UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO DE ESPECIALIZAÇÃO EM CONFIGURAÇÃO E GERENCIAMENTO DE SERVIDORES E EQUIPAMENTOS DE REDE

MAURICIO DADA FONSECA DE FREITAS

PROJETO E SIMULAÇÃO DE REDE MPLS

MONOGRAFIA

CURITIBA
2013

MAURICIO DADA FONSECA DE FREITAS

PROJETO E SIMULAÇÃO DE REDE MPLS

Monografia apresentada como requisito parcial para a obtenção do grau de Especialista em Configuração e Gerenciamente de servidores e equipamentos de rede, do Departamento Acadêmico de Eletrônica da Universidade Tecnológica Federal do Paraná – UTFPR
Orientador: Prof. Dr. Augusto Foronda

CURITIBA
2013

RESUMO

FREITAS, Mauricio D. F. de – **<Título>.** 2013. 68 f. Monografia (Especialização em Configuração e Gerenciamento de Servidores e Equipamentos de Redes). Universidade Tecnológica Federal do Paraná. Curitiba, 2013.

Esta monografia busca sintetizar a teoria básica da tecnologia MPLS (MultiProtocol Label Switching). Esta serve como um primeiro contato com a tecnologia, para alguém que já possua conhecimento básico sobre redes WAN (Wide Area Network). É feito um embasamento teórico dos principais conceitos e estes são ilustrados através de uma simulação de rede. A simulação parte da escolha de equipamentos compatíveis, definição da topologia e conexões, distribuição de endereços IP, configuração dos protocolos, análise aprofundada das tabelas de encaminhamento e noções básicas de *troubleshooting*.

Palavras-chave: MPLS. Projeto de Rede. Simulações de Redes. *Troubleshooting*. LDP. L2VPNs.

ABSTRACT

FREITAS, Mauricio D. F. de – **<Título>.** 2013. 68 pages.

Monograph (Specialization in Configuration and Management of Servers and Network Equipment) - Federal Technological University of Paraná. Curitiba, 2013.

This monograph aims at synthesizing the basic theory around the MPLS (MultiProtocol Label Switching) technology. It serves as a first contact with the technology, requiring some basic prior knowledge in WAN (Wide Area Networks) concepts. The project comprises the explanation of the main concepts and those are illustrated by a simulation of the network. The simulation begins with choosing a compatible equipment, defining the topology and connections, distributing IP addresses, configuring all protocols, meticulously analyzing the forwarding tables and a introducing some basic notion of troubleshooting.

Keywords: MPLS. Network Project. Network Simulation. Troubleshooting. LDP. L2VPNs.

LISTA DE ILUSTRAÇÕES

LISTA DE SIGLAS

AToM – Any Transport over MPLS
BDR – Backup Designated Router
BGP – Border Gateway Protocol
BoS – Bottom of Stack
CEF – Cisco Express Forwarding
CLI – Command Line Interface
DR – Designated Router
EoMPLS – Ethernet over MPLS
FEC – Forwarding Equivalence Class
IANA – Internet Assigned Numbers Authority
ICMP – Internet Control Message Protocol
IETF – Internet Engineering Taskforce
IGP – Interior Gateway Protocol
IP – Internet Protocol
IS-IS – Intermediate System to Intermediate System
LDP – Label Distribution Protocol
LER – Label Edge Router
LSP – Label Switched Path
LSR – Label Switched Router
MPLS – Multiprotocol Label Switching
OAM - Operations, Administration and Management
OSPF – Open Shortest Path First
P – Provider Router
PE – Provider Edge
PW - Pseudowire
QoS – Quality of Service
RFC – Request For Comments
TCP – Transmission Control Protocol
TE – Traffic Engineering
TLV – Type Lenght Value
TTL – Time to Live
UDP – User Datagram Protocol
VLAN – Virtual Local Area Network
VPLS – Virtual Private LAN Service
VPN – Virtual Private Network
VPWS – Virtual Private Wire Service
WAN – Wide Area Network

SUMÁRIO

1 INTRODUÇÃO

1.1 TEMA

O *Multiprotocol Label Switching* (MPLS) é um padrão IETF de redes de telecomunicação que utiliza “rótulos” ou “etiquetas” (*labels*) como maneira de comutar pacotes (GHEIN, 2007). O protocolo é definido pela RFC 3031 (IETF, 2001).

Estes rótulos são valores curtos e de comprimento fixo de 32 bits. Este valor é arbitrário e somente significativo para cada roteador em questão.

Este conceito de chaveamento por rótulos não é novo ou particular do MPLS, também sido utilizado em tecnologias WAN como ATM e Frame Relay (GHEIN, 2007).

Não há um conceito de endereçamento fim-a-fim, sendo que estas *labels* são inseridas nos pacotes IP (processo conhecido como encapsulamento), permitindo que os roteadores comutem estes pacotes através deste novo rótulo, ao invés do endereço de IP de destino (GHEIN, 2007).

Uma das principais diferença entre o IP e protocolos *label switching* é que o rótulo, ou pelo menos a interpretação de seu valor, é alterado a cada salto, diferente do cabeçalho IP que sempre possui o mesmo endereço de destino (GHEIN, 2007). Este é justamente uma das características que tornou o MPLS popular, já que o torna uma excelente opção para soluções de VPN (SANTOS, 2003).

Como principais benefícios do MPLS é possível citar: Infraestrutura de rede unificada, núcleo da rede sem BGP, modelo ponto-a-ponto de VPNs, fluxo de tráfego otimizado e possibilidades de *Traffic Engineering* (TE) (GHEIN, 2007);

1.2 OBJETIVOS

1.2.1 Objetivos Gerais

O trabalho a ser realizado visa principalmente apontar diretrizes de projeto de redes MPLS.

Por diretrizes, entenda-se um conjunto de itens essenciais e de boas práticas na concepção de uma rede.

Como maneira de obter conclusões, também se pretende simular a rede projetada, via software, utilizando imagens de *firmware* de equipamentos Cisco.

1.2.2 Objetivos Específicos

Familiarizar-se com o funcionamento da tecnologia MPLS, ou seja, levantar e estudar literatura específica, entender como o protocolo se comporta, em quais situações este é apropriado.

Realizar um projeto de exemplo, partindo do detalhamento de conexões físicas dos equipamentos, configurações básicas da infraestrutura (Endereços IP, VLANs, Protocolos de roteamento, etc.), infraestrutura MPLS até o provisionamento de circuitos virtuais para serviços.

Ilustrar os conceitos apresentados com capturas de pacotes nos links e da saída de comando de visualização através da interface CLI dos equipamentos.

1.3 JUSTIFICATIVA

A tecnologia MPLS já faz parte do vocabulário de muitas empresas da área de telecomunicação e há credibilidade de que esta venha a ser aplicada em larga escala e se tornar algo dominante (SANTOS, 2003).

No entanto, por se tratar de uma tecnologia nova, ainda há um receio do mercado em concretizar a implementação do MPLS. Parte disso se deve a pequena quantidade de profissionais com conhecimento aprofundado na tecnologia (SANTOS, 2003).

Por maior que seja esse receio, há demonstração de um forte interesse em se conhecer mais e dominar a tecnologia, de forma a esta ser assimilada melhor pelo mercado (SANTOS, 2003).

A melhor maneira de solucionar estes desafios é tornar familiar o protocolo para os profissionais interessados, mostrando boas práticas de projeto e implementação da tecnologia, bem como apontando os maiores desafios práticos e como solucioná-los.

1.4 METODOLOGIA

O trabalho proposto é tanto teórico, quanto prático. Pretende-se apresentar uma revisão teórica do funcionamento do protocolo MPLS, compreendendo a apresentação e seus vários elemento, incluindo nomenclatura e função, as dinâmicas do processo e como estes devem ser previstos e considerados no projeto da rede. Tendo uma boa visão geral do protocolo, será demonstrado como esta teoria pode ser aplicada na prática, através de um exemplo de projeto. Finalmente, com o projeto pronto, será feito uma simulação da rede, de forma a concluir realmente quão eficiente foi o projeto.

Considerando que existem tecnologias similares ao MPLS, haverá também uma comparação com um destes protocolos legados de forma a evidenciar pontos em comum entre os dois protocolos e em quais aspectos a nova tecnologia é superior.

1.5 EMBASAMENTO TEÓRICO

O MPLS e a maioria dos protocolos auxiliares para sua configuração e operação são padrões abertos e definidos em publicações do tipo *Requests for Comments* (RFC) pela *Internet Engineering Task Force* (IETF). Os fabricantes de equipamentos que desejam realizar implementação destes protocolos devem seguir estas publicações. Por este motivo, estas são fontes excelentes para consulta de detalhes específicos destes protocolos, como nomenclatura, estados de operação, fluxo de mensagens e formatação de pacotes. As principais RFCs utilizadas neste trabalho são a RFC 2328 (1998), 3031 (2001), 3036 (2001) e 4379 (2006), relativas ao OSPFv2, MPLS, LDP e MPLS OAM, respectivamente.

As RFCs possuem um foco mais operacional dos protocolos e muitas vezes é interessante ter uma visão das possíveis aplicação destas tecnologias, em contextos reais. Existe uma vasta bibliografia com esse foco, geralmente elaborada por profissionais com experiência na área e com certificações dos principais fabricantes como Cisco e Juniper.

Os livros *MPLS Fundamentals*, Ghein (2007, p. 1) e *MPLS-Enabled Applications*, Minei e Lucek (2005, p. 1), cobrem muito bem as principais

aplicações do MPLS, desde a introdução dos principais conceitos até noções mais avançadas para situações não triviais ou específicas.

1.6 ESTRUTURA

A monografia é composta por quatro capítulos. O primeiro capítulo é a introdução do trabalho, sendo apresentados o tema, os objetivos a serem atingidos, a justificativa da escolha e os problemas a serem resolvidos.

Também nesta primeira parte, apresenta-se o embasamento teórico, procedimento metodológico e a estrutura da monografia.

O capítulo dois trata do referencial teórico do projeto. Neste serão apresentados os conceitos básicos do MPLS, o protocolo LDP, para distribuição de rótulos e o funcionamento dos serviços de VPN utilizando a tecnologia.

O capítulo três é o desenvolvimento prático do trabalho, aplicando todos os conceitos tratados no capítulo dois de forma a projetar, configurar e analisar um simulação de rede MPLS.

O capítulo quatro conclui o trabalho, fazendo considerações gerais sobre todo o conjunto desenvolvido nos capítulos anteriores e um parecer final.

Após o capítulo quatro e a listagem das referências bibliográficas, há um apêndice com a configuração final dos equipamentos no cenário simulado.

2 REFERENCIAL TEÓRICO

2.1 MPLS

O MPLS (Multiprotocol Label Switching) é um protocolo de roteamento baseado em pacotes rotulados, onde cada rótulo (*label*) apresenta um índice na tabela de roteamento do próximo roteador.

Uma propriedade fundamental do MPLS é que ele pode ser utilizado para "transportar" diversos tipos de tráfego através do núcleo de uma rede. Ou seja, apenas os roteadores nas extremidades da rede necessitam entender o protocolo do qual o tráfego pertence, poupando os roteadores de core deste papel (MINEI e LUCEK, 2005, p. 6).

2.1.1 Formatação dos Rótulos

Um rótulo MPLS é um campo de 32 bits, como ilustrado na Figura 1.

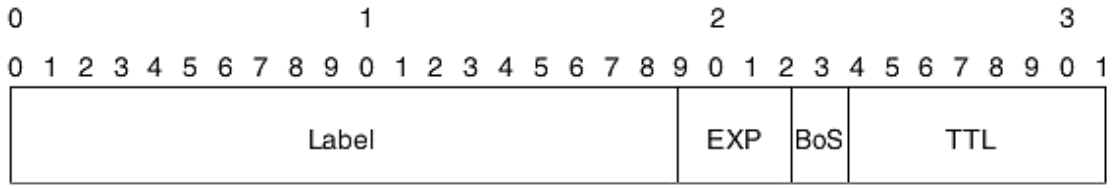

Figura 1 - Campos de um rótulo MPLS
Fonte: Ghein, 2007

Os primeiros 20 bits são o valor do rótulo, permitindo até 1048575 valores. Destes, os valores de 0 a 15 possuem significados especiais (GHEIN, 2007, p. 26).

Os bits entre 20 e 22 são bits chamados experimentais (EXP) por inicialmente não possuírem uso previsto, porém podem ser aproveitados para realizar QoS (GHEIN, 2007).

O bit 23 é chamado *Bottom of Stack* (BoS) e é utilizado para o empilhamento de rótulos (GHEIN, 2007). Este conceito será melhor detalhado na seção 2.1.6.

Os últimos 8 bits são utilizados como contador de *Time To Live* (TTL). Este serve para evita que um pacote fique preso em um loop. O campo funciona

de maneira análoga do cabeçalho IP, sendo decrementado a cada salto e caso este atinja o valor 0, o pacote é descartado (GHEIN, 2007).

2.1.2 Label Switch Router

Os Label Switch Router (LSR) são roteadores que suportam MPLS. Estes são capazes de interpretar os rótulos MPLS e realizar operações de inserção, retirada ou troca destes rótulos (GHEIN, 2007, p. 29).

É possível classificar os LSRs em três tipo, conforme sua função: *Ingress*, *Egress* ou *Intermediate.* Os LRSs de ingresso, recebem pacotes não rotulados, fazem a inserção do rótulo e encaminham estes pacotes através dos devidos *links*. Os de egresso, fazem a ação contrária, retirando o rótulo dos pacotes, antes do encaminhamento. Os intermediários, fazem operações de troca de rótulos e encaminham o pacote para um determinado *link* de dados (GHEIN, 2007).

Os LSRs que possuem as funções Ingress e Egress, são os roteadores que ficam na borda da rede e também podem ser denominados Label Edge Router (LER) (MINEI e LUCEK, 2005, p. 7).

No contexto de VPNs, um LSR intermediário pode ter a nomenclatura de Provider (P) e um LER pode também ser chamado de Provider Edge (PE) (GHEIN, 2007).

A Figura 2 ilustra uma topologia MPLS típica com esta outra nomenclatura.

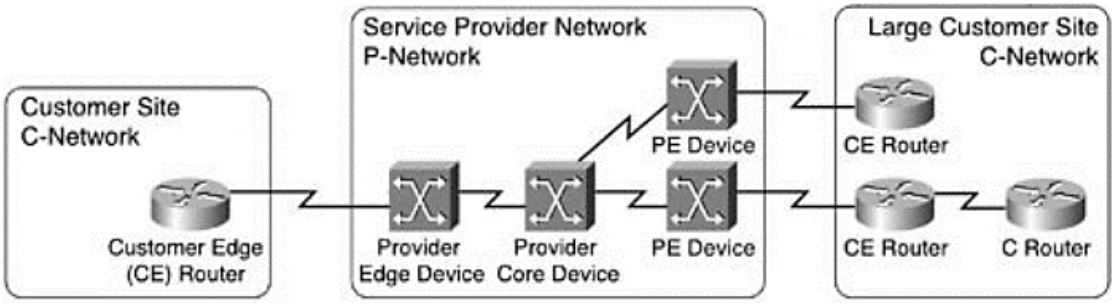

Figura 2 - Exemplo de rede MPLS com terminologia no contexto de VPNs
Fonte: Guichard, Pepelnjak e Apcar, 2003

2.1.3 Label Switched Path

Um Label Switched Path (LSP) é uma sequência de roteadores que um pacote rotulado percorre através de uma rede MPLS. Este começa em um LER de ingresso e termina no LER de saída. (GHEIN, 2007, p. 29). A Figura 3 ilustra um LSP formado sobre um conjunto de LSRs.

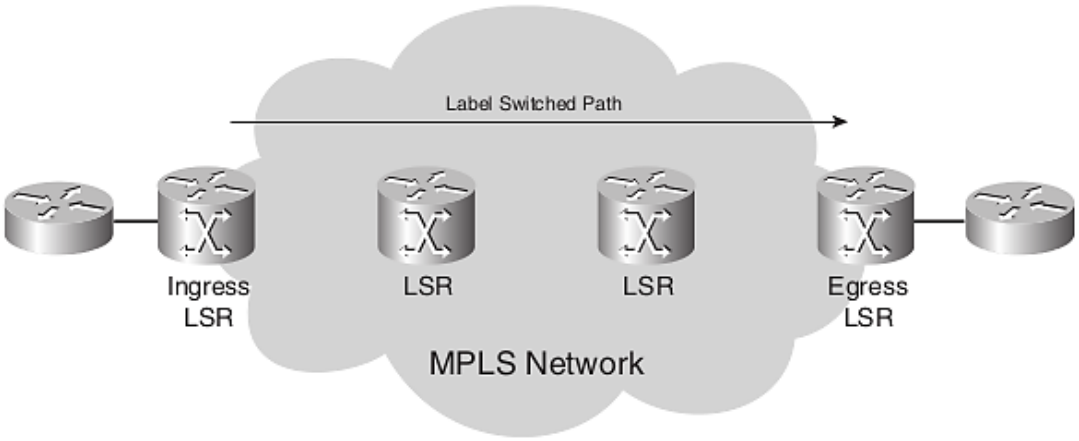

Figura 3 - Ilustração de um LSP simples em uma rede MPLS
Fonte: Ghein, 2007

Um LSP é unidirecional. O fluxo de pacotes rotulados no sentido contrário entre os mesmos LERs é considerado um outro LSP (GHEIN, 2007, p. 30).

Os LSRs adjacentes em uma topologia necessitam estar de acordo com quais *labels* utilizarem na definição de cada LSP. Para isto, é necessário um mecanismo para dizer aos roteadores quais rótulos utilizar, não havendo um significado global para um único rótulo em toda a rede, sendo o significado apenas local a cada par de LSPs (GHEIN, 2007, p. 32).

Há duas maneiras de distribuir rótulos: adaptando um protocolo de roteamento já existente na rede ou utilizando um protocolo separado apenas com esta função (GHEIN, 2007, p. 33). O LDP, utilizado neste trabalho, é o segundo caso.

2.1.4 Forwarding Equivalence Class

Forwarding Equivalence Class (FEC) é uma classe de pacotes que são encaminhados através do mesmo trajeto e são tradados da mesma maneira para encaminhamento. Todos os pacotes pertencentes à mesma FEC possuem o mesmo rótulo (GHEIN, 2007).

O LSR de ingresso é o responsável por determina a qual FEC um certo pacote pertence. Alguns exemplos de FEC são: pacotes L3 com IP de destino pertencente à certo prefixo; pacotes Multicast pertencendo ao mesmo grupo;

pacotes L2 recebidos em determinada (sub)interface; pacotes L3 com IP de destino que pertença à um conjunto de prefixos BGP com o mesmo *next-hop* (GHEIN, 2007).

2.1.5 Penultimate Hop Popping

Em alguns casos simples, o uso de um único rótulo MPLS é suficiente, como é o caso de transportar tráfego de IPs públicos através da rede. Neste caso, quando um pacote rotulado chega ao LER de egresso, este consulta uma tabela de roteamento para poder encaminhar adiante este pacote IP (MINEI e LUCEK, 2005).

Geralmente é utilizado um esquema chamado Penultimate Hop Popping, em que um LSR intermediário anterior ao LER de egresso (o penúltimo roteador ao longo do LSP) retira o rótulo MPLS antes de encaminhar ao LER um pacote IP não encapsulado (MINEI e LUCEK, 2005).

Isto simplifica o processo realizado no LER de egresso, já que este não necessita retirar o rótulo MPLS e faz somente a consulta à tabela de roteamento (MINEI e LUCEK, 2005).

2.1.6 Empilhamento de Rótulos

Para algumas aplicações, um único rótulo MPLS pode não ser suficiente. Isto ocorre quando os LERs de uma rede estejam envolvidos em mais de um tipo de serviço, como L3 VPN, L2 VPN e VPLSs. Nestes casos, o LER de egresso deve saber para qual serviço e para qual instância do serviço (qual cliente) o pacote pertence. Isto é atingido através da inserção de um rótulo adicional, pelo LER de ingresso, correspondente ao tipo e instância do serviço (MINEI e LUCEK, 2005, p. 9). A Figura 4 ilustra mais de uma *label* MPLS empilhada:

Figura 4 - Empilhamento de rótulos
Fonte: Ghein, 2007

A Figura 4 também mostra que o bit BoS de todos os rótulos é 0, com exceção do rótulo de fundo, em que o bit é 1 (GHEIN, 2007, p. 27). O rótulo de fundo é o último a ser tratado pelos roteadores.

A habilidade de empilhar rótulos desta maneira dá ao MPLS propriedades de multiplexação e hierarquização, permitindo a um mesmo LSP entre um ponto de ingresso e egresso a transportar todo tipo de tráfego entre este dois pontos (MINEI e LUCEK, 2005).

A Figura 5 auxilia o entendimento da aplicação do conceito de empilhamento.

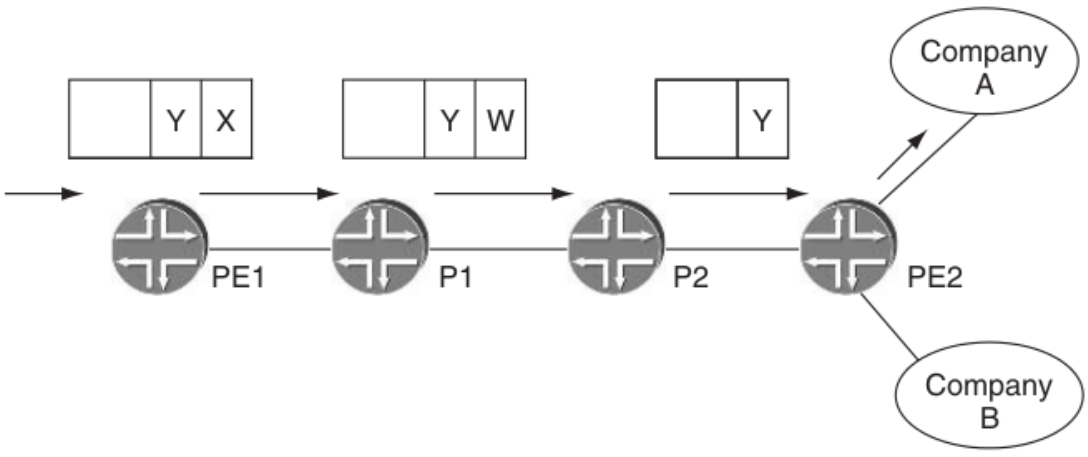

Figura 5 - Encaminhamento de de pacotes contendo mais de um rótulo.
Fonte: Minei e Lucek, 2005

O pacote deixa o PE1 com um rótulo interno (Y) empilhado em um externo (X). Os roteadores P1 e P2 fazem uma consulta ao rótulo externo, sem necessitar interpretar ou tomar ação baseado no rótulo interno. P1 faz uma operação de *swap* do rótulo X para o W. Caso o PHP esteja sendo utilizado,

como geralmente é o caso, o roteador P2 retira o rótulo externo e envia o restante do pacote para o PE2. Quando o pacote chega ao PE2, a única label presente no pacote é a label interna original, Y, que é então utilizada para o PE2 identificar o pacote como pertencendo à instância de VPN da companhia A (MINEI e LUCEK, 2005).

Este conceito de encapsulamento é fundamental para o entendimento de VPNs.

2.2 LABEL DISTRIBUTION PROTOCOL

O LDP é definido na RFC 3036, que é resultado do MPLS Working Group da IETF. O LDP foi criado especificamente para a distribuição de rótulos ao longo da rede (MINEI e LUCEK, 2005, p. 13).

O LDP é focado em realizar este papel de distribuição de rótulos, dependendo de um Interior Gateway Protocol (IGP) para todas as decisões de roteamento (MINEI e LUCEK, 2005). Uma implicação desta dependência é que os LSPs formados sempre seguem o caminho mais curto determinado pelo IGP, sendo alterado conforme o protocolo IGP converge (MINEI e LUCEK, 2005, p. 14). Isto é importante em casos de redundância de topologia e também implica que o tempo de convergência dos serviços MPLS, em caso de quedas de link, irá depender do tempo de reconvergência do protocolo de roteamento escolhido.

A especificação original do LDP foi definida para a formação de LSPs para FECs representando endereços IPv4 ou IPv6. No entanto, esta especificação foi estendida para a formação e sinalização de *pseudowires* (MINEI e LUCEK, 2005, p. 13).

2.2.1 Cabeçalho LDP

O LDP utiliza um mecanismo de codificação *Type-Length-Value* (TLV) para transmitir a maioria das informações. (IETF, 2001)

O tipo identifica qual informação está sendo trocada e determina como deve ser feito o resto da decodificação da informação. O valor é justamente a informação a ser decodificada e o comprimento é o delimitador da informação (MINEI e LUCEK, 2005, p. 13).

O uso de TLVs é interessante pois permite estender as funcionalidades do protocolo e facilita ignorar informações não conhecidas por um roteador que não possua certas funcionalidades (MINEI e LUCEK, 2005).

2.3 VIRTUAL PRIVATE NETWORK

Uma Virtual Private Network (VPN) pode ser definida como uma rede que fornece conexão à localizações distintas de diversos clientes, enquanto compartilhando da mesma infraestrutura, porém mantendo as mesmas características de acesso e segurança que uma rede privada (PEPELNJAK e GUICHARD, 2001, p. 95). O requisito básico de conexão de uma VPN é que todos os *sites* de um mesmo cliente estejam conectados e que estes estejam completamente isolados dos demais clientes (GHEIN, 2007, p. 173).

Uma VPN pode oferecer conexão de camada 2 ou camada 3, do modelo OSI (GHEIN, 2007). Neste trabalho serão focadas as VPNs de camada 2, por serem mais simples.

2.3.1 Any Transport over MPLS

As soluções Any Transport over MPLS (AToM) surgiram após o sucesso das VPNS MPLS em camada 3, utilizadas para carregar tráfego IP dos clientes. No entanto, ainda havia um grande mercado para tecnologias de *leased lines*, como ATM e Frame Relay. Muitos clientes não haviam interesse em trocar migrar de tecnologia com o interesse em manter completo controle sobre suas redes e a maneira como estas são construídas ou por possuírem equipamentos antigos que utilizavam protocolos que não o IP. Isto fazia com que um provedor de serviços com uma rede com a tecnologia de MPLS ainda tinha que manter a rede legada. (GHEIN, 2007, p. 383).

Uma maneira de solucionar isto foi o desenvolvido o conceito de AToM, de forma a permitir conexão em camada 2 entre os domínios do cliente, utilizando a mesma infraestrutura MPLS utilizada para as L3VPNs (GHEIN, 2007).

A arquitetura utilizada pelao AToM para transporte do tráfego L2 é baseada em *pseudowires* (PW). Este carrega o tráfego L2 de uma borda da rede

à outra, através do *backbone* da rede, de maneira transparente, como se fosse um único cabo (GHEIN, 2007, p. 384-385).

A Figura 6 ilustra um *psweudowire* formado em cima de uma rede MPLS.

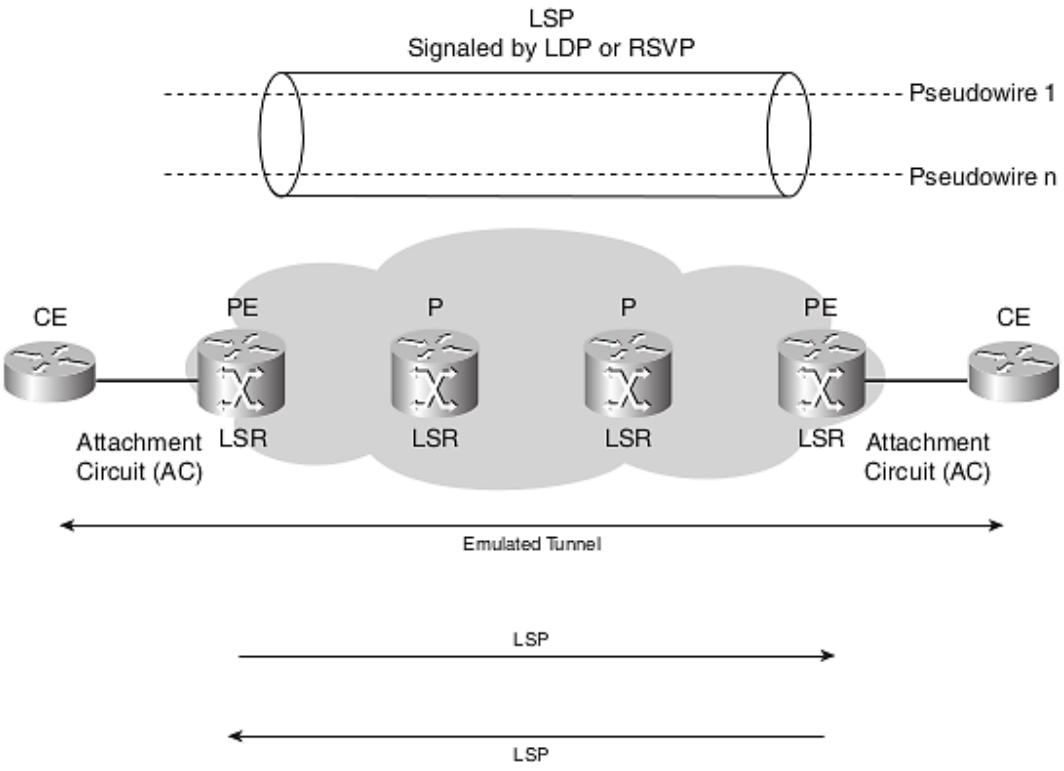

Figura 6 - Ilustração de um pseudowire em uma infraestrutura MPLS
Fonte: Ghein, 2007

3 SIMULAÇÃO

Para poder demonstrar todos os conceitos teóricos cobertos, foi montada uma rede utilizando simuladores da Cisco.

Para realizar a simulação, é utilizado o GNS3.

O GNS3 é um software de código aberto que simula redes complexas tentando chegar o mais próximo possível do comportamento de redes reais, sem necessitar hardware dedicado para isto, como com roteadores e switchs. (GNS3, 2007)

Para carregar e rodar as imagens de roteadores da Cisco, o GNS3 utiliza o Dynamips. (GNS3, 2007)

3.1 INFRAESTRUTURA BASICA

3.1.1 Escolha do equipamento e IOS

Optou-se por simular roteadores da série 7200.

Estes são roteadores *Enterprise*, anunciados como *High-Performance Services Aggregation*. (CISCO, 2012) Estes fornecem uma grande variedade de serviços e há bastante aplicação como agregadores de uma rede de WAN. (CISCO, 2003)

Apesar de serem voltados para redes *enterprise*, possuem todas as funcionalidades necessárias para a simulação proposta.

A imagem utilizada para as simulações é a c7200-adventerprisek9-mz.150-1.M.bin.

Para verificar todas as *features* disponíveis na imagem, utilizou-se o Cisco *Feature Navigator*, fornecido no site deles.

As *features* verificadas foram:

- OSPF;
- MPLS LDP – Label Distribution Protocol (LDP);
- Any Transport over MPLS (AToM): Ethernet over MPLS: Port Mode (EoMPLS);
- Any Transport over MPLS (AToM): Ethernet over MPLS (EoMPLS);

http://tools.cisco.com/ITDIT/CFN/jsp/by-feature-technology.jsp

3.1.2 Topologia

Optou-se por criar uma topologia em anel, por ser uma topologia simples, comum, que possui redundância e que permite a demonstração de todas as funcionalidades.

Foram utilizados 5 roteadores para permitindo uma quantidade razoável de saltos e melhor ilustrar as ações realizadas nos rótulos MPLS dos pacotes.

Para conexão entre os roteadores, foram utilizados links Gigabit Ethernet, através de 2 módulos de interface PA-GE (CISCO, 2012).

O diagrama da disposição e conexão dos roteadores está representado na Figura 7 - Topologia em Anel e links Ethernet.

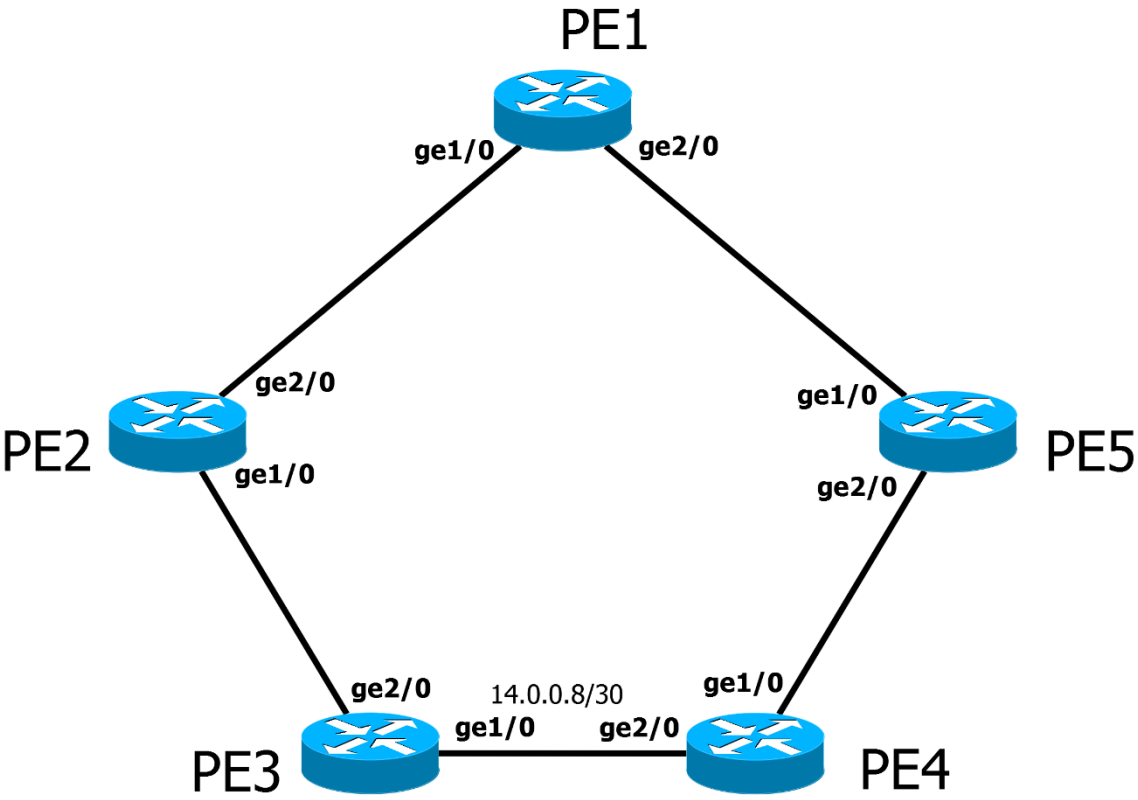

Figura 7 - Topologia em Anel e links Ethernet
Fonte: Autoria Própria

3.1.3 Configuração das Interfaces

Como todos os links entre os roteadores são ponto-a-ponto, optou-se por distribuir endereços IP de redes com máscara /30.

Utilizou-se o prefixo 14.0.0.0 apenas para destacar os valores como IPs de interface. As redes são adjacentes, portanto ficaram 14.0.0.0/30, 14.0.0.4/30, 14.0.0.8/30, 14.0.0.12/30 e 14.0.0.16/30.

Para a configuração do OSPF e depois do LDP, é interessante possuir interfaces de *loopback*. Interfaces de *loopback* são interfaces virtuais e que sempre se mantêm no estado UP, a menos que estejam em *shutdown* administrativo (CISCO, 2001).

As interfaces de *loopback* terão endereços de IP com máscara /32, já que são interfaces virtuais e não haverá outros hosts diretamente conectados a elas.

Para destacar o IPs das *loopbacks*, utilizou-se o prefixo 8.0.0.0, alterando apenas o ultimo octeto para refletir o mesmo índice dos roteadores.

A Figura 8 - Diagrama da Topologia com os endereços IP distribuídos. resume a distribuição de IPs nos *switchs*.

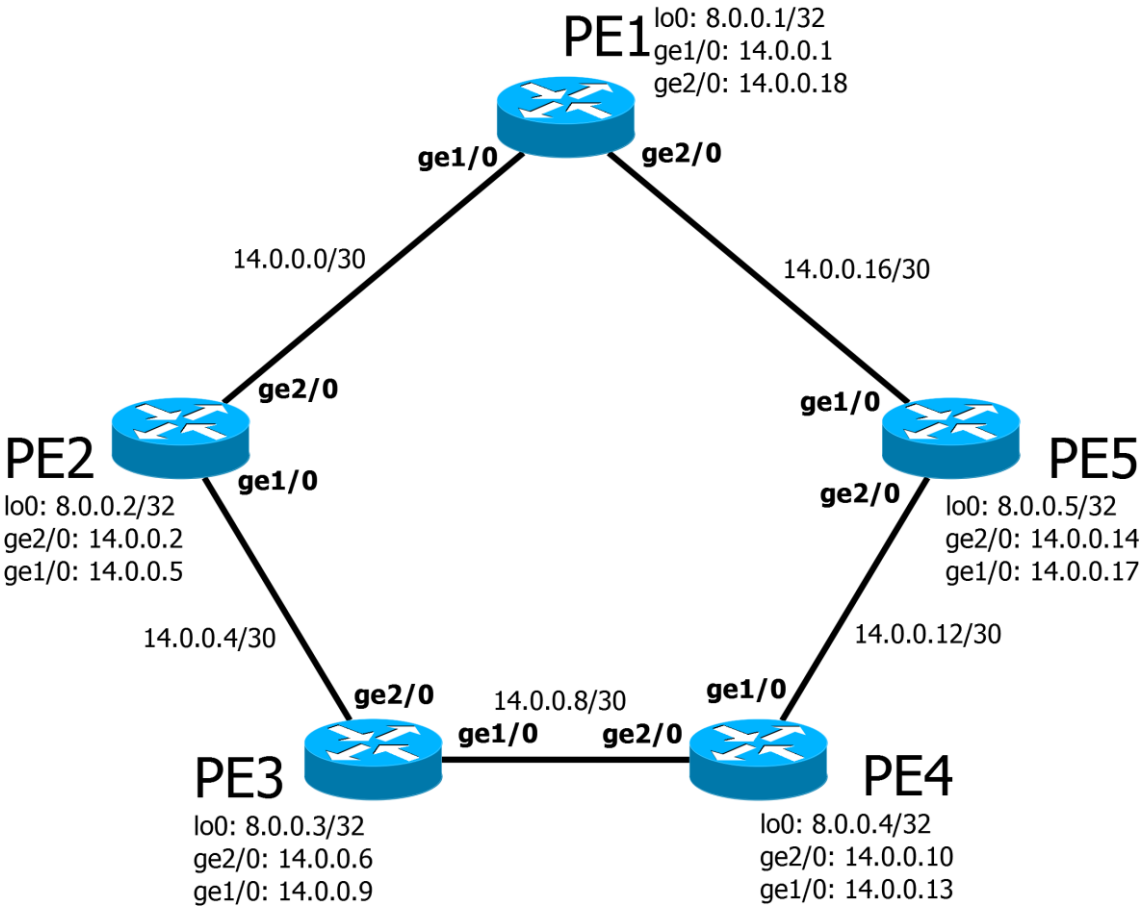

Figura 8 - Diagrama da Topologia com os endereços IP distribuídos.
Fonte: Autoria Própria

Para configurar as interfaces, foi utilizada a seguinte sequência de comandos:

```
PE1(config)#interface gigabitEthernet 1/0
PE1(config-if)#ip address 14.0.0.1 255.255.255.252
PE1(config-if)#no shutdown
*Dec  7 20:49:29.547: %LINK-3-UPDOWN: Interface GigabitEthernet1/0,
changed state to up
*Dec  7 20:49:30.547: %LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet1/0, changed state to up
```

Figura 9 - Configuração das Interfaces
Fonte: Autoria Própria

Todas as configurações foram verificadas com o comando **show ip interface brief**:

```
PE1#show ip interface brief
Interface                  IP-Address      OK? Method Status
Protocol
FastEthernet0/0            unassigned      YES NVRAM  administratively
down down
GigabitEthernet1/0         14.0.0.1        YES NVRAM  up
up
GigabitEthernet2/0         14.0.0.18       YES NVRAM  up
up
Loopback0                  8.0.0.1         YES NVRAM  up
up
```

Figura 10 - Visualização do estado das interfaces
Fonte: Autoria Própria

Finalmente, testou-se a conectividade das redes diretamente conectadas com o *ping*:

```
PE1#ping 14.0.0.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 14.0.0.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/41/56
ms
```

Figura 11 - Verificação de conexão IP entre os PEs
Fonte: Autoria Própria

3.1.4 Protocolo de Roteamento

Para as demais redes, são necessárias configurações de rotas. É possível conectar toda a infraestrutura IP apenas com rotas estáticas, porém há um custo administrativo muito grande em fazer isto. A melhor maneira é rodar um protocolo de roteamento.

Assume-se que a rede a ser simulada seja de um único provedor e portanto esteja contida dentro de um mesmo sistema autônomo (AS). É possível

utilizar qualquer protocolo do tipo IGP, como o OSPF ou o IS-IS (GHEIN, 2007, p. 9).

Optou-se pelo OSPF, que é um protocolo bem estabelecido, permite interoperabilidade entre fabricantes e possui tempos de convergência rápidos (TENENBAUM, 2003, p. 348).

Para a configurar o OSPF, utilizou-se a seguinte sequência de comandos:

```
PE1(config)#router ospf 1
PE1(config-router)#router-id 8.0.0.1
PE1(config-router)#network 8.0.0.1 0.0.0.0 area 0
PE1(config-router)#network 14.0.0.0 0.0.0.3 area 0
PE1(config-router)#network 14.0.0.16 0.0.0.3 area 0
```

Figura 12 - Configuração do OSPF em PE1
Fonte: Autoria Própria

Onde:

- **router ospf <instance>** – Habilita o OSPF com instância 1;
- **router-id <x.x.x.x>** - Força um valor para o Router ID;
- **network <net> <wildcard> área <área>** - Configura as redes que formaram adjacência e serão anunciadas no OSPF;

Como as redes das interfaces físicas usam máscara /30, o *wildcard* delas é 0.0.0.3 (b00000011). É importante incluir também a interface de *loopback* que será anunciada aos demais roteadores. Como estas têm máscara /32 o *wildcard* é 0.0.0.0.

Por padrão, é assimilado o maior endereço de IP configurado no equipamento ou o maior endereço de IP das interfaces de *loopback* (HALABI, 1996, p. 14).

Se o *Router ID* utilizasse um endereço de uma interface física e esta viesse a ficar indisponível, o OSPF teria de recalcular um novo RID e anunciar este a todos os demais roteadores. Por isto é importante utilizar a interface de *loopback*, já que esta estará sempre disponível, garantindo que o RID seja o mesmo, independente do estado dos links (THOMAS, 2003, p. 264).

Após repetir as configurações com as devidas redes em todos os roteadores, as adjacências são formadas.

```
*Dec  7 20:49:35.603: %OSPF-5-ADJCHG: Process 1, Nbr 8.0.0.2 on
GigabitEthernet1/0 from LOADING to FULL, Loading Done
```

Figura 13 - Log de troca de status de adjacência OSPF
Fonte: Autoria Própria

Para verificar todas adjacências formadas em um equipamento, utiliza-se o comando **show ip ospf neighbor**:

```
PE1#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address
Interface
8.0.0.5           1   FULL/DR         00:00:39    14.0.0.17
GigabitEthernet2/0
8.0.0.2           1   FULL/BDR        00:00:39    14.0.0.2
GigabitEthernet1/0
```

Figura 14 - Verificação de status das adjacências OSPF
Fonte: Autoria Própria

Nota-se que houve eleição do DR e do BDR. Em interfaces seriais ponto a ponto, o estado do OSPF as adjacências são sempre formadas sem o conceito de DR e BDR. (HALABI, 1996, p. 21)

O mesmo não ocorre com interfaces Ethernet, que permitem broadcast e podem possuir mais de dois roteadores no mesmo domínio. Como os links entre os roteadores nesse cenário são ponto a ponto, esse comportamento se torna desnecessário.

É possível suprimir a eleição do DR através do comando: **ip ospf network point-to-point** nas interfaces:

```
PE1(config)#interface gigabitEthernet 1/0
PE1(config-if)#ip ospf network point-to-point
*Dec  7 21:58:20.811: %OSPF-5-ADJCHG: Process 1, Nbr 8.0.0.2 on
GigabitEthernet1/0 from FULL to DOWN, Neighbor Down: Interface down or
detached
*Dec  7 21:58:21.715: %OSPF-5-ADJCHG: Process 1, Nbr 8.0.0.2 on
GigabitEthernet1/0 from LOADING to FULL, Loading Done
PE1(config-if)#exit
PE1(config)#interface gi
PE1(config)#interface gigabitEthernet 2/0
PE1(config-if)#ip ospf network point-to-point
*Dec  7 21:58:31.023: %OSPF-5-ADJCHG: Process 1, Nbr 8.0.0.5 on
GigabitEthernet2/0 from FULL to DOWN, Neighbor Down: Interface down or
detached
*Dec  7 21:58:31.199: %OSPF-5-ADJCHG: Process 1, Nbr 8.0.0.5 on
GigabitEthernet2/0 from LOADING to FULL, Loading Done
```

Figura 15 - Configuração de adjacência OSPF point-to-point
Fonte: Autoria Própria

Verificando novamente as adjacências:

```
PE1#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address
Interface
8.0.0.5           0   FULL/  -        00:00:35    14.0.0.17
GigabitEthernet2/0
```

```
8.0.0.2           0   FULL/  -        00:00:32    14.0.0.2
GigabitEthernet1/0
```

Figura 16 - Verificação das adjacências OSPF point-to-point
Fonte: Autoria Própria

Tendo repetido estas configurações em todos os roteadores, todas as rotas são aprendidas:

```
PE1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B -
BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS
level-2
       ia - IS-IS inter area, * - candidate default, U - per-user
static route
       o - ODR, P - periodic downloaded static route, + - replicated
route

Gateway of last resort is not set

      8.0.0.0/32 is subnetted, 5 subnets
C        8.0.0.1 is directly connected, Loopback0
O        8.0.0.2 [110/2] via 14.0.0.2, 00:03:59, GigabitEthernet1/0
O        8.0.0.3 [110/3] via 14.0.0.2, 00:03:59, GigabitEthernet1/0
O        8.0.0.4 [110/3] via 14.0.0.17, 00:03:49, GigabitEthernet2/0
O        8.0.0.5 [110/2] via 14.0.0.17, 00:03:49, GigabitEthernet2/0
      14.0.0.0/8 is variably subnetted, 7 subnets, 2 masks
C        14.0.0.0/30 is directly connected, GigabitEthernet1/0
L        14.0.0.1/32 is directly connected, GigabitEthernet1/0
O        14.0.0.4/30 [110/2] via 14.0.0.2, 00:03:59,
GigabitEthernet1/0
O        14.0.0.8/30 [110/3] via 14.0.0.17, 00:03:49,
GigabitEthernet2/0
                     [110/3] via 14.0.0.2, 00:03:59,
GigabitEthernet1/0
O        14.0.0.12/30 [110/2] via 14.0.0.17, 00:03:49,
GigabitEthernet2/0
C        14.0.0.16/30 is directly connected, GigabitEthernet2/0
L        14.0.0.18/32 is directly connected, GigabitEthernet2/0
```

Figura 17 - Verificação da tabela de roteamento completa
Fonte: Autoria Própria

É possível verificar a conectividade de todas as redes com um *ping*:

```
PE1#ping ip 8.0.0.3 source loopback 0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.0.0.3, timeout is 2 seconds:
Packet sent with a source address of 8.0.0.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/55/84
ms
```

Figura 18 - Verificação de conectividade entre as loopbacks do PE1 e PE3

Fonte: Autoria Própria

Neste caso utilizou-se um *ping* estendido para garantir que havia conectividade entre as *loopbacks*.

3.2 LDP BASIC

Com a infraestrutura IP configurada e com todos os roteadores tendo conectividade com todas as redes da topologia, é possível começar a configurar o MPLS nos equipamentos.

É possível criar os LSPs estaticamente, assim como com as rotas, porém isto não é prático. Para realizar isto de maneira dinâmica, é possível utilizar o protocolo LDP.

De forma a poder trocar informações de *labels* para construir os LSPs, um roteador deve fazer a descoberta dos outros elementos na rede.

Em sua forma mais básica, o LDP faz essa descoberta através do envio de pacotes de *Hello* em todas as interfaces com o protocolo habilitado. (GHEIN, 2007)

3.2.1 Configuração

Para habilitar o LDP, basta configurar **mpls label protocol ldp** globalmente e habilitar **mpls ip** globalmente e nas interfaces:

```
PE1(config)#mpls label protocol ldp
PE1(config)#mpls ip
PE1(config)#interface gigabitEthernet 1/0
PE1(config-if)#mpls ip
PE1(config-if)#exit
PE1(config)#interface gigabitEthernet 2/0
PE1(config-if)#mpls ip
```

Figura 19 - Habilitando MPLS IP globalmente e nas interfaces
Fonte: Autoria Própria

A documentação do fabricante também indica ser necessário a configuração do *Cisco Express Forwarding* (CEF) globalmente, através do comando **ip cef** (GHEIN, 2007). Esta funcionalidade já estava configurado por padrão no equipamento utilizado para esta simulação.

```
PE1#show running-config
Building configuration...

<omitido>
```

```
!
ip source-route
no ip icmp rate-limit unreachable
ip cef
!
```

Figura 20 - Running Config do PE1
Fonte: Autoria Própria

O protocolo possui alguns temporizadores configuráveis, como o *backoff* e o *holdtime*. Para a demonstração, estes foram mantidos com os valores padrões, 15 e 120 segundos (inicial e máximo) para o *backoff* e 180 segundos para o *holdtime*. (GHEIN, 2007, p. 73)

3.2.2 Análise e Troubleshooting

É possível verificar os valores dos parâmetros configurados para o LDP através do comando **show mpls ldp parameters**:

```
PE1#show mpls ldp parameters
Protocol version: 1
Session hold time: 180 sec; keep alive interval: 60 sec
Discovery hello: holdtime: 15 sec; interval: 5 sec
Discovery targeted hello: holdtime: 90 sec; interval: 10 sec
Downstream on Demand max hop count: 255
LDP for targeted sessions
LDP initial/maximum backoff: 15/120 sec
LDP loop detection: off
```

Figura 21 - Verificação dos parâmetros do LDP
Fonte: Autoria Própria

Logo após habilitar o LDP nas interfaces, há uma mensagem de log indicando a descoberta de um *neighbor* e o estabelecimento da sessão:

```
PE1(config-if)#mpls ip
*Dec  8 00:02:58.399: %LDP-5-NBRCHG: LDP Neighbor 8.0.0.2:0 (1) is UP
```

Figura 22 - Log de alteração de status de sessão LDP
Fonte: Autoria Própria

Para ver informações sobre as sessões formadas, utiliza-se o comando **show mpls ldp neighbor**:

```
PE1#show mpls ldp neighbor
    Peer LDP Ident: 8.0.0.5:0; Local LDP Ident 8.0.0.1:0
        TCP connection: 8.0.0.5.21380 - 8.0.0.1.646
        State: Oper; Msgs sent/rcvd: 52/50; Downstream
        Up time: 00:33:53
        LDP discovery sources:
          GigabitEthernet2/0, Src IP addr: 14.0.0.17
        Addresses bound to peer LDP Ident:
```

```
          14.0.0.17       8.0.0.5         14.0.0.14
    Peer LDP Ident: 8.0.0.2:0; Local LDP Ident 8.0.0.1:0
        TCP connection: 8.0.0.2.24459 - 8.0.0.1.646
        State: Oper; Msgs sent/rcvd: 20/21; Downstream
        Up time: 00:06:47
        LDP discovery sources:
          GigabitEthernet1/0, Src IP addr: 14.0.0.2
        Addresses bound to peer LDP Ident:
          14.0.0.5        8.0.0.2         14.0.0.2
```

Figura 23 - Informações sobre os vizinhos LDP
Fonte: Autoria Própria

A partir do comando é possível observar que o roteador fez a descoberta dos dois roteadores adjacentes. Nota-se, que o Identificador dos *neighbors* são seus IPs de *loopback*, como garantido pela configuração, porém nas informações de descoberta, o IP do vizinho é o da interface física.

Mais informações sobre o processo de descoberta de vizinhos aparece em **show mpls ldp discovery**:

```
PE1#show mpls ldp discovery
 Local LDP Identifier:
    8.0.0.1:0
    Discovery Sources:
    Interfaces:
        GigabitEthernet1/0 (ldp): xmit/recv
            LDP Id: 8.0.0.2:0
        GigabitEthernet2/0 (ldp): xmit/recv
            LDP Id: 8.0.0.5:0
```

Figura 24 - Informações sobre o método de descoberta do LDP
Fonte: Autoria Própria

Os vizinhos que foram descobertos pelo mecanismo básico do LDP são mostrados no nível *Interfaces*.

Com as sessões formadas, é possível verificar os LSPs formados através do comando **show mpls forwarding**:

```
PE1#show mpls forwarding-table
Local      Outgoing    Prefix          Bytes Label   Outgoing    Next Hop
Label      Label       or Tunnel Id    Switched      interface
16         Pop Label   8.0.0.5/32      0             Gi2/0       14.0.0.17
17         16          8.0.0.4/32      0             Gi2/0       14.0.0.17
18         18          8.0.0.3/32      0             Gi1/0       14.0.0.2
19         Pop Label   8.0.0.2/32      0             Gi1/0       14.0.0.2
20         Pop Label   14.0.0.12/30    0             Gi2/0       14.0.0.17
21         21          14.0.0.8/30     0             Gi1/0       14.0.0.2
           20          14.0.0.8/30     0             Gi2/0       14.0.0.17
22         Pop Label   14.0.0.4/30     0             Gi1/0       14.0.0.2
```

Figura 25 - Forwarding table do PE1 após sessões LDP Basic
Fonte: Autoria Própria

A primeira coisa a notar é que, mesmo a sessão LDP tendo sido formada somente com os roteadores adjacentes (PE2 e PE5), há LSPs formados para todos os outros equipamentos.

Nota-se também que foram formados LSPs para os endereços de *loopback* e endereços de interface.

Para testar o *dataplane* destes LSPs é necessário utilizar o Ping MPLS. Este será descrito na sessão 0.

3.2.3 Fluxo de Pacotes

Como já foi citado, o mecanismo básico de descoberta de vizinhos do LDP é através do envio de mensagens *Hello* através das interfaces habilitadas.

A Figura 26 mostra a captura de três pacotes *Hello Message*:

No.	Time	Source	Destination	Protocol	Length	Info
4	0.349000	14.0.0.1	224.0.0.2	LDP	76	Hello Message
10	4.504000	14.0.0.1	224.0.0.2	LDP	76	Hello Message
11	9.072000	14.0.0.1	224.0.0.2	LDP	76	Hello Message

Figura 26 - Captura de pacotes LDP Basic
Fonte: Autoria Própria

```
Frame 42: 76 bytes on wire (608 bits), 76 bytes captured (608 bits)
Ethernet II, Src: ca:02:19:a4:00:38 (ca:02:19:a4:00:38), Dst: IPv4mcast_00:00:02 (01:00:5e:00:00:02)
Internet Protocol Version 4, Src: 14.0.0.2 (14.0.0.2), Dst: 224.0.0.2 (224.0.0.2)
User Datagram Protocol, Src Port: ldp (646), Dst Port: ldp (646)
Label Distribution Protocol
```

Figura 27 - Composição do pacote LDP Hello Basic
Fonte: Autoria Própria

O pacote é gerado com o IP de origem da interface e com destino 224.0.0.2, que é um endereço *multicast* reservado pela IANA (*Internet Assigned Numbers Authority*) e indica que se destina a todos os roteadores nesta sub-rede. (IETF, 2001) (IANA, 2013)

Como o pacote é enviado para um endereço *multicast*, podendo ser destinado a mais de um roteador, o cabeçalho do LDP tem de ser transportado via UDP. A porta utilizada é a 646, reservada para o protocolo (GHEIN, 2007, p. 70).

O intervalo de envio de pacotes está em torno de 4.5 segundos, quando, supostamente, deveria ser de 5 segundos. Esta diferença pode ser resultado de alguma imprecisão do simulador. No entanto, não afeta o funcionamento.

Quando um roteador que também possui o LDP habilitado detecta a mensagem de um outro roteador, estes passam a se comunicar por *unicast* através de uma sessão TCP entre eles:

No.	Time	Source	Destination	Protocol	Length	Info
53	26.934000	14.0.0.1	224.0.0.2	LDP	76	Hello Message
59	29.588000	14.0.0.2	224.0.0.2	LDP	76	Hello Message
60	29.608000	8.0.0.2	8.0.0.1	TCP	60	55048 > ldp [SYN] Seq=0 Win=4128 Len=0 MSS=536
61	29.650000	8.0.0.1	8.0.0.2	TCP	60	ldp > 55048 [SYN, ACK] Seq=0 Ack=1 Win=4128 Len=0 MSS=536
62	29.691000	8.0.0.2	8.0.0.1	TCP	60	55048 > ldp [ACK] Seq=1 Ack=1 Win=4128 Len=0
63	29.729000	8.0.0.2	8.0.0.1	LDP	100	Initialization Message
64	29.786000	8.0.0.1	8.0.0.2	LDP	108	Initialization Message Keep Alive Message

Figura 28 - Inicialização de sessão LDP entre dois roteadores
Fonte: Autoria Própria

Nota-se que a após a descoberta, estes passam a utilizar o IP de *loopback*. A informação do IP a ser utilizado para a sessão LDP é passado como um dos campos do cabeçalho da mensagem de *Hello*. (IETF, 2001)

Com a sessão TCP criada entre eles, os roteadores trocam mensagens de inicialização e logo em seguida começam a trocar as informações sobre a topologia, para poder formaram os LSPs:

No.	Time	Source	Destination	Protocol	Length	Info
66	29.886000	8.0.0.2	8.0.0.1	LDP	370	Address Message Label Mapping Message Label
67	29.976000	8.0.0.1	8.0.0.2	LDP	342	Address Message Label Mapping Message Label

Figura 29 – Troca de pacotes com as informações de FEC
Fonte: Autoria Própria

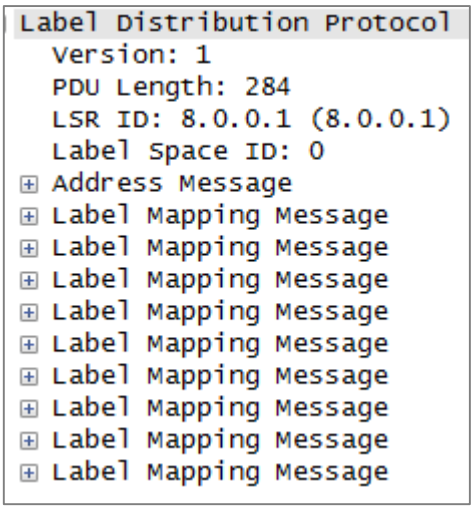

Figura 30 - Header LDP e as diversas Label Mapping Messages
Fonte: Autoria Própria

Nota-se com a Figura 30 que o LDP consegue transmitir diversas informações de maneira bem eficiente. Um único pacote carrega diversos TLVs.

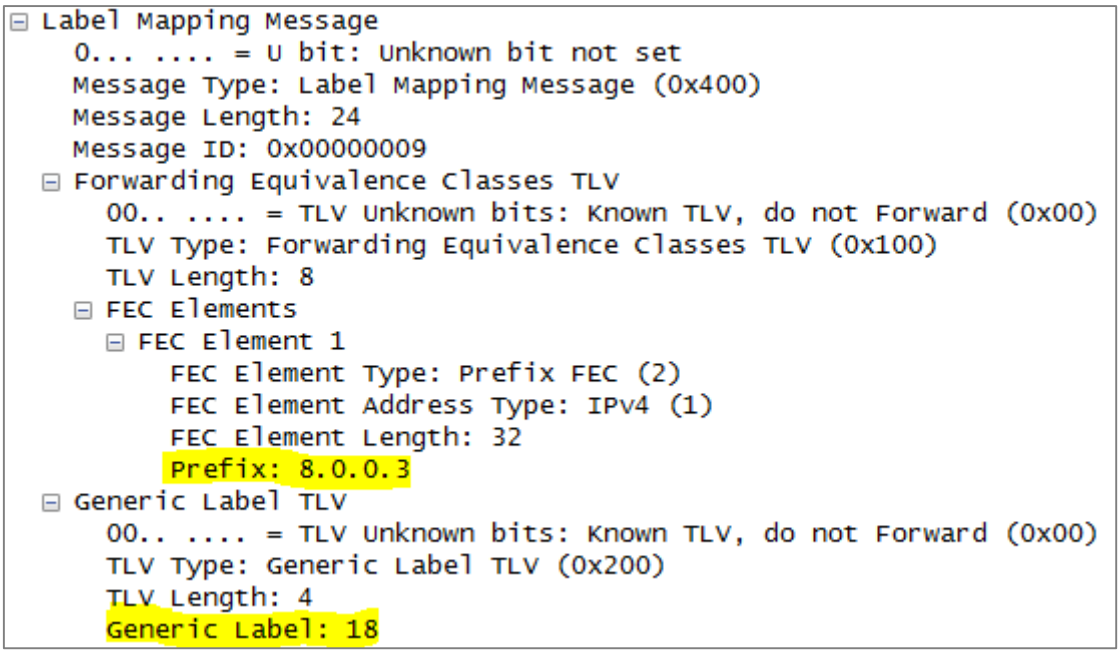

Figura 31 – Label Mapping Message em detalhes
Fonte: Autoria Própria

Analisando o conteúdo do pacote enviado pelo PE1 de maneira minuciosa, é possível observar a troca das informações para formação dos LSPs. Na Figura 31 está o TLV que anuncia o prefixo 8.0.0.3 com o Label 18, como já havia sido visto no **show mpls forwarding-table** do PE2.

3.3 LDP TARGETED

Como foi visto anteriormente, só habilitar o LDP em todos os roteadores já fez com que estes estabelecessem sessões com os equipamentos diretamente conectados. Também notou-se que foram criados LSPs para todos os demais roteadores.

No entanto, em alguns casos, é necessário estabelecer uma sessão com um roteador remoto, ou seja, não diretamente conectado. Este é justamente o caso de redes AToM, já que é necessário haver uma sessão entre cada par de PEs (GHEIN, 2007, p. 84).

Nestes casos, configura-se uma *Targeted Session*, apontando para o IP do roteador.

3.3.1 Configuração

Para configurar uma LDP Targeted Session, basta executar o comando **mpls ldp neighbor <endereço> targeted ldp**:

```
PE1(config)#mpls ldp neighbor 8.0.0.3 targeted ldp
```

Figura 32 - Configuração de LDP Targeted Session entre PE1 e PE3
Fonte: Autoria Própria

Após a configuração, o PE1 passa a tentar descobrir o PE3. Ele faz isto enviando pacotes *Hello* de maneira similar ao LDP *basic*, porém *unicast* com o endereço especificado na configuração:

No.	Time	Source	Destination	Protocol	Length	Info
20	15.682000	8.0.0.1	8.0.0.3	LDP	80	Hello Message
27	20.389000	8.0.0.1	8.0.0.3	LDP	80	Hello Message

Figura 33 - Pacotes Hello Targeted de PE1 para PE3
Fonte: Autoria Própria

A sessão só é completa quando se faz o mesmo comando no PE3, apontando para o PE1:

```
PE3(config)#mpls ldp neighbor 8.0.0.1 targeted ldp
*Dec 14 16:19:28.423: %LDP-5-NBRCHG: LDP Neighbor 8.0.0.1:0 (3) is UP
```

Figura 34 - Estabelecimento de LDP Targeted Sessions entre PE3 e PE1
Fonte: Autoria Própria

Este momento é quando os dois PEs detectam os pacotes de *Hello targeted* e passam a trocar informações do LDP:

No.	Time	Source	Destination	Protocol	Length	Info
22	13.617000	8.0.0.1	8.0.0.3	LDP	80	Hello Message
25	13.744000	8.0.0.3	8.0.0.1	LDP	76	Hello Message
28	13.837000	8.0.0.3	8.0.0.1	LDP	100	Initialization Message
29	13.907000	8.0.0.1	8.0.0.3	LDP	112	Initialization Message Keep Alive Message
30	13.999000	8.0.0.3	8.0.0.1	LDP	72	Keep Alive Message
31	14.019000	8.0.0.3	8.0.0.1	LDP	370	Address Message Label Mapping Message Label
32	14.091000	8.0.0.1	8.0.0.3	LDP	374	Address Message Label Mapping Message Label

Figura 35 – Captura de pacotes da formação da Sessão LDP Targeted PE1 e PE3
Fonte: Autoria Própria

Para determinar qual papel um host LDP terá em uma sessão, é feita uma comparação das IDs de forma que a maior assumira o papel de ativa e a menor de passiva (IETF, 2001, p. 13).

Se o LSR é ativo, este tenta estabelecer a conexão TCP para a sessão LDP. Caso seja passivo, ele simplesmente aguarda pela tentativa de conexão do ativo (IETF, 2001). Na Figura 35 nota-se que quem iniciou a sessão foi o PE3.

A configuração foi repetida fazendo todos os roteadores fazerem sessões com todos os roteadores, formando uma topologia *Full Mesh*.

É importante incluir *targeted sessions* com os roteadores adjacentes, como forma de redundância. Se fossem mantidos apenas as *sessões basic*, caso ocorre queda do link, estes podem deixar de ser adjacentes, podendo interromper serviços de AToM, que necessitam da sessão LDP.

3.3.1.1 LDP Auth

As sessões LDP utilizam sessões TCP como transporte. Sessões TCP podem ser atacadas através de segmentos TCP forjados (GHEIN, 2007, p. 86). Isto tem o potencial de ser um grande problema, pois é possível que um invasor afete toda a infraestrutura de LSPs, podendo causar indisponibilidades.

Uma maneira de proteger uma rede contra este tipo de ataque é utilizando autenticação MD5. Este mecanismo insere uma assinatura no cabeçalho TCP que confirma a identidade do *neighbor* que enviou o pacote (GHEIN, 2007).

Para habilitar a autenticação, basta incluir o comando **mpls ldp neighbor <endereço> password [0|7] <senha>**:

```
PE1(config)#mpls ldp neighbor 8.0.0.3 password target_13
PE1(config)#

PE3(config)#mpls ldp neighbor 8.0.0.1 password target_13
PE3(config)#
```

Figura 36 - Configuração de LDP Authentication em sessão Targeted
Fonte: Autoria Própria

Com os dois *neighbors* configurados com a mesma senha, estes passam a incluir o MD5 *digest* no cabeçalho do TCP:

```
⊞ Ethernet II, Src: ca:01:19:90:00:1c (ca:01:19:90:00:1c), Dst: ca:02:19:90:00:38 (ca:02:19:90:00:38)
⊞ MultiProtocol Label Switching Header, Label: 16, Exp: 6, S: 1, TTL: 255
⊞ Internet Protocol Version 4, Src: 8.0.0.1 (8.0.0.1), Dst: 8.0.0.3 (8.0.0.3)
⊟ Transmission Control Protocol, Src Port: ldp (646), Dst Port: 21179 (21179), Seq: 19, Ack: 19, Len: 18
    Source port: ldp (646)
    Destination port: 21179 (21179)
    [Stream index: 2]
    Sequence number: 19    (relative sequence number)
    [Next sequence number: 37    (relative sequence number)]
    Acknowledgment number: 19    (relative ack number)
    Header length: 40 bytes
  ⊞ Flags: 0x010 (ACK)
    Window size value: 3730
    [Calculated window size: 3730]
    [window size scaling factor: -1 (unknown)]
  ⊞ Checksum: 0x8b7e [validation disabled]
  ⊞ Options: (20 bytes), TCP MD5 signature, End of Option List (EOL)
  ⊞ [SEQ/ACK analysis]
⊞ Label Distribution Protocol
```

Figura 37 - MD5 Digest no cabeçalho TCP

Fonte: Autoria Propria

3.3.2 Análise e Troubleshooting

Como visto antes, para verificar os mecanismo de *discovery* das sessões, utiliza-se o **show mpld ldp discovery**:

```
PE1#show mpls ldp discovery
 Local LDP Identifier:
    8.0.0.1:0
    Discovery Sources:
    Interfaces:
        GigabitEthernet1/0 (ldp): xmit/recv
            LDP Id: 8.0.0.2:0
        GigabitEthernet2/0 (ldp): xmit/recv
            LDP Id: 8.0.0.5:0
    Targeted Hellos:
        8.0.0.1 -> 8.0.0.2 (ldp): active/passive, xmit/recv
            LDP Id: 8.0.0.2:0
        8.0.0.1 -> 8.0.0.3 (ldp): active/passive, xmit/recv
            LDP Id: 8.0.0.3:0
        8.0.0.1 -> 8.0.0.4 (ldp): active/passive, xmit/recv
            LDP Id: 8.0.0.4:0
        8.0.0.1 -> 8.0.0.5 (ldp): active/passive, xmit/recv
```

Figura 38 - Verificação do método de descoberta LDP com sessões Targeted
Fonte: Autoria Própria

Para verificar o estado de todas as sessões criadas utiliza-se o **show mpls ldp neighbor**:

```
PE1#show mpls ldp neighbor
    Peer LDP Ident: 8.0.0.5:0; Local LDP Ident 8.0.0.1:0
        TCP connection: 8.0.0.5.57721 - 8.0.0.1.646
        State: Oper; Msgs sent/rcvd: 28/29; Downstream
        Up time: 00:14:03
        LDP discovery sources:
          GigabitEthernet2/0, Src IP addr: 14.0.0.17
          Targeted Hello 8.0.0.1 -> 8.0.0.5, active, passive
        Addresses bound to peer LDP Ident:
          8.0.0.5         14.0.0.17       14.0.0.14
    Peer LDP Ident: 8.0.0.2:0; Local LDP Ident 8.0.0.1:0
        TCP connection: 8.0.0.2.47636 - 8.0.0.1.646
        State: Oper; Msgs sent/rcvd: 29/29; Downstream
        Up time: 00:14:03
        LDP discovery sources:
          GigabitEthernet1/0, Src IP addr: 14.0.0.2
          Targeted Hello 8.0.0.1 -> 8.0.0.2, active, passive
        Addresses bound to peer LDP Ident:
          8.0.0.2         14.0.0.5        14.0.0.2
    Peer LDP Ident: 8.0.0.3:0; Local LDP Ident 8.0.0.1:0
        TCP connection: 8.0.0.3.21179 - 8.0.0.1.646
        State: Oper; Msgs sent/rcvd: 29/28; Downstream
        Up time: 00:14:01
        LDP discovery sources:
          Targeted Hello 8.0.0.1 -> 8.0.0.3, active, passive
        Addresses bound to peer LDP Ident:
          8.0.0.3         14.0.0.9        14.0.0.6
```

```
    Peer LDP Ident: 8.0.0.4:0; Local LDP Ident 8.0.0.1:0
        TCP connection: 8.0.0.4.19390 - 8.0.0.1.646
        State: Oper; Msgs sent/rcvd: 29/28; Downstream
        Up time: 00:14:00
        LDP discovery sources:
          Targeted Hello 8.0.0.1 -> 8.0.0.4, active, passive
        Addresses bound to peer LDP Ident:
          8.0.0.4         14.0.0.13       14.0.0.10
           LDP Id: 8.0.0.5:0
```

Figura 39 - Informações de todas as sessões LDP em PE1
Fonte: Autoria Própria

É possível também verificar informações de autenticação de um *neighbor* com o comando detalhado:

```
PE1#show mpls ldp neighbor 8.0.0.3 detail
    Peer LDP Ident: 8.0.0.3:0; Local LDP Ident 8.0.0.1:0
        TCP connection: 8.0.0.3.16015 - 8.0.0.1.646; MD5 on
        Password: required, neighbor, in use
        State: Oper; Msgs sent/rcvd: 14/14; Downstream; Last TIB rev
sent 20
        Up time: 00:00:25; UID: 3; Peer Id 2;
        LDP discovery sources:
          Targeted Hello 8.0.0.1 -> 8.0.0.3, active, passive;
            holdtime: infinite, hello interval: 10000 ms
        Addresses bound to peer LDP Ident:
          8.0.0.3         14.0.0.9        14.0.0.6
        Peer holdtime: 180000 ms; KA interval: 60000 ms; Peer state:
estab
        Clients: Dir Adj Client
        Capabilities Sent:
          [Dynamic Announcement (0x0506)]
          [Typed Wildcard (0x0970)]
        Capabilities Received:
          [Dynamic Announcement (0x0506)]
          [Typed Wildcard (0x0970)]
```

Figura 40 - Informações detalhadas de sessão LDP entre PE1 e PE3
Fonte: Autoria Própria

3.4 MPLS OAM

Quando um LSP para de encaminhar tráfego dos usuários, esta falha nem sempre é facilmente detectada somente através do *controlplane*. Por isso, é importante a disponibilidade de ferramentas capazes de detectar e isolar em tempo hábil estas falhas. (IETF, 2006)

Como ainda não há circuitos configurados, não há como passar tráfego comum sobre a infraestrutura MPLS. Estas ferramentas de OAM também permitem realizar uma demonstração de análise do *dataplane* MPLS.

Os roteadores da Cisco fornecem funcionalidades de *ping* e *traceroute* MPLS. (CISCO, 2011)

3.4.1 Análise de um LSP através da Forwarding Table

Antes de realizar a verificação com as ferramentas de OAM, é interessante fazer uma análise da *forwarding-table*, montada através do LDP, de forma a prever quais ações serão tomadas pelos roteadores.

```
PE1#show mpls forwarding-table
Local      Outgoing   Prefix         Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id   Switched      interface
16         Pop Label  8.0.0.5/32     10800         Gi2/0      14.0.0.17
17         16         8.0.0.4/32     0             Gi2/0      14.0.0.17
18         20         8.0.0.3/32     0             Gi1/0      14.0.0.2
19         Pop Label  8.0.0.2/32     10976         Gi1/0      14.0.0.2
20         Pop Label  14.0.0.12/30   0             Gi2/0      14.0.0.17
21         22         14.0.0.8/30    0             Gi1/0      14.0.0.2
           17         14.0.0.8/30    0             Gi2/0      14.0.0.17
22         Pop Label  14.0.0.4/30    0             Gi1/0      14.0.0.2
```

Figura 41 - Verificação da Forwarding Table em PE1
Fonte: Autoria Própria

A entrada destacada diz que um pacote MPLS que chegue com *label* 18, será realizado um swap para 20. Caso seja originado no próprio PE1, será realizado uma operação de *push* com o valor 20 e então este será encaminhado via interface Gi1/0, *next-hop* 14.0.0.2, ou seja, o PE2.

```
PE2#show mpls forwarding-table
Local      Outgoing   Prefix         Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id   Switched      interface
16         Pop Label  8.0.0.1/32     11008         Gi2/0      14.0.0.1
17         Pop Label  14.0.0.16/30   0             Gi2/0      14.0.0.1
18         16         8.0.0.5/32     0             Gi2/0      14.0.0.1
19         17         8.0.0.4/32     0             Gi1/0      14.0.0.6
20         Pop Label  8.0.0.3/32     10272         Gi1/0      14.0.0.6
21         20         14.0.0.12/30   0             Gi2/0      14.0.0.1
           21         14.0.0.12/30   0             Gi1/0      14.0.0.6
22         Pop Label  14.0.0.8/30    0             Gi1/0      14.0.0.6
```

Figura 42 - Verificação da Forwarding Table em PE2
Fonte: Autoria Própria

No PE2, o pacote que chega com *label* 20, possui o mesmo prefixo. Este sofrerá uma ação de *Pop Label* (devido ao P2 realizar o PHP) e será encaminhado via Gi1/0, *next-hop* 14.0.0.6.

```
PE3#show mpls forwarding-table
Local      Outgoing   Prefix         Bytes Label   Outgoing   Next Hop
```

```
Label      Label      or Tunnel Id  Switched      interface
16         16         8.0.0.5/32    0             Gi1/0      14.0.0.10
17         Pop Label  8.0.0.4/32    11336         Gi1/0      14.0.0.10
18         Pop Label  8.0.0.2/32    12700         Gi2/0      14.0.0.5
19         16         8.0.0.1/32    0             Gi2/0      14.0.0.5
20         17         14.0.0.16/30  0             Gi2/0      14.0.0.5
           17         14.0.0.16/30  0             Gi1/0      14.0.0.10
21         Pop Label  14.0.0.12/30  0             Gi1/0      14.0.0.10
22         Pop Label  14.0.0.0/30   0             Gi2/0      14.0.0.5
```

Figura 43 - Verificação da Forwarding Table em PE3
Fonte: Autoria Própria

No PE3, há um LSP no sentido contrário. Caso este origine um pacote ou receba um pacote com *label* 19, este será encaminhado via Gi2/0, *next-hop* 14.0.0.5, com *label* 16.

No PE2, novamente, a tabela indica que recebendo um pacote com *label* 16, é removido o rótulo e o pacote é redirecionado via Gi2/0, *next-hop* 14.0.0.1.

3.4.2 Ping MPLS

O MPLS LSP *ping* utiliza pacotes de MPLS *Echo Request* e *Reply* para validar um LSP. Esta funciona de maneira análoga ao ICMP para redes IP. (CISCO, 2011)

Para executar este comando, procura-se um prefixo incluso na *forwarding table* e executa-se o comando **ping mpls ipv4 <prefixo>**:

```
PE1#ping mpls ipv4 8.0.0.3/32
Sending 5, 100-byte MPLS Echos to 8.0.0.3/32,
     timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label
entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 72/92/116
ms
```

Figura 44 - Ping MPLS sobre o LSP entre PE1 e PE3
Fonte: Autoria Própria

Via CLI, nota-se que houve sucesso na verificação do *dataplane* do LSP entre PE1 e PE3.

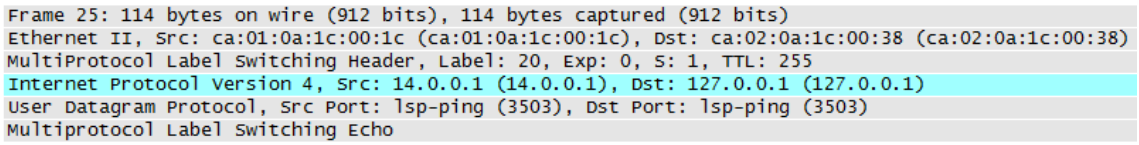

Figura 45 - Composição do pacote MPLS Echo Request entre PE1 e PE2
Fonte: Autoria Própria

O pacote destacado na Figura 45 foi enviado no sentido PE1 para PE2. Como havia sido previsto, o *label* MPLS para este LSP é o 20.

Nota-se também que o conteúdo encapsulado possui uma camada IPv4 com destino 127.0.0.1. O padrão específica a utilização deste endereço como uma maneira de minimizar a possibilidade de um pacote ser entregue ao usuário final ou que este seja reenviado à algum outro destino, caso o LSP esteja com falha. (IETF, 2006)

Em cima disto, há um cabeçalho UDP com portas 3503, porta bem conhecida reservada para este serviço (IETF, 2006), e por fim o cabeçalho MPLS Echo.

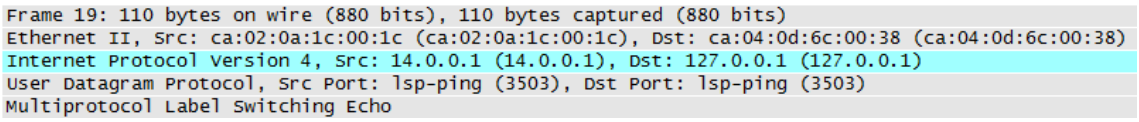

Figura 46 - Composição do pacote MPLS Echo Request entre PE2 e PE3
Fonte: Autoria Própria

Já na Figura 46, o pacote destacado foi capturado no sentido PE2 para o PE3. Mais uma vez como previsto na análise, este sofreu uma ação de *pop* pelo PE2 e não possui *label* MPLS.

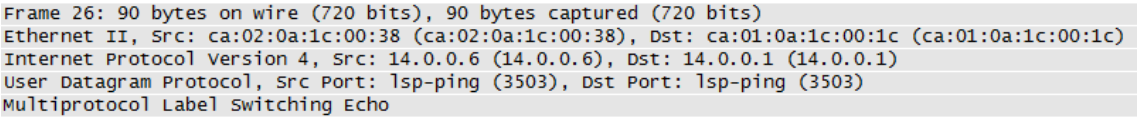

Figura 47 - Composição do pacote MPLS Echo Reply
Fonte: Autoria Própria

Finalmente, após a mensagem de MPLS Echo Request chegar ao PE3, este responde ao PE com um MPLS Echo Reply, ilustrado na Figura 47.

O endereço IP e porta UDP de destino deste pacote são copiados do pacote de Echo Request. (IETF, 2006)

Nota-se que o pacote não possui *labels* MPLS. A resposta é feita de maneira *unicast* utilizando a estrutura de roteamento IP.

Isto se deve à natureza assimétrica dos LSPs. Se o pacote de retorno também fosse comutado através da infraestrutura MPLS, o *ping* estaria testando mais de um LSP simultaneamente.

Por este motivo, também é importante verificar o *ping* no sentido contrário, ou seja, do PE3 ao PE1.

3.4.3 Traceroute MPLS

A ferramenta *ping* MPLS permite detectar possíveis falhas, no entanto, ele não serve para indicar com precisão aonde está a falha, principalmente em casos em que hajam muitos saltos entre os dois roteadores de ponta.

Por isto, há a funcionalidade de *traceroute* MPLS.

```
PE1#traceroute mpls ipv4 8.0.0.3/32
Tracing MPLS Label Switched Path to 8.0.0.3/32, timeout is 2 seconds

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label
entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
  0 14.0.0.1 MRU 1500 [Labels: 20 Exp: 0]
L 1 14.0.0.2 MRU 1504 [Labels: implicit-null Exp: 0] 120 ms
! 2 14.0.0.6 116 ms
```

Figura 48 - Traceroute MPLS sobre LSP entre PE1 e PE3
Fonte: Autoria Própria

```
PE3#traceroute mpls ipv4 8.0.0.1/32
Tracing MPLS Label Switched Path to 8.0.0.1/32, timeout is 2 seconds

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label
entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
  0 14.0.0.6 MRU 1500 [Labels: 16 Exp: 0]
L 1 14.0.0.5 MRU 1504 [Labels: implicit-null Exp: 0] 116 ms
! 2 14.0.0.1 92 ms
```

Figura 49 - Traceroute MPLS sobre LSP entre PE3 e PE1
Fonte: Autoria Própria

Como é possível ver na saída do comando, o *traceroute* faz uma varredura, detalhando todos os saltos realizados e as *labels* utilizadas. Os trechos destacados mostram que os rótulos indicados condizem com os da análise.

O *traceroute* MPLS utiliza os mesmos pacotes de ECHO *Request* e *Reply* que o *ping*. No entanto, os pacotes de *Request* vão sendo gerados com TTL incremental de forma a serem destinados ao *controlplane* de todos os roteadores intermediários. A cada passo, são feitas várias verificações se o roteador realmente faz parte do LSP em questão.

A Figura 50 e Figura 51 mostram o dois pacotes MPLS *Echo Request* consecutivos capturados na interface entre PE1 e PE2.

```
Frame 51: 146 bytes on wire (1168 bits), 146 bytes captured (1168 bits)
Ethernet II, Src: ca:01:0a:1c:00:1c (ca:01:0a:1c:00:1c), Dst: ca:02:0a:1c:00:38 (ca:02:0a:1c:00:38)
MultiProtocol Label Switching Header, Label: 20, Exp: 0, S: 1, TTL: 1
Internet Protocol Version 4, Src: 14.0.0.1 (14.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
User Datagram Protocol, Src Port: lsp-ping (3503), Dst Port: lsp-ping (3503)
Multiprotocol Label Switching Echo
```

Figura 50 - Composição do pacote Traceroute - TTL=1
Fonte: Autoria Própria

```
Frame 53: 146 bytes on wire (1168 bits), 146 bytes captured (1168 bits)
Ethernet II, Src: ca:01:0a:1c:00:1c (ca:01:0a:1c:00:1c), Dst: ca:02:0a:1c:00:38 (ca:02:0a:1c:00:38)
MultiProtocol Label Switching Header, Label: 20, Exp: 0, S: 1, TTL: 2
Internet Protocol Version 4, Src: 14.0.0.1 (14.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
User Datagram Protocol, Src Port: lsp-ping (3503), Dst Port: lsp-ping (3503)
Multiprotocol Label Switching Echo
```

Figura 51 - Composição do pacote Traceroute - TTL = 2
Fonte: Autoria Própria

3.5 EoMPLS (L2VPN)

Com o estabelecimento das sessões LDP e criação e verificação dos LSPs, o cenário então possui uma infraestrutura MPLS formada.

No entanto, só a infraestrutura não serve para atender o tráfego dos clientes da rede. É preciso configurar os circuitos virtuais que irão encapsular o tráfego L2 e então utilizar a infraestrutura para poder encaminhar este tráfego até o destino.

Para testar esta funcionalidade, o cenário foi estendido, conforme a Figura 52.

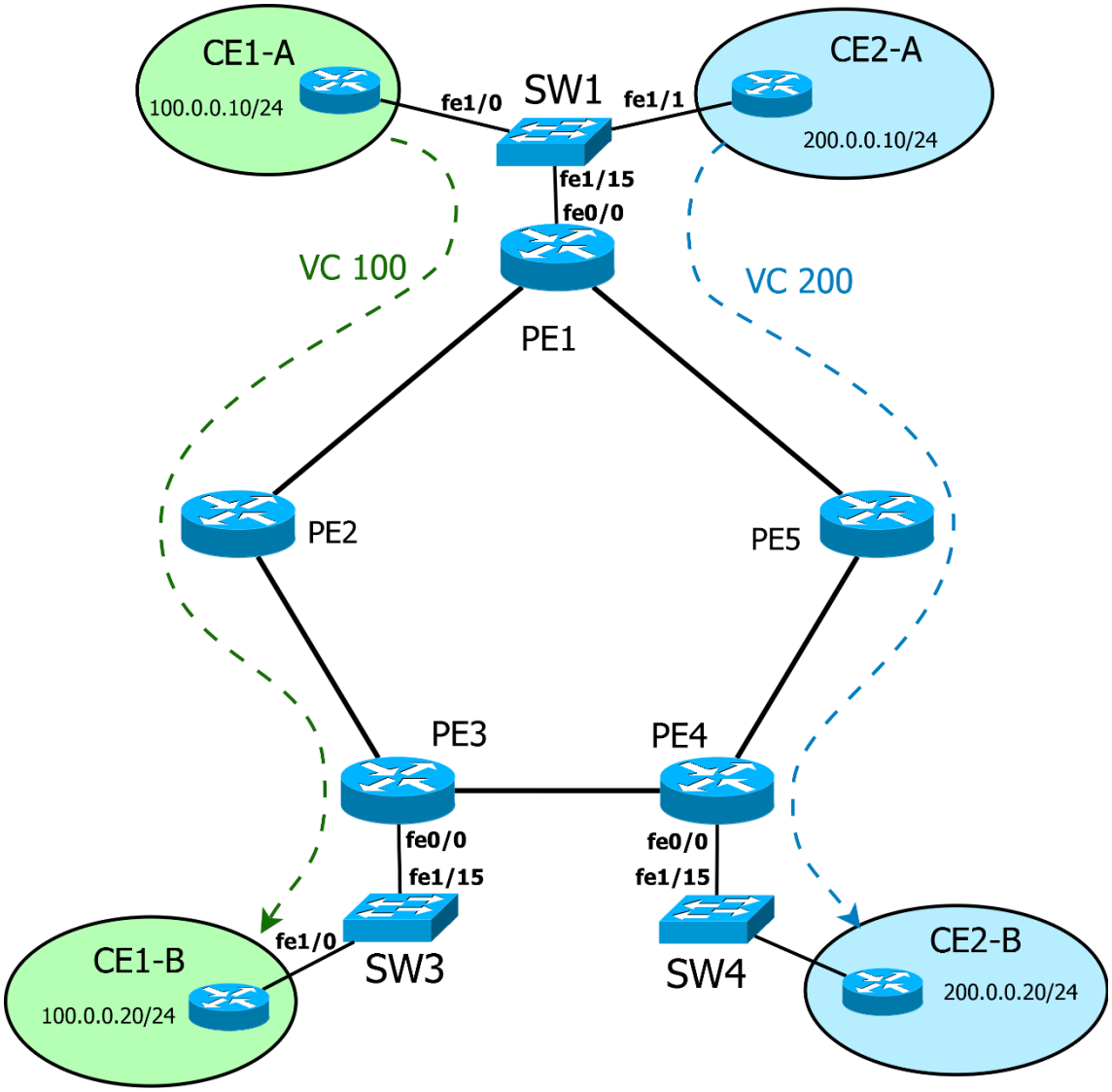

Figura 52 - Cenário final com os clientes
Fonte: Autoria Própria

Foram inclusos *switchs* conectados à fe0/0 dos roteadores PE1, PE3 e PE4. Os *switchs* ainda são considerados domínio do provedor, estando sob a mesma gerência dos roteadores.

O objetivo é conectar os domínios A e B dos dois clientes criando um circuito virtual. Para os roteadores CEs, é como se estes estivessem diretamente conectados, sendo toda a topologia do provedor transparente a eles.

Para simular os *switchs* são utilizados equipamentos Cisco da linha 3600 com imagem c3660-jk9o3s-mz[1].124-17.bin. Este equipamento é um roteador, porém é possível utilizar algumas funcionalidades de *switching* através da placa de extensão NC-ESW16 (CISCO, 2005).

A função dos *switchs* nesta topologia é agregar mais de um cliente em uma mesma interface dos roteadores PE, separando estes em VLANs distintas.

Nos roteadores MPLS, será criado um circuito virtual (VC) com ID 100 para atender o cliente 1 e um VC com ID 200 para o cliente 2.

3.5.1 Configuração

Inicialmente, é necessário configurar os switches. No SW1, são criadas as VLANs 100 e 200 e as interfaces fe1/0 e fe1/1 são configuradas como interfaces de acesso, atendendo os clientes CE1 e CE2, respectivamente.

```
SW1#vlan database
SW1(vlan)#vlan 10
VLAN 100 added:
    Name: VLAN0100
SW1(vlan)#vlan 20
VLAN 200 added:
    Name: VLAN0200
SW1(vlan)#exit
SW1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SW1(config)#interface fastEthernet 1/0
SW1(config-if)#switchport access vlan 100
SW1(config-if)#exit
SW1(config)#interface fastEthernet 1/1
SW1(config-if)#switchport access vlan 200
```

Figura 53 - Configuração de VLANs em SW1
Fonte: Autoria Própria

A interface fe1/15 é configurada como uma interface *trunk* dot1q:

```
SW1(config)#interface fastEthernet 1/15
SW1(config-if)#switchport mode trunk
SW1(config-if)#switchport trunk encapsulation dot1q
```

Figura 54 - Configuração de porta trunk em SW1
Fonte: Autoria Própria

As configurações em SW3 e SW4 são similares. Em SW3, é criada apenas a VLAN 100, a porta fe1/0 configurada como acesso e a fe1/15 como *trunk*. Em SW4, a VLAN criada é a 200, com a porta 1/0 de acesso e a fe1/15 como *trunk*.

Para receber o tráfego das VLANs, é necessário criar sub interfaces nos PEs.

```
PE1(config)#interface fastEthernet 0/0.100
PE1(config-subif)#encapsulation dot1Q 100
PE1(config)#interface fastEthernet 0/0.200
```

```
PE1(config-subif)#encapsulation dot1Q 200
```

Figura 55 - Configuração de sub interfaces em PE1
Fonte: Autoria Própria

```
PE3(config)#interface fastEthernet 0/0.100
PE3(config-subif)#encapsulation dot1Q 100
```

Figura 56 - Configuração de sub interfaces em PE3
Fonte: Autoria Própria

```
PE4(config)#interface fastEthernet 0/0.200
PE4(config-subif)#encapsulation dot1Q 200
```

Figura 57 - Configuração de sub interfaces em PE4
Fonte: Autoria Própria

Para criar os circuitos virtuais, utiliza-se o comando **xconnect <endereço> <vc id> encapsulation mpls**:

```
PE1(config)#interface fastEthernet 0/0.100
PE1(config-subif)#xconnect 8.0.0.3 100 encapsulation mpls
PE1(config-subif-xconn)#exit
PE1(config-subif)#exit
PE1(config)#interface fastEthernet 0/0.200
PE1(config-subif)#xconnect 8.0.0.4 200 encapsulation mpls
PE1(config-subif-xconn)#exit
```

Figura 58 - Criação de VC através de xconnect nas sub interfaces de PE1
Fonte: Autoria Própria

```
PE3(config)#interface fastEthernet 0/0.100
PE3(config-subif)#xconnect 8.0.0.1 100 encapsulation mpls
PE3(config-subif-xconn)#exit
```

Figura 59 - Criação de VC através de xconnect na sub interface de PE3
Fonte: Autoria Própria

```
PE4(config)#interface fastEthernet 0/0.200
PE4(config-subif)#xconnect 8.0.0.1 200 encapsulation mpls
PE4(config-subif-xconn)#exit
```

Figura 60 - Criação de VC através de xconnect na sub interface de PE4
Fonte: Autoria Própria

A partir de então, os domínios dos clientes devem estar conectados.

```
CE1-A#ping 100.0.0.20

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 100.0.0.20, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/84/120
ms
```

Figura 61 - Verificação de conectividade entre roteadores CE1-A e CE1-B

Fonte: Autoria Própria

```
CE2-A#ping 200.0.0.20

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 200.0.0.20, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 60/75/88
ms
```

Figura 62 - Verificação de conectividade entre roteadores CE2-A e CE2-B
Fonte: Autoria Própria

3.5.2 Análise

A verificação do estado dos circuitos virtuais é feita através do comando **show mpls l2transport vc** nos PEs:

```
PE1#show mpls l2transport vc

Local intf     Local circuit     Dest address    VC ID      Status
-------------  ----------------- --------------- ---------- ----------
Fa0/0.100      Eth VLAN 100      8.0.0.3         100        UP
Fa0/0.200      Eth VLAN 200      8.0.0.4         200        UP
```

Figura 63 - Verificação do estado dos circuitos virtuais em PE1
Fonte: Autoria Própria

```
PE3#show mpls l2transport vc

Local intf     Local circuit     Dest address    VC ID      Status
-------------  ----------------- --------------- ---------- ----------
Fa0/0.100      Eth VLAN 100      8.0.0.1         100        UP
```

Figura 64 - Verificação do estado dos circuitos virtuais em PE3
Fonte: Autoria Própria

```
PE4#show mpls l2transport vc

Local intf     Local circuit     Dest address    VC ID      Status
-------------  ----------------- --------------- ---------- ----------
Fa0/0.200      Eth VLAN 200      8.0.0.1         200        UP
```

Figura 65 - Verificação do estado dos circuitos virtuais em PE4
Fonte: Autoria Própria

É possível obter mais detalhes ou especificar um único VC ID no comando:

```
PE1#show mpls l2transport vc 100 detail
Local interface: Fa0/0.100 up, line protocol up, Eth VLAN 100 up
  Destination address: 8.0.0.3, VC ID: 100, VC status: up
    Output interface: Gi1/0, imposed label stack {18 16}
    Preferred path: not configured
    Default path: active
```

```
    Next hop: 14.0.0.2
  Create time: 00:32:37, last status change time: 00:32:15
  Signaling protocol: LDP, peer 8.0.0.3:0 up
    Targeted Hello: 8.0.0.1(LDP Id) -> 8.0.0.3
    Status TLV support (local/remote)   : enabled/supported
      Label/status state machine        : established, LruRru
      Last local dataplane   status rcvd: no fault
      Last local SSS circuit status rcvd: no fault
      Last local SSS circuit status sent: no fault
      Last local  LDP TLV    status sent: no fault
      Last remote LDP TLV    status rcvd: no fault
    MPLS VC labels: local 17, remote 16
    Group ID: local 0, remote 0
    MTU: local 1500, remote 1500
    Remote interface description:
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 0, send 0
    byte totals:   receive 0, send 0
    packet drops:  receive 0, seq error 0, send 0
```

Figura 66 - Informações detalhadas do circuito virtual 100
Fonte: Autoria Própria

Este comando reforça a utilização do LDP como protocolo de sinalização. O principal papel dele é anunciar as *labels* MPLS associadas ao *pseudowire* (GHEIN, 2007, p. 388). Por este motivo é que é necessário manter sempre uma sessão *Targeted* entre os dois PEs.

Nota-se que, para o VC 100, foram distribuídas a *label* 17 localmente e a 16 remotamente.

Para ver mais detalhes sobre as *labels* associadas aos VCs, utiliza-se o comando **show mpls l2transport binding**:

```
PE1#show mpls l2transport bind
  Destination Address: 8.0.0.3,  VC ID: 100
    Local Label:  17
        Cbit: 1,    VC Type: Eth VLAN,    GroupID: 0
        MTU: 1500,   Interface Desc: n/a
        VCCV: CC Type: CW [1], RA [2]
              CV Type: LSPV [2]
    Remote Label: 16
        Cbit: 1,    VC Type: Eth VLAN,    GroupID: 0
        MTU: 1500,   Interface Desc: n/a
        VCCV: CC Type: CW [1], RA [2]
              CV Type: LSPV [2]
  Destination Address: 8.0.0.4,  VC ID: 200
    Local Label:  16
        Cbit: 1,    VC Type: Eth VLAN,    GroupID: 0
        MTU: 1500,   Interface Desc: n/a
        VCCV: CC Type: CW [1], RA [2]
              CV Type: LSPV [2]
    Remote Label: 16
        Cbit: 1,    VC Type: Eth VLAN,    GroupID: 0
        MTU: 1500,   Interface Desc: n/a
        VCCV: CC Type: CW [1], RA [2]
```

```
                CV Type: LSPV [2]
```

Figura 67 - Verificação das associações de label com os circuitos virtuais em PE1
Fonte: Autoria Própria

É interessante reparar as novas entradas na *forwarding table*:

```
PE1#show mpls forwarding-table
Local      Outgoing    Prefix          Bytes Label  Outgoing    Next Hop
Label      Label       or Tunnel Id    Switched     interface
16         No Label    l2ckt(200)      1797         Fa0/0.200   point2point
17         No Label    l2ckt(100)      231451       Fa0/0.100   point2point
18         Pop Label   8.0.0.5/32      79288        Gi2/0       14.0.0.17
19         16          8.0.0.4/32      0            Gi2/0       14.0.0.17
20         18          8.0.0.3/32      0            Gi1/0       14.0.0.2
21         Pop Label   8.0.0.2/32      76           Gi1/0       14.0.0.2
22         Pop Label   14.0.0.12/30    0            Gi2/0       14.0.0.17
23         22          14.0.0.8/30     0            Gi1/0       14.0.0.2
           17          14.0.0.8/30     0            Gi2/0       14.0.0.17
24         Pop Label   14.0.0.4/30     0            Gi1/0       14.0.0.2
```

Figura 68 - Forwarding Table MPLS de PE1 após configuração dos VCs
Fonte: Autoria Própria

3.5.3 Captura de Pacotes

Assim que é configurado o *xconnect* em uma das *interfaces* de um dos PEs, este envia uma LDP *Label Mapping Message* para o destino:

No.	Time	Source	Destination	Protocol	Length	Info
72	42.867000	8.0.0.1	8.0.0.3	LDP	136	Label Mapping Message
122	64.718000	8.0.0.3	8.0.0.1	LDP	132	Label Mapping Message

Figura 69 - Troca de Label Mapping Messages para formação de VC.
Fonte: Autoria Própria

Nesta mensagem LDP, o PE envia um TLV de FEC, indicando o respectivo VC ID e qual o rótulo que será associado à ele, como está destacado na Figura 70.

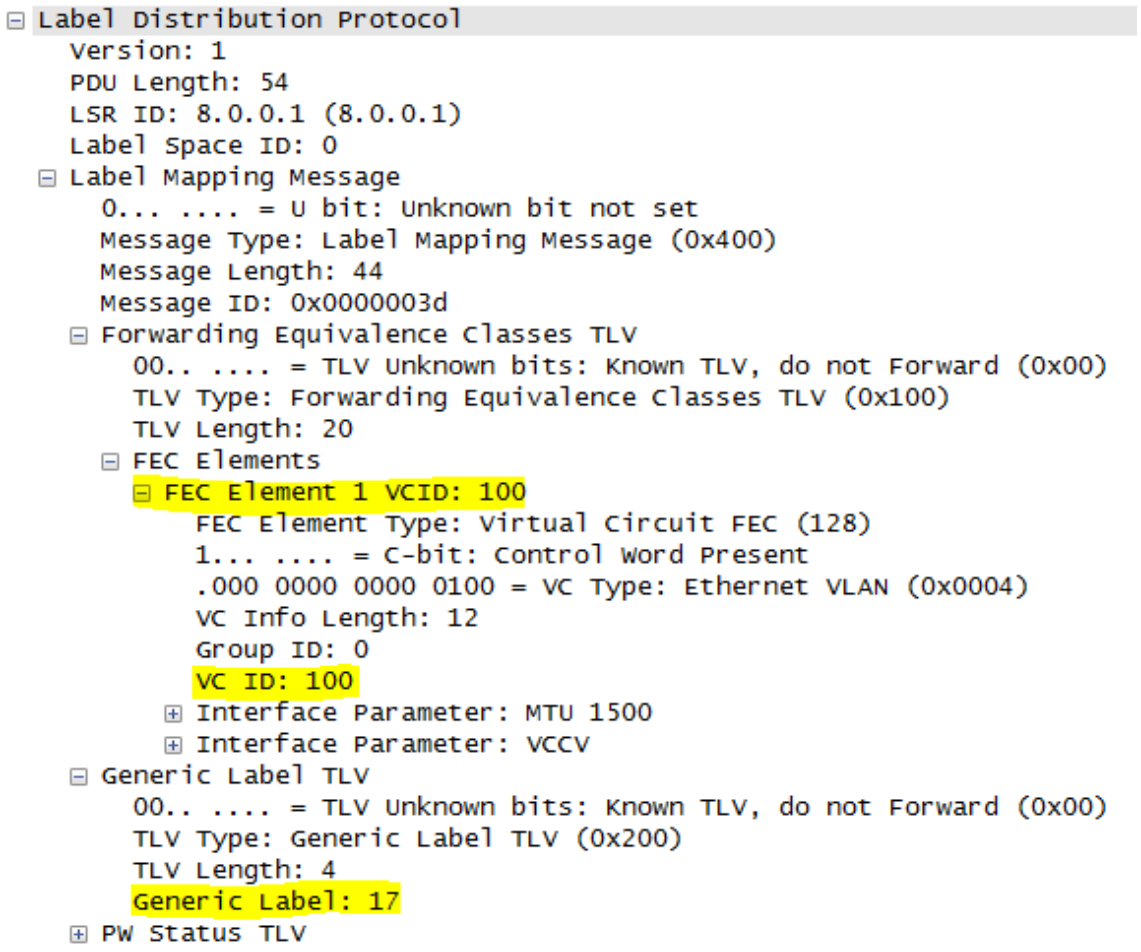

Figura 70 - Informações trocadas no Label Mapping Message para formação de VC.
Fonte: Autoria Própria

Como visto anteriormente na Figura 67, o *label* local do PE1 para o VC 100 é 17, justamente como anunciado pelo LDP.

Após a formação dos VCs, foi verificado a conectividade entre os roteadores CE. A Figura 71 e a Figura 72 mostram o encapsulamento dos cabeçalhos do pacote de *ICMP Echo Request* enviado pelo CE1-A:

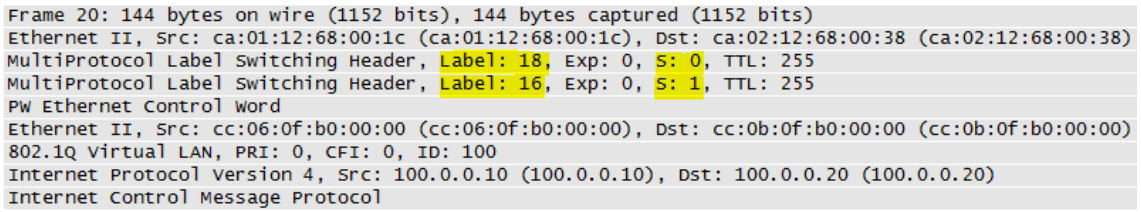

Figura 71 - Cabeçalho do tráfego do CE1 encapsulado pelo VC 100, entre PE1 e PE2
Fonte: Autoria Própria

A primeira coisa a se notar é que há 2 rótulos MPLS empilhados. O rótulo 18 segue os LSPs formados com as sessões LDP. Já o rótulo 16 serve para indicar ao PE3 que o tráfego pertence ao VC 100.

Nota-se que o bit se *label stack* está setado para o rótulo que indica o VC, já que este é o *Bottom of Stack*, ou seja o rótulo mais interno.

```
Frame 16: 140 bytes on wire (1120 bits), 140 bytes captured (1120 bits)
Ethernet II, Src: ca:02:12:68:00:1c (ca:02:12:68:00:1c), Dst: ca:04:13:08:00:38 (ca:04:13:08:00:38)
MultiProtocol Label Switching Header, Label: 16, Exp: 0, S: 1, TTL: 254
PW Ethernet Control Word
Ethernet II, Src: cc:06:0f:b0:00:00 (cc:06:0f:b0:00:00), Dst: cc:0b:0f:b0:00:00 (cc:0b:0f:b0:00:00)
802.1Q Virtual LAN, PRI: 0, CFI: 0, ID: 100
Internet Protocol Version 4, Src: 100.0.0.10 (100.0.0.10), Dst: 100.0.0.20 (100.0.0.20)
Internet Control Message Protocol
```

Figura 72 - Cabeçalho do tráfego do CE1 encapsulado pelo VC 100, entre PE2 e PE3
Fonte: Autoria Própria

No LSP formado entre o PE1 e PE3, o PE2 faz papel de PHP, por isso o pacote é entregue ao PE3 somente com o rótulo que identifica o VC.

A partir de então, o PE3 remove o rótulo 16 e o pacote é enviado através da interface fe0/0, exatamente da mesma maneira que entrou no PE1.

```
Frame 15: 118 bytes on wire (944 bits), 118 bytes captured (944 bits)
Ethernet II, Src: cc:06:0f:b0:00:00 (cc:06:0f:b0:00:00), Dst: cc:0b:0f:b0:00:00 (cc:0b:0f:b0:00:00)
802.1Q Virtual LAN, PRI: 0, CFI: 0, ID: 100
Internet Protocol Version 4, Src: 100.0.0.10 (100.0.0.10), Dst: 100.0.0.20 (100.0.0.20)
Internet Control Message Protocol
```

Figura 73 - Cabeçalho do tráfego do CE1 sem rótulos, entre o PE3 e SW3
Fonte: Autoria Própria

4 CONSIDERAÇÕES FINAIS

O referencial teórico e o conteúdo da simulação introduziu os conceitos básicos e um precedente de como criar uma rede MPLS utilizando um serviço de L2VPNs ponto-a-ponto para fazer a conexão de *sites* remotos de um mesmo cliente.

Na prática, uma rede será muito mais complexa, contendo uma grande quantidade de clientes situados em localidades distintas, além da possibilidade de fornecimento de mais de um tipo de VPN como L3VPNs ou VPLSs.

É cada vez mais comum também fornecer serviços de vídeo e voz integrados à rede de dados. Estes são serviços mais sensíveis à latência, *jitter* e perda de pacotes e em geral necessitam a aplicação de conceitos de QoS (TENENBAUM, 2003, p. 307).

Um outro conceito importante que é comum estar relacionado com o MPLS é o Traffic Engineering, que confere ao operador um controle muito maior sobre os recursos de largura de banda, carga dos links e direcionamento e distribuição do tráfego ao longo de toda a topologia (GHEIN, 2007, p. 249).

Não estava no escopo do trabalho tratar de todos estes aspectos mais avançados, porém este já serviu para familiarização dos termos mais frequentes, da configuração mínima necessária, das informações disponíveis nos equipamentos e de como realizar uma breve análise do estado dos serviços e de funcionamento da topologia.

REFERÊNCIAS

CISCO. Configuring Logical Interfaces, 2001. Disponivel em: <http://www.cisco.com/en/US/docs/ios/12_2/interface/configuration/guide/icflogin.html>. Acesso em: dez. 2012.

CISCO. Cisco 7200 Series Enterprise WAN Aggregation Application, 23 jan. 2003. Disponivel em: <http://www.cisco.com/en/US/products/hw/routers/ps341/prod_brochure09186a0080092124.html>. Acesso em: 07 dez. 2013.

CISCO. Cisco EtherSwitch Modules for the Integrated Services Routers. **Cisco**, 27 dez. 2005. Disponivel em: <http://www.cisco.com/c/en/us/products/collateral/routers/2600-series-multiservice-platforms/product_data_sheet09186a00801aca3e.html>. Acesso em: dez. 2013.

CISCO. Configuring MPLS OAM, 09 out. 2011. Disponivel em: <http://www.cisco.com/en/US/docs/wireless/asr_901/Configuration/Guide/mplsoam.html>. Acesso em: dez. 2013.

CISCO. Cisco 7200 Series Routers, 05 maio 2012. Disponivel em: <http://www.cisco.com/en/US/products/hw/routers/ps341/index.html>. Acesso em: 07 dez. 2013.

CISCO. Cisco Gigabit Ethernet Port Adapter. **Cisco**, 2012. Disponivel em: <http://www.cisco.com/en/US/products/hw/modules/ps2033/ps2595/index.html>. Acesso em: dez. 2012.

GHEIN, L. D. **MPLS Fundamentals**. Indianapolis: Cisco Press, 2007.

GNS3. Graphical Network Simulator - GNS3. **Graphical Network Simulator - GNS3**, 2007. Disponivel em: <http://www.gns3.net/>. Acesso em: dez. 2013.

GUICHARD, J.; APCAR, J.; PEPELNJAK, I. **MPLS and VPN Architectures**. Indianopolis: Cisco Press, v. II, 2003.

HALABI, S. **OSPF Design Guide**. [S.l.]: [s.n.], 1996.

IANA. IPv4 Multicast Address Space Registry, 2013. Disponivel em: <http://www.iana.org/assignments/multicast-addresses/multicast-addresses.xhtml>. Acesso em: dez. 2013.

IETF. **RFC 3036 - LDP Specification**, jan. 2001. Disponivel em: <https://tools.ietf.org/html/rfc3036>. Acesso em: dez. 2013.

IETF. **RFC 3031 - Multiprotocol Label Switching Architecture**, 2001. Disponivel em: <https://tools.ietf.org/html/rfc3031>. Acesso em: dez. 2013.

IETF. **RFC 4379 - Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures**, 2006. Disponivel em: <https://tools.ietf.org/html/rfc4379>. Acesso em: dez. 2013.

MINEI, I.; LUCEK, J. **MPLS-Enabled Applications**. Chichester: John Wiley & Sons, 2005.

PEPELNJAK, I.; GUICHARD, J. MPLS and VPN Architectures. Indianapolis: Cisco Press, v. I, 2001.

SANTOS, R. C. **Um estudo do uso da Tecnologia MPLS em Backbones no Brasil**. Florianópolis: UFSC, 2003.

TENENBAUM, A. S. **Redes de Computadores**. 4ª ed. ed. Rio de Janeiro: Editoda Campus (Elsevier), 2003.

THOMAS, T. M. **OSPF Network Design Solutions**. Indianapolis: Cisco Press, 2003.

APÊNDICE A

PE1)

```
!
upgrade fpd auto
version 15.0
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE1
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
ip source-route
no ip icmp rate-limit unreachable
ip cef
!
no ip domain lookup
no ipv6 cef
!
multilink bundle-name authenticated
mpls ldp password required
mpls ldp neighbor 8.0.0.2 password target_12
mpls ldp neighbor 8.0.0.2 targeted ldp
mpls ldp neighbor 8.0.0.3 password target_13
mpls ldp neighbor 8.0.0.3 targeted ldp
mpls ldp neighbor 8.0.0.4 password target_14
mpls ldp neighbor 8.0.0.4 targeted ldp
mpls ldp neighbor 8.0.0.5 password target_15
mpls ldp neighbor 8.0.0.5 targeted ldp
mpls label protocol ldp
!
redundancy
!
ip tcp synwait-time 5
!
interface Loopback0
 ip address 8.0.0.1 255.255.255.255
 !
!
interface FastEthernet0/0
 no ip address
 duplex full
 no keepalive
 !
!
interface FastEthernet0/0.100
 encapsulation dot1Q 100
 xconnect 8.0.0.3 100 encapsulation mpls
!
interface FastEthernet0/0.200
 encapsulation dot1Q 200
 xconnect 8.0.0.4 200 encapsulation mpls
!
```

```
interface GigabitEthernet1/0
 ip address 14.0.0.1 255.255.255.252
 ip ospf network point-to-point
 negotiation auto
 mpls ip
 !
!
interface GigabitEthernet2/0
 ip address 14.0.0.18 255.255.255.252
 ip ospf network point-to-point
 negotiation auto
 mpls ip
 !
!
router ospf 1
 log-adjacency-changes
 network 8.0.0.1 0.0.0.0 area 0
 network 14.0.0.0 0.0.0.3 area 0
 network 14.0.0.16 0.0.0.3 area 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
control-plane
 !
!
mgcp fax t38 ecm
mgcp behavior g729-variants static-pt
!
gatekeeper
 shutdown
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line vty 0 4
 login
!
end
```

PE2)

```
!
upgrade fpd auto
version 15.0
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE2
!
```

```
boot-start-marker
boot-end-marker
!
no aaa new-model
!
ip source-route
no ip icmp rate-limit unreachable
ip cef
!
no ip domain lookup
no ipv6 cef
!
multilink bundle-name authenticated
mpls ldp password required
mpls ldp neighbor 8.0.0.1 password target_12
mpls ldp neighbor 8.0.0.1 targeted ldp
mpls ldp neighbor 8.0.0.3 password target_23
mpls ldp neighbor 8.0.0.3 targeted ldp
mpls ldp neighbor 8.0.0.4 password target_24
mpls ldp neighbor 8.0.0.4 targeted ldp
mpls ldp neighbor 8.0.0.5 password target_25
mpls ldp neighbor 8.0.0.5 targeted ldp
mpls label protocol ldp
!
redundancy
!
ip tcp synwait-time 5
!
interface Loopback0
 ip address 8.0.0.2 255.255.255.255
 !
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
 !
!
interface GigabitEthernet1/0
 ip address 14.0.0.5 255.255.255.252
 ip ospf network point-to-point
 negotiation auto
 mpls ip
 !
!
interface GigabitEthernet2/0
 ip address 14.0.0.2 255.255.255.252
 ip ospf network point-to-point
 negotiation auto
 mpls ip
 !
!
router ospf 1
 log-adjacency-changes
 network 8.0.0.2 0.0.0.0 area 0
 network 14.0.0.0 0.0.0.3 area 0
 network 14.0.0.4 0.0.0.3 area 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
```

```
!
control-plane
 !
!
mgcp fax t38 ecm
mgcp behavior g729-variants static-pt
!
gatekeeper
 shutdown
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line vty 0 4
 login
!
end
```

PE3)

```
!
upgrade fpd auto
version 15.0
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE3
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
ip source-route
no ip icmp rate-limit unreachable
ip cef
!
no ip domain lookup
no ipv6 cef
!
multilink bundle-name authenticated
mpls ldp password required
mpls ldp neighbor 8.0.0.1 password target_13
mpls ldp neighbor 8.0.0.1 targeted ldp
mpls ldp neighbor 8.0.0.2 password target_23
mpls ldp neighbor 8.0.0.2 targeted ldp
mpls ldp neighbor 8.0.0.4 password target_34
mpls ldp neighbor 8.0.0.4 targeted ldp
mpls ldp neighbor 8.0.0.5 password target_35
mpls ldp neighbor 8.0.0.5 targeted ldp
mpls label protocol ldp
!
```

```
redundancy
!
ip tcp synwait-time 5
!
interface Loopback0
 ip address 8.0.0.3 255.255.255.255
 !
!
interface FastEthernet0/0
 no ip address
 duplex full
 no keepalive
 !
!
interface FastEthernet0/0.100
 encapsulation dot1Q 100
 xconnect 8.0.0.1 100 encapsulation mpls
!
interface GigabitEthernet1/0
 ip address 14.0.0.9 255.255.255.252
 ip ospf network point-to-point
 negotiation auto
 mpls ip
 !
!
interface GigabitEthernet2/0
 ip address 14.0.0.6 255.255.255.252
 ip ospf network point-to-point
 negotiation auto
 mpls ip
 !
!
router ospf 1
 log-adjacency-changes
 network 8.0.0.3 0.0.0.0 area 0
 network 14.0.0.4 0.0.0.3 area 0
 network 14.0.0.8 0.0.0.3 area 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
control-plane
 !
!
mgcp fax t38 ecm
mgcp behavior g729-variants static-pt
!
gatekeeper
 shutdown
!
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
```

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

PE4)

```
!
upgrade fpd auto
version 15.0
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE4
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
ip source-route
no ip icmp rate-limit unreachable
ip cef
!
no ip domain lookup
no ipv6 cef
!
multilink bundle-name authenticated
mpls ldp password required
mpls ldp neighbor 8.0.0.1 password target_14
mpls ldp neighbor 8.0.0.1 targeted ldp
mpls ldp neighbor 8.0.0.2 password target_24
mpls ldp neighbor 8.0.0.2 targeted ldp
mpls ldp neighbor 8.0.0.3 password target_34
mpls ldp neighbor 8.0.0.3 targeted ldp
mpls ldp neighbor 8.0.0.5 password target_45
mpls ldp neighbor 8.0.0.5 targeted ldp
mpls label protocol ldp
!
redundancy
!
ip tcp synwait-time 5
!
interface Loopback1
 ip address 8.0.0.4 255.255.255.255
 !
!
interface FastEthernet0/0
 no ip address
 duplex full
 !
!
interface FastEthernet0/0.200
 encapsulation dot1Q 200
 xconnect 8.0.0.1 200 encapsulation mpls
!
interface GigabitEthernet1/0
 ip address 14.0.0.13 255.255.255.252
```

```
 ip ospf network point-to-point
 negotiation auto
 mpls ip
 !
!
interface GigabitEthernet2/0
 ip address 14.0.0.10 255.255.255.252
 ip ospf network point-to-point
 negotiation auto
 mpls ip
 !
!
router ospf 1
 log-adjacency-changes
 network 8.0.0.4 0.0.0.0 area 0
 network 14.0.0.8 0.0.0.3 area 0
 network 14.0.0.12 0.0.0.3 area 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
control-plane
 !
!
mgcp fax t38 ecm
mgcp behavior g729-variants static-pt
!
gatekeeper
 shutdown
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line vty 0 4
 login
!
end
```

PE5)

```
!
upgrade fpd auto
version 15.0
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE5
!
boot-start-marker
boot-end-marker
!
```

```
no aaa new-model
!
ip source-route
no ip icmp rate-limit unreachable
ip cef
!
no ip domain lookup
no ipv6 cef
!
multilink bundle-name authenticated
mpls ldp password required
mpls ldp neighbor 8.0.0.1 password target_15
mpls ldp neighbor 8.0.0.1 targeted ldp
mpls ldp neighbor 8.0.0.2 password target_25
mpls ldp neighbor 8.0.0.2 targeted ldp
mpls ldp neighbor 8.0.0.3 password target_35
mpls ldp neighbor 8.0.0.3 targeted ldp
mpls ldp neighbor 8.0.0.4 password target_45
mpls ldp neighbor 8.0.0.4 targeted ldp
mpls label protocol ldp
!
redundancy
!
ip tcp synwait-time 5
!
interface Loopback0
 ip address 8.0.0.5 255.255.255.255
 !
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
 !
!
interface GigabitEthernet1/0
 ip address 14.0.0.17 255.255.255.252
 ip ospf network point-to-point
 negotiation auto
 mpls ip
 !
!
interface GigabitEthernet2/0
 ip address 14.0.0.14 255.255.255.252
 ip ospf network point-to-point
 negotiation auto
 mpls ip
 !
!
router ospf 1
 log-adjacency-changes
 network 8.0.0.5 0.0.0.0 area 0
 network 14.0.0.12 0.0.0.3 area 0
 network 14.0.0.16 0.0.0.3 area 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
control-plane
 !
```

```
!
mgcp fax t38 ecm
mgcp behavior g729-variants static-pt
!
gatekeeper
 shutdown
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line vty 0 4
 login
!
end
```

SW1)

```
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname SW1
!
boot-start-marker
boot-end-marker
!
no aaa new-model
memory-size iomem 5
no ip icmp rate-limit unreachable
!
ip cef
no ip domain lookup
!
ip tcp synwait-time 5
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet1/0
 switchport access vlan 100
!
interface FastEthernet1/1
```

```
 switchport access vlan 200
!
interface FastEthernet1/2
!
interface FastEthernet1/3
!
interface FastEthernet1/4
!
interface FastEthernet1/5
!
interface FastEthernet1/6
!
interface FastEthernet1/7
!
interface FastEthernet1/8
!
interface FastEthernet1/9
!
interface FastEthernet1/10
!
interface FastEthernet1/11
!
interface FastEthernet1/12
!
interface FastEthernet1/13
!
interface FastEthernet1/14
!
interface FastEthernet1/15
 switchport mode trunk
!
interface Vlan1
 no ip address
!
interface Vlan100
 no ip address
!
interface Vlan200
 no ip address
!
no ip http server
no ip http secure-server
ip forward-protocol nd
!
control-plane
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
!
end
```

SW3)

```
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname SW3
!
boot-start-marker
boot-end-marker
!
no aaa new-model
memory-size iomem 5
no ip icmp rate-limit unreachable
!
ip cef
no ip domain lookup
!
ip tcp synwait-time 5
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet1/0
 switchport access vlan 100
!
interface FastEthernet1/1
!
interface FastEthernet1/2
!
interface FastEthernet1/3
!
interface FastEthernet1/4
!
interface FastEthernet1/5
!
interface FastEthernet1/6
!
interface FastEthernet1/7
!
interface FastEthernet1/8
!
interface FastEthernet1/9
!
interface FastEthernet1/10
!
interface FastEthernet1/11
!
interface FastEthernet1/12
!
interface FastEthernet1/13
!
```

```
interface FastEthernet1/14
!
interface FastEthernet1/15
 switchport mode trunk
!
interface Vlan1
 no ip address
!
interface Vlan100
 no ip address
!
no ip http server
no ip http secure-server
ip forward-protocol nd
!
control-plane
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
!
!
end
```

SW4)

```
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname SW4
!
boot-start-marker
boot-end-marker
!
no aaa new-model
memory-size iomem 5
no ip icmp rate-limit unreachable
!
ip cef
no ip domain lookup
!
ip tcp synwait-time 5
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet0/1
```

```
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet1/0
 switchport access vlan 200
!
interface FastEthernet1/1
!
interface FastEthernet1/2
!
interface FastEthernet1/3
!
interface FastEthernet1/4
!
interface FastEthernet1/5
!
interface FastEthernet1/6
!
interface FastEthernet1/7
!
interface FastEthernet1/8
!
interface FastEthernet1/9
!
interface FastEthernet1/10
!
interface FastEthernet1/11
!
interface FastEthernet1/12
!
interface FastEthernet1/13
!
interface FastEthernet1/14
!
interface FastEthernet1/15
 switchport mode trunk
!
interface Vlan1
 no ip address
!
interface Vlan200
 no ip address
!
no ip http server
no ip http secure-server
ip forward-protocol nd
!
control-plane
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
```

```
!
end
```