
 no shutdown
 ip address 10.1.1.9 255.255.255.252
 description Connected to P2 g2/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 negotiation auto
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 1.1.1.1
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0
!
!
router bgp 111
 no synchronization
 bgp log-neighbor-changes
 neighbor 11.11.11.11 remote-as 111
 neighbor 11.11.11.11 update-source Loopback0
 neighbor 11.11.11.11 route-reflector-client
 neighbor 22.22.22.22 remote-as 111
 neighbor 22.22.22.22 update-source Loopback0
 neighbor 22.22.22.22 route-reflector-client
 neighbor 33.33.33.33 remote-as 111
 neighbor 33.33.33.33 update-source Loopback0
 neighbor 33.33.33.33 route-reflector-client
 neighbor 44.44.44.44 remote-as 111
 neighbor 44.44.44.44 update-source Loopback0
 neighbor 44.44.44.44 route-reflector-client
 neighbor 55.55.55.55 remote-as 111
 neighbor 55.55.55.55 update-source Loopback0
 neighbor 55.55.55.55 route-reflector-client
 no auto-summary
 !
 address-family vpnv4
  neighbor 11.11.11.11 activate
  neighbor 11.11.11.11 send-community extended
  neighbor 11.11.11.11 route-reflector-client
  neighbor 22.22.22.22 activate
  neighbor 22.22.22.22 send-community extended
  neighbor 22.22.22.22 route-reflector-client
  neighbor 133.33.33.33 activate
  neighbor 33.33.33.33 send-community extended
  neighbor 33.33.33.33 route-reflector-client
  neighbor 44.44.44.44 activate
  neighbor 44.44.44.44 send-community extended
  neighbor 44.44.44.44 route-reflector-client
  neighbor 55.55.55.55 activate
  neighbor 55.55.55.55 send-community extended
  neighbor 55.55.55.55 route-reflector-client
 exit-address-family
!
logging alarm informational
!
mpls ldp router-id Loopback0 force
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

P2

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname P2
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
no logging console
!
no aaa new-model
ip source-route
ip cef
!
mpls label protocol ldp
mpls traffic-eng tunnels
!
no ipv6 cef
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 2.2.2.2 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface Serial1/0
 no shutdown
 ip address 22.1.1.1 255.255.255.252
 description Connected to PE1 s2/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 serial restart-delay 0
!
interface Serial1/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
!
interface GigabitEthernet2/0
 no shutdown
 ip address 10.1.1.10 255.255.255.252
 description Connected to P1 g3/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 negotiation auto
!
interface GigabitEthernet3/0
 no shutdown
 ip address 10.1.1.5 255.255.255.252
 description Connected to P4 g2/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 negotiation auto
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 2.2.2.2
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0
```

```
!
logging alarm informational
!
mpls ldp router-id Loopback0 force
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

P3

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname P3
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
no logging console
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
multilink bundle-name authenticated
!
mpls label protocol ldp
mpls traffic-eng tunnels
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 3.3.3.3 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface Serial1/0
 no shutdown
 ip address 23.1.1.1 255.255.255.252
 description Connected to PE4 s2/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 serial restart-delay 0
!
interface Serial1/1
 no shutdown
 ip address 23.2.2.1 255.255.255.252
 description Connected to PE3 s2/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 serial restart-delay 0
!
interface Serial1/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
!
interface GigabitEthernet2/0
 no shutdown
 ip address 10.1.1.2 255.255.255.252
 description Connected to P1 g2/0
 half-duplex
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 negotiation auto
!
interface GigabitEthernet3/0
 no shutdown
 ip address 10.1.1.13 255.255.255.252
 description Connected to P4 g3/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 negotiation auto
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 3.3.3.3
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0
!
logging alarm informational
!
mpls ldp router-id Loopback0 force
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end

P4
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname P4
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
no logging console
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
!
mpls label protocol ldp
mpls traffic-eng tunnels
!
multilink bundle-name authenticated
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption

!
mpls label protocol ldp
mpls traffic-eng tunnels
!
```

```
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 4.4.4.4 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface Serial1/0
 no shutdown
 ip address 24.1.1.1 255.255.255.252
 description Connected to PE5 s2/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 serial restart-delay 0
!
interface Serial1/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
!
interface GigabitEthernet2/0
 no shutdown
 ip address 10.1.1.6 255.255.255.252
 description Connected to P2 g3/0
 half-duplex
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 negotiation auto
!
interface GigabitEthernet3/0
 no shutdown
 ip address 10.1.1.14 255.255.255.252
 description Connected to P3 g3/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 negotiation auto
```

```
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 4.4.4.4
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0
!
router bgp 100
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.0.2.1 remote-as 100
 neighbor 10.0.2.1 update-source Loopback0
 neighbor 10.0.2.1 route-reflector-client
 neighbor 10.0.2.2 remote-as 100
 neighbor 10.0.2.2 update-source Loopback0
 neighbor 10.0.2.2 route-reflector-client
 neighbor 10.0.2.3 remote-as 100
 neighbor 10.0.2.3 update-source Loopback0
 neighbor 10.0.2.3 route-reflector-client
 neighbor 10.0.2.4 remote-as 100
 neighbor 10.0.2.4 update-source Loopback0
 neighbor 10.0.2.4 route-reflector-client
 no auto-summary
 !
 address-family vpnv4
  neighbor 10.0.2.1 activate
  neighbor 10.0.2.1 send-community extended
  neighbor 10.0.2.1 route-reflector-client
  neighbor 10.0.2.2 activate
  neighbor 10.0.2.2 send-community extended
  neighbor 10.0.2.2 route-reflector-client
  neighbor 10.0.2.3 activate
  neighbor 10.0.2.3 send-community extended
  neighbor 10.0.2.3 route-reflector-client
  neighbor 10.0.2.4 activate
  neighbor 10.0.2.4 send-community extended
  neighbor 10.0.2.4 route-reflector-client
 exit-address-family
!
ip http server
no ip http secure-server
!
mpls ldp router-id Loopback0 force
!
control-plane
!
line con 0
 exec-timeout 35791 0
line aux 0
line vty 0 4
 login
!
end
```

PE1

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE3
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
no logging console
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
ip vrf Net1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
!
!
ip vrf Net2
 rd 200:10
 route-target export 200:0
 route-target import 200:1
 route-target import 200:2
!
mpls label protocol ldp
mpls traffic-eng tunnels
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 11.11.11.11 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no shutdown
 ip address 33.1.1.2 255.255.255.0
 description Connected to CE3 f1/0
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no shutdown
 ip address 34.1.1.2 255.255.255.0
 description Connected to CE4 f1/0
 duplex auto
 speed auto
!
interface Serial2/0
 no shutdown
 ip address 22.1.1.2 255.255.255.252
 description Connected to P2 s1/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 serial restart-delay 0
!
interface Serial2/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/3
 no ip address
 shutdown
 serial restart-delay 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 11.11.11.11
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0
!
router bgp 111
 no synchronization
 bgp log-neighbor-changes
 neighbor 1.1.1.1 remote-as 111
 neighbor 1.1.1.1 update-source Loopback0
 no auto-summary
!
!
 address-family vpnv4
  neighbor 1.1.1.1 activate
  neighbor 1.1.1.1 send-community both
 exit-address-family
 !
 !
 address-family ipv4 vrf Net1
  redistribute connected
  neighbor 1.1.1.1 remote-as 111
  neighbor 1.1.1.1 activate
  maximum-paths eibgp 2 import 2
  no synchronization
 exit-address-family
!
 !
 address-family ipv4 vrf Net2
  redistribute connected
  neighbor 1.1.1.1 remote-as 111
  neighbor 1.1.1.1 activate
  maximum-paths eibgp 2 import 2
  no synchronization
 exit-address-family
 !
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

PE2

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Pe2
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
no logging console
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
!
ip vrf Net1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
!
!
ip vrf Net2
 rd 200:1
 route-target export 200:1
 route-target import 200:0
!
!
mpls label protocol ldp
mpls traffic-eng tunnels
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 22.22.22.22 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no shutdown
 ip address 31.1.1.2 255.255.255.0
 description Connected to CE1 f1/0
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no shutdown
 ip address 32.1.1.2 255.255.255.0
 description Connected to CE2 f1/0
 duplex auto
 speed auto
!
interface Serial2/0
 no shutdown
 ip address 21.1.1.2 255.255.255.252
 description Connected to P1 s1/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 serial restart-delay 00
!
interface Serial2/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/3
 no ip address
 shutdown
 serial restart-delay 0
!
```

```
ip forward-protocol nd
no ip http server
no ip http secure-server
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 22.22.22.22
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0
!
!
router bgp 111
 no synchronization
 bgp log-neighbor-changes
 neighbor 1.1.1.1 remote-as 111
 neighbor 1.1.1.1 update-source Loopback0
 no auto-summary
!
!
 address-family vpnv4
  neighbor 1.1.1.1 activate
  neighbor 1.1.1.1 send-community both
 exit-address-family
 !
 !
 address-family ipv4 vrf Net1
  redistribute connected
  neighbor 1.1.1.1 remote-as 111
  neighbor 1.1.1.1 activate
  maximum-paths eibgp 2 import 2
  no synchronization
 exit-address-family
!
 !
 address-family ipv4 vrf Net2
  redistribute connected
  neighbor 1.1.1.1 remote-as 111
  neighbor 1.1.1.1 activate
  maximum-paths eibgp 2 import 2
  no synchronization
 exit-address-family
 !
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

PE3

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE1
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
no logging console
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
!
ip vrf Net1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
!
!
ip vrf Net2
 rd 200:2
 route-target export 200:2
 route-target import 200:0
!
!
mpls label protocol ldp
mpls traffic-eng tunnels
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 33.33.33.33 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no shutdown
 ip address 39.1.1.2 255.255.255.0
 description Connected to CE9 f1/0
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no shutdown
 ip address 30.1.1.2 255.255.255.0
 description Connected to CE10 f1/0
 duplex auto
 speed auto
!
interface Serial2/0
 no shutdown
 ip address 23.2.2.2 255.255.255.252
 description Connected to P3 s1/1
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 serial restart-delay 0
!
interface Serial2/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/3
 no ip address
 shutdown
 serial restart-delay 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 33.33.33.33
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0
!
!
router bgp 111
 no synchronization
 bgp log-neighbor-changes
 neighbor 1.1.1.1 remote-as 111
 neighbor 1.1.1.1 update-source Loopback0
 no auto-summary
!
!
 address-family vpnv4
  neighbor 1.1.1.1 activate
  neighbor 1.1.1.1 send-community both
 exit-address-family
 !
 !
 address-family ipv4 vrf Net1
  redistribute connected
  neighbor 1.1.1.1 remote-as 111
  neighbor 1.1.1.1 activate
  maximum-paths eibgp 2 import 2
  no synchronization
 exit-address-family
!
 !
 address-family ipv4 vrf Net2
  redistribute connected
  neighbor 1.1.1.1 remote-as 111
  neighbor 1.1.1.1 activate
  maximum-paths eibgp 2 import 2
  no synchronization
 exit-address-family
 !
logging alarm informational
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

PE4

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE4
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
no ipv6 cef
!
ip vrf Net4
 rd 100:140
 route-target export 100:400
 route-target import 100:400
!
ip vrf Net6
 rd 100:160
 route-target export 100:600
 route-target import 100:600
```

```
!
!
mpls label protocol ldp
mpls traffic-eng tunnels
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 44.44.44.44 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no shutdown
 ip address 38.1.1.2 255.255.255.0
 description Connected to CE8 f1/0
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no shutdown
 ip address 37.1.1.2 255.255.255.0
 description Connected to CE7 f1/1
 duplex auto
 speed auto
!
interface Serial2/0
 no shutdown
 ip address 23.1.1.2 255.255.255.252
 description Connected to P3 s1/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 serial restart-delay 00
!
interface Serial2/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/3
 no ip address
 shutdown
 serial restart-delay 0
!
ip forward-protocol nd
```

```
no ip http server
no ip http secure-server
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 44.44.44.44
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0
!
!
router bgp 45
 bgp log-neighbor-changes
 neighbor 55.55.55.55 remote-as 45
 neighbor 55.55.55.55 update-source Loopback0
 no auto-summary
!
logging alarm informational
!
control-plane
!
!
 address-family vpnv4
 neighbor 55.55.55.55 activate
 neighbor 55.55.55.55 send-community both
 exit-address-family
!
!
address-family ipv4 vrf Net4
 redistribute connected
 no auto-summary
 no synchronization
 exit-address-family
 !
address-family ipv4 vrf Net6
 redistribute connected
 no auto-summary
 no synchronization
 exit-address-family
 !
ip classless
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

PE5

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
```

```
no service password-encryption
!
hostname PE5
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
ip vrf Net4
 rd 100:140
 route-target export 100:400
 route-target import 100:400
!
ip vrf Net5
 rd 100:150
 route-target export 100:500
 route-target import 100:500
!
ip vrf Net6
 rd 100:160
 route-target export 100:600
 route-target import 100:600
!
no ipv6 cef
!
!
mpls label protocol ldp
mpls traffic-eng tunnels
!
multilink bundle-name authenticated
!
archive
 log config
  hidekeys
!
interface Loopback0
 ip address 55.55.55.55 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface FastEthernet1/0
 no shutdown
 ip address 35.1.1.2 255.255.255.0
 description Connected to CE5 f1/0
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no shutdown
 ip address 36.1.1.2 255.255.255.0
 description Connected to CE6 f1/1
 duplex auto
 speed auto
!
interface Serial2/0
 no shutdown
 ip address 24.1.1.2 255.255.255.252
 description Connected to P4 s1/0
 mpls ip
 mpls mtu 1600
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000
 ip rsvp resource-provider none
 serial restart-delay 0
!
interface Serial2/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/3
 no ip address
 shutdown
 serial restart-delay 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 55.55.55.55
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0
!
!
router bgp 45
 bgp log-neighbor-changes
 neighbor 44.44.44.44 remote-as 45
 neighbor 44.44.44.44 update-source Loopback0
 no auto-summary
!
logging alarm informational
!
control-plane
!
 address-family vpnv4
 neighbor 44.44.44.44 activate
 neighbor 44.44.44.44 send-community both
 exit-address-family
!
address-family ipv4 vrf Net4
 redistribute connected
 no auto-summary
 no synchronization
 exit-address-family
 !
address-family ipv4 vrf Net5
 redistribute connected
 no auto-summary
 no synchronization
 exit-address-family
 !
address-family ipv4 vrf Net6
 redistribute connected
 no auto-summary
 no synchronization
 exit-address-family
 !
ip classless
!
gatekeeper
 shutdown
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```