Universidade de Aveiro
2010
Departamento de Electrónica, Telecomunicações e Informática

Nuno Gonçalo da Silva Lourenço

Solução Linux para “backhaul” de redes móveis usando IP/MPLS


Universidade de Aveiro
2010

Departamento de Electrónica, Telecomunicações e Informática

Nuno Gonçalo da Silva Lourenço

Solução Linux para "backhaul" de redes móveis usando IP/MPLS

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

o júri

presidente

Professor Doutor Anibal Manuel de Oliveira Duarte
Professor Catedrático do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

arguente

Professor Doutor Carlos Manuel Da Silva Rabadão
Professor Adjunto do Departamento de Engenharia Informática da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria

orientador

Professor Doutor Amaro Fernandes de Sousa
Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

agradecimentos

As minhas primeiras palavras de agradecimento têm de ir, forçosamente, para a minha família, em especial para os meus pais e para o meu irmão. A paciência, o apoio e o carinho constante demonstrado por eles foram determinantes para o sucesso na elaboração desta dissertação. Apesar de estar já integrado no mercado de trabalho, a conclusão desta fase não deixa de ser um marco importante na minha vida, possível apenas graças à coragem transmitida pela minha família.

Uma palavra de agradecimento à PT Inovação, em especial ao departamento DSR, pela oportunidade e logística fornecida indispensáveis à realização desta dissertação.

Queria também agradecer a todo o grupo DSR2 pelo companheirismo demonstrado, não só durante a realização da dissertação, mas durante toda a minha permanência na PT Inovação. Um agradecimento muito especial ao Pedro Mendes. Sem ele, esta dissertação teria sido impossível. O meu obrigado pelas suas ideias, sugestões e orientações. À Daniela Castro pela revisão do plano de testes e elaboração do procedimento de instalação do software. Ao Sérgio Marques pelas dicas sobre o *quagga.*

Por último, um agradecimento especial ao meu orientador Dr. Prof. Amaro de Sousa pela total disponibilidade, pelas críticas e sugestões. Um obrigado também pela sua total cooperação com a PT Inovação.

palavras-chave

Redes móveis, IP, MPLS, LDP, etiqueta, LSP, RSVP-TE, FRR, VPN, testes.

resumo

Os operadores de redes móveis enfrentam actualmente um problema na procura de largura de banda para o núcleo das suas redes, principalmente devido à proliferação dos serviços de dados 3G, às recentes melhorias nas interfaces rádio de alto débito, como é o caso do HSPA, e às tecnologias emergentes como o LTE. O IP/MPLS torna-se uma solução atractiva para o núcleo destas redes devido à sua flexibilidade, proporcionando redes multi-serviços, escaláveis e mais económicas.

O objectivo principal desta dissertação é a selecção e avaliação de uma solução IP/MPLS para utilização nos equipamentos de núcleo de redes móveis da PT Inovação. Inicialmente é realizada uma pesquisa bibliográfica sobre as diversas tecnologias MPLS, usadas actualmente nas redes móveis.

Para a pesquisa das diversas soluções existentes no mercado, é realizado um levantamento dos requisitos exigidos pelas redes móveis. Estes requisitos têm em conta não só os serviços suportados actualmente pelas redes móveis mas que serão descontinuados no futuro, mas também a evolução prevista para as mesmas. De acordo com estes requisitos, são apresentadas algumas soluções existentes no mercado.

A solução escolhida é então sujeita a um conjunto de testes, previamente definidos e justificados. Estes testes são divididos em testes funcionais, garantindo a conformidade da solução com as normas, e em testes de desempenho, caracterizando-a em termos de robustez e capacidade de funcionamento.

Por fim, a tese conclui sobre as capacidades e as lacunas da solução testada, permitindo avaliar a adequabilidade da sua implementação nos equipamentos da PT Inovação.

keywords

Mobile networks, IP, MPLS, LDP, label, LSP, RSVP-TE, FRR, VPN, tests.

abstract

Mobile operators are facing a significant spike in bandwidth demands in the core of their networks due to the proliferation of 3G-based data services, the recent improvements of high-speed air interface enhancements such as HSPA, and the emergence of new technologies such as LTE. IP/MPLS becomes an attractive solution for the core of these networks due to its flexibility, providing scalable, multi-propose and more cost-effective infrastructures.

The main goal of this thesis is the selection and evaluation of an IP/MPLS solution to use in the core equipments for mobile networks of PT Inovação. Initially, a bibliography research about MPLS technologies, actually used in mobile networks, is realized.

For the software solutions research, it is made a previous survey of the requirements. These requirements take in account not only the services supported actually in mobile networks but that will become legacy, but also the evolution forecast for these networks. According to these requirements, some solutions, available on market, are exposed.

Then, the chosen solution is subject to test procedures, previously defined and justified. These tests are divided in functional tests, ensuring the solution conformity with the standards, and in performance tests, featuring the solution in terms of stability and functional capacity.

Finally, the thesis concludes about the tested solution capabilities and gaps, allowing the evaluation for the solution implementation in PT Inovação equipments.

Índice

Índice

Índice

Índice de figuras

Índice de figuras

Índice de tabelas

Acrónimos

API	*Application Programming Interface*
APS	*Automatic Protection Switching*
ASN	*Autonomous System Number*
ATM	*Asynchronous Transmission Mode*
BGP	*Border Gateway Protocol*
BTS	*Base Transceiver Station*
C	*Costumer*
CDMA2000	*Code Division Multiple Access* 2000
CE	*Costumer Edge*
CR-LDP	*Constraint-based routed Label Distribution Protocol*
CSPF	*Constraint-based Shortest Path First*
CU	*Current Unused*
DiffServ	Diferenciação de Serviços
DLCI	*Data Link Connection Identifier*
DPS	*Data Path Software*
DSCP	*Differentiated Services Codepoint*
DS-TE	*DiffServ with Traffic Engineering*
DUT	*Device Under Test*
EGP	*Exterior Gateway Protocol*
FEC	*Forwarding Equivalency Class*
FIB	*Forwarding Information Base*
FRR	*Fast Reroute*
GPL	*General Public License*
HAL	*Hardware Abstraction Layer*
HDLC	*High-Level Data Link*
HSPA	*High Speed Packet Access*
IANA	*Internet Assigned Numbers Authority*
IETF	*Internet Engineering Task Force*
IntServ	Serviços Integrados
IGP	*Interior Gateway Protocol*
IP	*Internet Protocol*
IS-IS	*Intermediate System-to-Intermediate System*
L2VPN	VPN de camada 2
LAN	*Local Area Network*
LDP	*Label Distribution Protocol*
LER	*Label Edge Router*
LFIB	*Label Forwarding Information Base*
LIB	*Label Information Base*
LSP	*Label Switched Path*
LSR	*Label Switched Router*
LTE	*Long Term Evolution*
MAC	*Media Access Control*
MMBI	*MPLS Mobile Backhaul Initiative*

Acrónimos

MP-BGP *Multi Protocol Border Gateway Protocol*

MPLS *Multiprotocol Label Switching*

MPLS-TE *Multiprotocol Label Switching with Traffic Engineering*

MPLS-TP *MPLS-Transport Profile*

NSM *Network Services Module*

OAM *Operation, Administration and Maintenance*

OSI *Open Systems Interconnection*

OSPF *Open Shortest Path First*

OTN *Optical Transport Network*

P *Provider*

PAL *Plataform Abstraction Layer*

PC Ponto de Convergência

PDH *Plesiochronous Digital Hierarchy*

PDU *Protocol Data Unit*

PE *Provider Edge*

PHP *Penultimate hop Popping*

PPP *Point-to-Point Protocol*

PSN *Packet-Switched Network*

PRL Ponto de Reparação Local

PW *Pseudowire*

QoS *Qualidade de Serviço*

RAN *Radio Access Network*

RIB *Routing Information Base*

RIP *Routing Information Protocol*

RSVP *Resource Reservation Protocol*

RSVP-TE *Resource Reservation Protocol with Traffic Engineering*

SDH *Synchronous Digital Hierarchy*

SPF *Shortest Path First*

TCP *Transmission Control Protocol*

TDM *Time-division multiplexing*

TE *Traffic Engineering*

TLV *Type Length Value*

TOS *Type of Service*

TTL *Time to Live*

UDP *User Datagram Protocol*

UMB *Ultra Mobile Broadband*

VC *Virtual Circuit*

VCCV-Ping *Virtual Circuit Connection Verification-Ping*

VCI *Virtual Circuit Identifier*

VLAN *Virtual LAN*

VP *Virtual Path*

VPI *Virtual Path Identifier*

VPLS *Virtual Private LAN Service*

VPN *Virtual Private Network*

VPWS *Virtual Private Wire Service*

VRF *VPN Routing and Forwarding*

WDM *Wavelength-Division Multiplexing*

Capítulo 1: Introdução

1.1. Motivação

O tráfego de dados nas redes móveis tem crescido nos últimos tempos acima das expectativas. Os analistas, nas suas previsões de mercado, antecipam que os volumes de dados irão crescer a um ritmo acelerado com base em amplos serviços de banda larga. Ao mesmo tempo, as receitas por quantidade de informação continuarão a diminuir, impulsionadas pelo aumento da concorrência e pela introdução de tarifários fixos. Para se manterem competitivos, os operadores de redes móveis devem estar preparados para disponibilizar uma rede que não pode apenas atender ao crescimento da largura de banda, mas também que proporcione uma margem de lucro saudável.

À medida que o acesso aos serviços de banda larga móvel cresce, mais e mais largura de banda será consumida nas redes. Para responder a este desafio, os operadores expandem a capacidade das suas redes implementando novas tecnologias de acesso, tais como o *High Speed Packet Access* (HSPA), o WiMAX e o *Long Term Evolution* (LTE). Este crescimento de largura de banda reflecte-se necessariamente na capacidade de transporte do *backhaul* da rede móvel. A Figura 1.1 mostra a previsão para a capacidade necessária para esse *backhaul*. Segundo [1], além do crescimento da quantidade de *cell sites*, prevê-se também um crescimento significativo da largura de banda necessária por *cell site* devido à introdução destas novas tecnologias. Em 2012, prevê-se a existência de *cell sites* que necessitarão de uma largura de banda de 155Mbps, em contraste com 2007 onde nenhum *cell site* exigia tal capacidade.

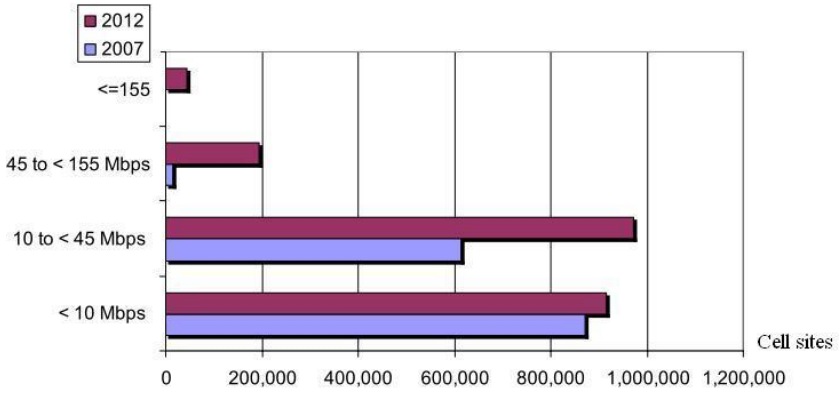


Figura 1.1 – Evolução da capacidade no *backhaul* de redes móveis [1].

Desta forma, os operadores de redes móveis procuram novas soluções para as suas redes. Estas novas soluções, além do crescimento da largura de banda, têm de suportar também as tecnologias já existentes baseadas em *Time-division multiplexing* (TDM). As redes *Internet Protocol / Multiprotocol Label Switching* (IP/MPLS) surgem assim como uma solução atraente por serem redes com arquitecturas mais flexíveis, escaláveis e que permitem reduzir os custos operacionais através da convergência das tecnologias descontinuadas.

A migração destas redes para IP/MPLS terá de ser realizada de um modo seguro e cauteloso. A satisfação dos clientes é primordial e não será tolerada a degradação da qualidade de voz, chamadas perdidas e interrupções. Assim, actualmente, os diversos fornecedores procuram soluções baseadas na tecnologia MPLS, testando diversos tipos de cenários, nos quais os seus equipamentos terão de cumprir os novos desafios em termos de atraso, variação do atraso, qualidade de serviço (QoS), segurança e controlo de congestionamento.

A PT Inovação, sendo uma empresa fornecedora de soluções para as redes de acesso e transporte móvel, procura actualmente uma solução MPLS para utilização nos seus equipamentos. Esta solução terá de garantir todos os requisitos mínimos para aplicação em redes móveis. Além disso, são necessários planos de teste bem delineados, que permitam que a solução encontrada possa satisfazer todas as exigências de uma rede móvel.

1.2. Objectivos

O objectivo central desta dissertação é a avaliação de uma solução MPLS para utilização em equipamentos da PT Inovação em que a pilha protocolar seja baseada em módulos disponíveis de software a correr em plataformas Linux. Será obrigatória a conformidade com os requisitos mínimos definidos pelo *Broadband-Forum* (especificação "*IP/MPLS Forum 20.0.0"*) para aplicação em redes de acesso rádio (RAN) baseadas em IP/MPLS. Com base nestes requisitos, é necessário realizar um levantamento das várias soluções existentes no mercado.

Por fim, esta dissertação tem também como objectivo traçar um plano de testes que possa avaliar a solução escolhida, de modo a identificar as capacidades disponibilizadas pela solução e as suas eventuais lacunas. Estas últimas serão objecto de uma breve análise em termos de disponibilidade no mercado e complexidade de desenvolvimento.

1.3. Metodologia

O desenvolvimento deste trabalho compreende três fases: revisão bibliográfica, levantamento de requisitos e descrição da solução escolhida, e planos de teste realizados com respectiva análise de resultados.

Durante a fase bibliográfica obteve-se um conhecimento mais aprofundado sobre as várias tecnologias MPLS, para que o levantamento de requisitos pudesse ser realizado de uma forma mais consciente e cuidada.

O plano de testes foi dividido em duas fases: a primeira que tem em conta todos os aspectos funcionais da solução e a sua conformidade com as normas; e uma segunda fase em que a solução é colocada num ambiente próximo de uma rede IP/MPLS real, permitindo avaliar o desempenho da mesma.

1.4. Organização da dissertação

Esta dissertação é composta por 6 capítulos, em linha com a metodologia seguida. O capítulo 2 apresenta de forma sucinta os principais conceitos e características do MPLS. Neste capítulo são também descritas algumas das tecnologias MPLS mais usadas no *backhaul* de redes móveis. O capítulo 3 aborda a evolução das redes móveis e o contexto em que se insere o IP/MPLS nestas redes. Este capítulo apresenta também os requisitos definidos, as características das várias soluções encontradas e por fim uma descrição da solução escolhida. Nos capítulos 4 e 5 são descritos planos de testes funcionais e de desempenho, respectivamente. Estes capítulos, além da descrição dos testes efectuados, contêm a exposição e análise dos resultados. Por fim, o capítulo 7 apresenta as conclusões do trabalho efectuado.

Capítulo 2: Conceitos MPLS

A tecnologia MPLS foi normalizada pelo *Internet Engineering Task Force* (IETF) e incorpora no paradigma do encaminhamento de pacotes as melhores características da comutação de circuitos. O termo *multiprotocol* resulta do facto do MPLS ser concebido para suportar qualquer protocolo da camada 3 do modelo *Open Systems Interconnection* (OSI), da qual o IP é o mais popular. A ideia básica do MPLS consiste em mapear toda a informação de encaminhamento de nível 3 apenas num único número, designado por etiqueta [2]. Assim, quando um pacote IP entra no domínio MPLS, toda a informação relativa ao encaminhamento é codificada numa etiqueta e, a partir daí, todo o encaminhamento dentro da rede passa a ser feito com base nela. Comparativamente ao encaminhamento IP, o MPLS torna-se mais eficiente uma vez que dispensa a consulta da informação de encaminhamento em todos os nós, realizando apenas comutação baseada nas etiquetas.

Os pacotes IP tratados da mesma forma pelos routers, com o mesmo caminho e mesma prioridade, podem ser agrupados num grupo designado por *Forwarding Equivalency Class* (FEC). Na terminologia MPLS, cada FEC é identificado por uma etiqueta. Esta, por sua vez, apenas é importante entre um par de equipamentos MPLS, garantindo assim a escalabilidade em ambientes mais complexos, uma vez que ela não necessita de ser a mesma em cada nó da rede [3].

A etiqueta MPLS pode estar também localizada em posições diferentes do pacote dependendo da tecnologia da camada 2 utilizada para transporte de dados. Se a tecnologia de nível 2 contempla um campo para a etiqueta, ela é encapsulada no cabeçalho nativo do protocolo. Por exemplo, em redes *Asynchronous Transmission Mode* (ATM) o campo *Virtual Path Identifier/Virtual Circuit Identifier* (VPI/VCI) pode ser utilizado para codificar a etiqueta. De modo análogo, o campo *Data Link Connection Identifier* (DLCI) pode codificar a etiqueta MPLS em redes *Frame Relay*. Nas tecnologias da camada 2 que não suportam etiquetas nativamente (por exemplo, a *Ethernet*), ela reside encapsulada entre o cabeçalho da camada 2 e o cabeçalho IP [3]. A Figura 2.1 apresenta a constituição de um cabeçalho MPLS, bem como a sua localização em tramas Ethernet e em células ATM.

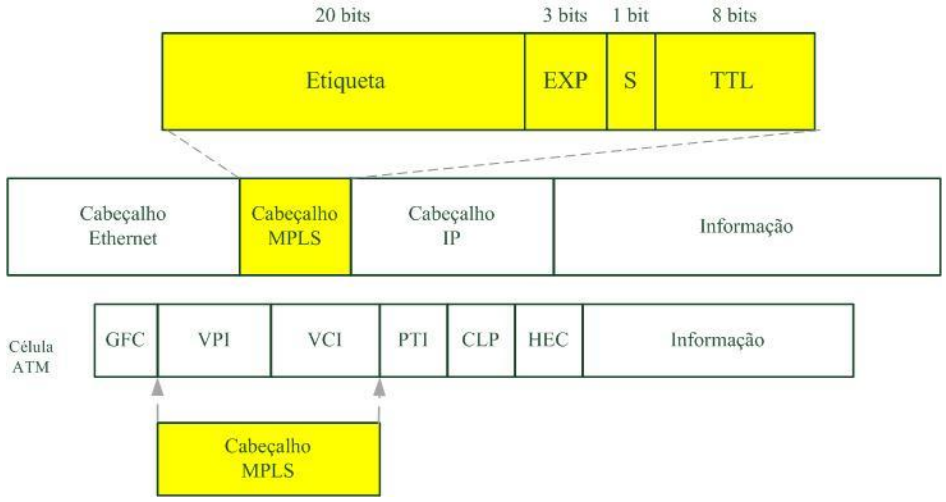


Figura 2.1 – Cabeçalho MPLS (32 bits) [2].

Na figura anterior são identificáveis os seguintes campos, que constituem o cabeçalho de um pacote MPLS [2]:

- **Etiqueta** – campo que contém o valor actual da etiqueta. Os valores de 0 a 15 encontram-se reservados.
- **EXP** – campo experimental, tipicamente usado para definir classes de serviço.
- **S** – o MPLS permite fazer pilha de etiquetas. Este campo indica o final da pilha.
- **TTL** – *Time to Live* – conta por quantos routers o pacote passou, num máximo de 255. No caso de o pacote viajar por mais de 255 routers, ele é descartado evitando assim possíveis *loops*.

2.1. Arquitectura e Funcionamento

Uma rede MPLS é constituída por diversos componentes, cuja função de cada um deles está ilustrada na Figura 2.2.

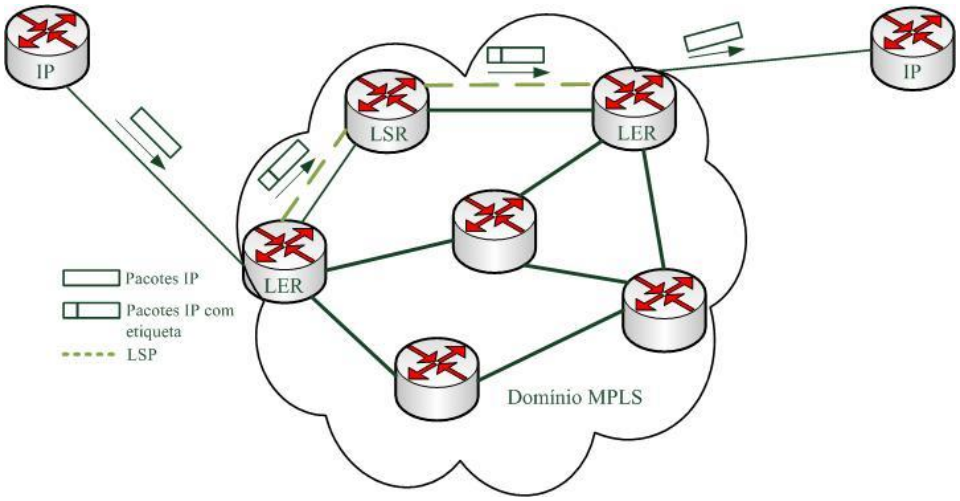


Figura 2.2 - Arquitectura e componentes de uma rede MPLS.

A figura anterior ilustra um cenário típico de uma rede MPLS, em que são identificáveis os seguintes componentes [2]:

- ***Label Switched Path* (LSP)** – túnel sobre a rede MPLS sobre o qual circulam pacotes que pertencem a um mesmo FEC.
- ***Label Switched Router* (LSR)** – nó que suporta protocolos de controlo MPLS, protocolos de encaminhamento IP e encaminha pacotes com base nas etiquetas. Estes componentes são capazes também de encaminhar pacotes IP nativos.
- ***Label Edge Router* (LER)** – processa o tráfego à entrada e à saída do domínio MPLS.
    - LER de entrada – examina o cabeçalho dos pacotes IP à entrada e classifica-os segundo um FEC, iniciando assim o LSP. Gera o cabeçalho MPLS e atribui a etiqueta inicial.
    - LER de saída – termina o LSP, removendo o cabeçalho MPLS.

No ponto de entrada de uma rede MPLS, um router LER adiciona uma etiqueta a cada um dos pacotes IP que chegam, com base no FEC a que pertencem, como referido anteriormente. Ao longo do caminho a troca de etiquetas baseia-se no estabelecimento de túneis sobre a rede, conhecidos como LSPs. Por outras palavras, um LSP é uma ligação MPLS lógica, que estabelece comunicação entre dois LERs via vários LSRs, como ilustrado na Figura 2.2.

Numa rede MPLS, existe um LSP por cada caminho existente na rede. Os routers examinam o cabeçalho dos pacotes que chegam à entrada da rede, determinam o LSP ao qual pertencem, adicionam a etiqueta correspondente e encaminham-nos para o próximo nó. Todos os nós seguintes encaminham os pacotes ao longo do LSP identificado pela etiqueta; tipicamente, os LSPs seguem o

caminho mais curto entre a origem e o destino. Dentro de uma rede MPLS apenas a etiqueta é usada para identificar o próximo nó para o qual o pacote vai ser enviado, em contraste com o tradicional encaminhamento realizado pelas redes IP. Esta propriedade das redes MPLS permite um eficiente uso dos recursos disponíveis e proporciona uma elevada velocidade de encaminhamento.

À medida que o tráfego circula pela rede MPLS, os LSRs consultam as suas tabelas de etiquetas, designadas por *Label Forwarding Information Base* (LFIB). Nestas tabelas, a cada interface e etiqueta de entrada está associada uma interface e etiqueta de saída. Assim, um LSR ao consultar a LFIB substitui a etiqueta de entrada pela de saída e transmite o pacote pela interface de saída [3].

A Figura 2.3 exemplifica aquilo que foi explicado anteriormente, com um cenário onde existem dois LSPs estabelecidos, e a forma como os pacotes são encaminhamos desde a entrada da rede MPLS até à saída da mesma.

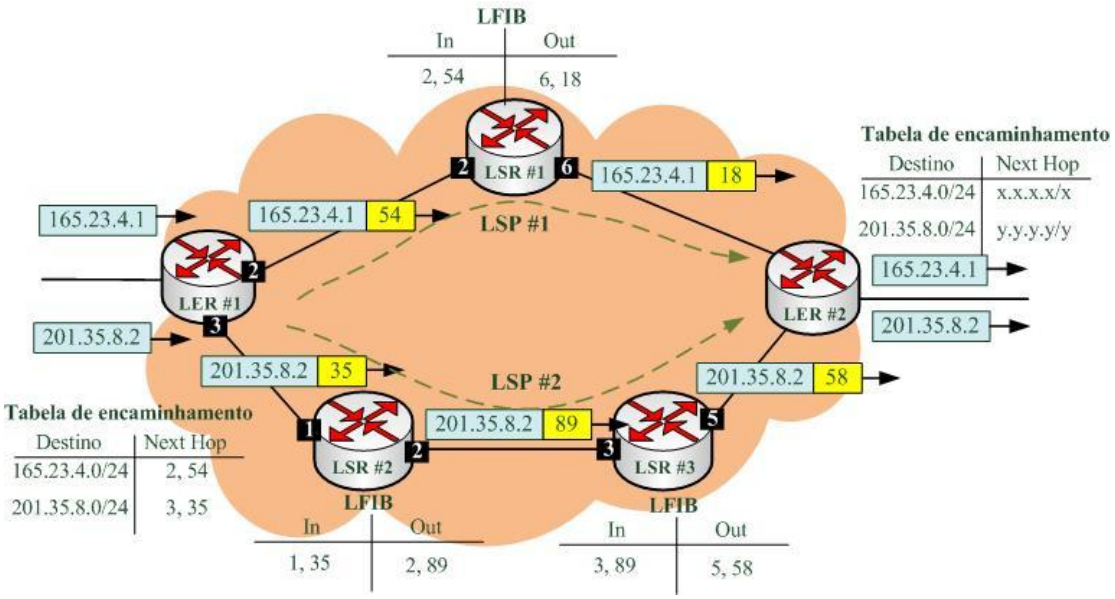


Figura 2.3 - Exemplo de transporte MPLS.

Na figura, o LER #1 é o ponto de entrada no domínio MPLS para os pacotes com o destino 165.23.4.1 e 201.35.8.2. Neste router, a tabela de encaminhamento IP tem a informação de qual o porto de saída e a etiqueta de saída para cada um dos pacotes. De seguida, os LSRs #1, #2 e #3 contém a tabela exclusivamente MPLS, a LFIB. Com base nesta tabela e na etiqueta de entrada, os pacotes MPLS são encaminhados para o porto de saída correcto, juntamente com a nova etiqueta. A etiqueta de entrada é removida antes de se inserir a nova. No router LER #3 os pacotes saem do domínio MPLS: são-lhes retiradas as etiquetas e encaminhados para o seu destino, com base nas tabelas de encaminhamento IP.

Na Figura 2.3, o LSR #2, por exemplo, é o LSR de *upstream* em relação ao LSR #3, ao passo que este é o LSR de *downstream* em relação ao LSR #2. Os conceitos de *Downstream* e *Upstream* dizem respeito ao sentido do fluxo de pacotes num LSP, na comunicação entre dois LSRs. Os pacotes viajam sempre de um LSR *upstream* (ou seja, o LSR anterior quanto ao sentido do fluxo), para um LSR *downstream* (ou seja, o LSR posterior quanto ao sentido do fluxo).

2.2. Componentes operacionais

Os componentes operacionais de uma rede MPLS podem ser divididos em dois grandes planos: plano de controlo e plano de dados. A Figura 2.4 ilustra as funções de cada um dos planos.

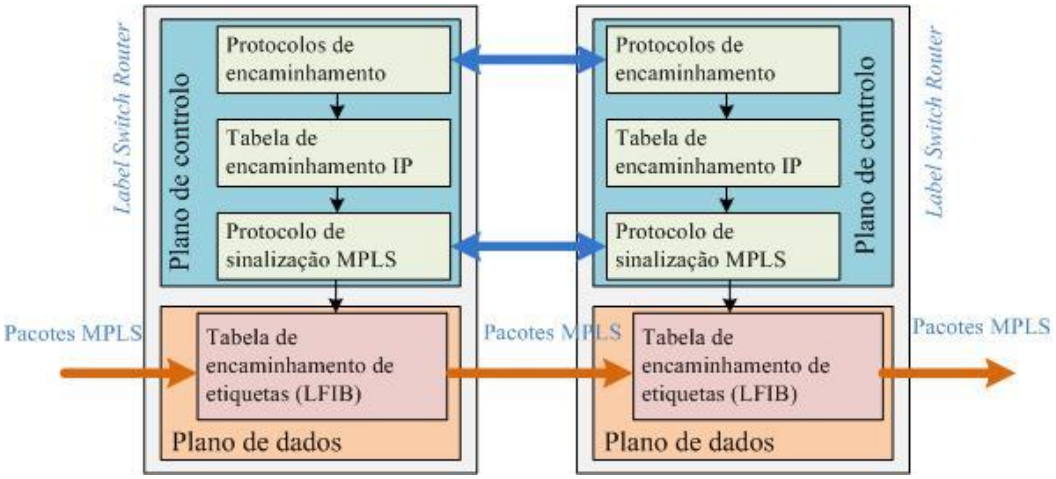


Figura 2.4 - Componentes operacionais numa rede MPLS.

O plano de controlo trata da troca de informação de encaminhamento e etiquetas entre equipamentos adjacentes [2]. Este plano constrói uma tabela de encaminhamento IP (*Routing Information Base* - RIB), baseando-se em protocolos de encaminhamento *Interior Gateway Protocol* (IGP), (por exemplo, o *Open Shortest Path First* (OSPF), o *Intermediate System-to-Intermediate System* (IS-IS) ou o *Routing Information Protocol* (RIP)), em rotas estáticas, ou mesmo em protocolos *Exterior Gateway Protocol* (EGP) (o *Border Gateway Protocol* (BGP)).

O plano de controlo inclui também protocolos de sinalização MPLS que têm o objectivo de especificar e manter as etiquetas internamente, e trocar a informação das etiquetas especificadas com outros equipamentos. Estes protocolos associam etiquetas a redes apreendidas via os protocolos de encaminhamentos referidos anteriormente. Exemplos destes protocolos de gestão de etiquetas são o *Label Distribution Protocol* (LDP), o *Multi Protocol* BGP (usado pelo MPLS *Virtual Private Network* - VPN) e o *Resource Reservation Protocol* (RSVP). Estes protocolos serão abordados posteriormente nesta dissertação [2].

O plano de dados é um simples mecanismo de encaminhamento que é independente do tipo de protocolo de encaminhamento ou do protocolo de gestão de etiquetas. O plano de dados encaminha os pacotes para a interface apropriada com base na tabela LFIB (conforme descrito na secção anterior).

2.3. *Merge* de etiquetas

Um LSR realiza *merge* de etiquetas se conseguir receber dois pacotes de interfaces diferentes, e/ou com diferentes etiquetas, e enviar ambos os pacotes pela mesma interface de saída com a mesma etiqueta. Esta situação está ilustrada do lado esquerdo da Figura 2.5: os pacotes com as etiquetas 34 e 56, em diferentes interfaces, chegam ao LSR e são ambos encaminhados com a etiqueta 23 pela mesma interface de saída. Isto só é possível se ambos os pacotes tiverem o mesmo destino X (mesmo FEC) e, claro, se o LSR suportar esta funcionalidade [1].

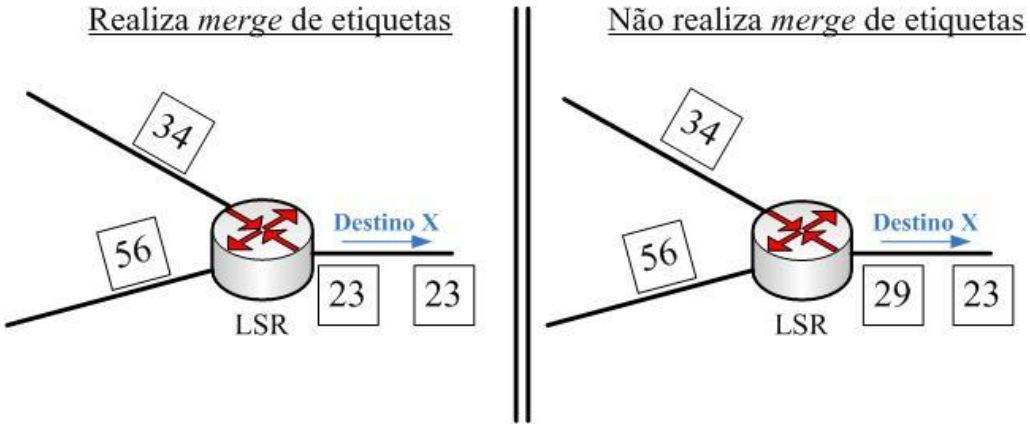


Figura 2.5 - LSRs com e sem *merge* de etiquetas.

Quando um LSR não realiza *merge* de etiquetas, quaisquer dois pacotes pertencentes ao mesmo FEC que cheguem por interfaces diferentes, ou com etiquetas diferentes, têm que ser encaminhados com etiquetas diferentes (mesmo que sejam encaminhados pela mesma interface de saída). O lado direito da Figura 2.5 ilustra uma situação em que o LSR não realiza *merge* de etiquetas: os pacotes com as etiquetas 34 e 56 são encaminhados no LSR para a mesma interface de saída mas com etiquetas diferentes, apesar de os pacotes terem o mesmo destino [1].

Assim, numa rede constituída por LSRs que realizem *merge* de etiquetas, o número de etiquetas necessárias por FEC em cada LSR será apenas uma. Com LSRs que não realizem *merge* de etiquetas, a quantidade de etiquetas por FEC, em cada LSR, pode ser igual ao número de LSRs na

rede. Nesta situação, cada LSR pode ter um conjunto de etiquetas associadas a um FEC, em que cada uma dessas etiquetas representa uma interface diferente do LSR.

Os LSRs que não realizam *merge* de etiquetas são tipicamente usados no transporte de tecnologias da camada 2. Por exemplo, no transporte do ATM é importante distinguir circuitos *Virtual Path/Virtual Circuit* (VP/VC) que cheguem por interfaces diferentes. Uma vez que pode chegar por interfaces diferentes o mesmo VP/VC, torna-se necessário distingui-los. Isso é conseguido quando o LSR não realiza *merge* de etiquetas.

2.4. Vantagens

O principal objectivo da comutação por etiquetas é o de trazer a velocidade de comutação do nível 2 para o nível 3. Métodos de comutação por etiquetas permitem aos routers tomar decisões de encaminhamento baseados apenas no conteúdo de uma simples etiqueta, em vez da consulta complexa de rotas baseadas nos endereços IP de destino. Com base nisto, surgem algumas vantagens importantes no uso do MPLS:

- Uma vez que é atribuído um FEC a um pacote que entra na rede MPLS, informação que não pode ser recolhida do cabeçalho da camada de rede, pode ser usada para atribuição de FECs. Por exemplo, a classificação de pacotes com base no endereço IP de origem.
- Os pacotes podem ser associados a etiquetas com diferentes prioridades, fazendo com que a qualidade de serviço das redes *Frame Relay* e ATM seja possível. Este ponto está relacionado com o campo EXP do cabeçalho MPLS, que será mencionado posteriormente nesta dissertação.
- As regras que determinam como é atribuído um FEC a um pacote podem tornar-se cada vez mais complexas sem que isto tenha qualquer impacto nos routers LSR que simplesmente encaminham os pacotes com base nas etiquetas previamente atribuídas.
- A informação contida nos pacotes não é analisada pelos routers de encaminhamento, permitindo assim diferentes níveis de encriptação e transporte de múltiplos protocolos.
- No MPLS, um pacote pode ser forçado a seguir uma determinada rota explícita em vez de seguir a rota escolhida pelos algoritmos de encaminhamento normais. Isto pode ser conseguido recorrendo à engenharia de tráfego (*Traffic Engineering* – TE), como uma questão de política ou então para suportar uma dada QoS.

2.5. *Label Distribution Protocol* (LDP)

2.5.1. Contexto

A principal característica do MPLS é o facto de os pacotes serem marcados com etiquetas, e cada LSR efectuar troca de etiquetas com o objectivo de encaminhar os pacotes. Isto significa que as etiquetas precisam de ser distribuídas pela rede para todos os LSPs. Tal tarefa poderia ser realizada ajustando os protocolos de encaminhamento IGP – como o OSPF, RIP ou IS-IS – para que transportassem as etiquetas. No entanto, isso iria fazer com que todos os protocolos IGP tivessem de ser modificados, o que não seria muito conveniente visto estes protocolos estarem já em funcionamento nas redes hoje em dia. Assim, foi necessário propor um novo protocolo, independente das questões de encaminhamento e, por isso, apto a funcionar com qualquer IGP. Esta é a principal razão pela qual surgiu o LDP; este protocolo associa etiquetas a FECs e permite que esta informação seja transportada numa rede MPLS [4].

Cada LSR cria localmente uma associação entre uma etiqueta e um prefixo IP. Estes prefixos IP são fornecidos pelas tabelas de encaminhamento IP e podem ser: endereço IP de destino, tipo de serviço, etc… De seguida, o LSR distribui esta informação a todos os seus vizinhos. Por sua vez, estes guardam esta informação e as suas associações locais numa tabela designada por *Label Information Base* (LIB). Dentro de todas as associações remotas para um prefixo IP, o LSR precisa de escolher uma e usá-la para determinar a etiqueta de saída. A esta etiqueta é associada a informação do próximo destino, fornecida pelos algoritmos de encaminhamento e guardada na tabela de rotas RIB. De seguida, toda esta informação é usada para construir a tabela LFIB, onde a etiqueta local é tratada como etiqueta de entrada e a etiqueta proveniente das tabelas de encaminhamento dos vizinhos é tratada como etiqueta de saída. Assim, quando um LSR recebe um pacote com etiqueta, já será capaz de remover essa etiqueta e inserir uma nova de saída atribuída ao próximo LSR [4].

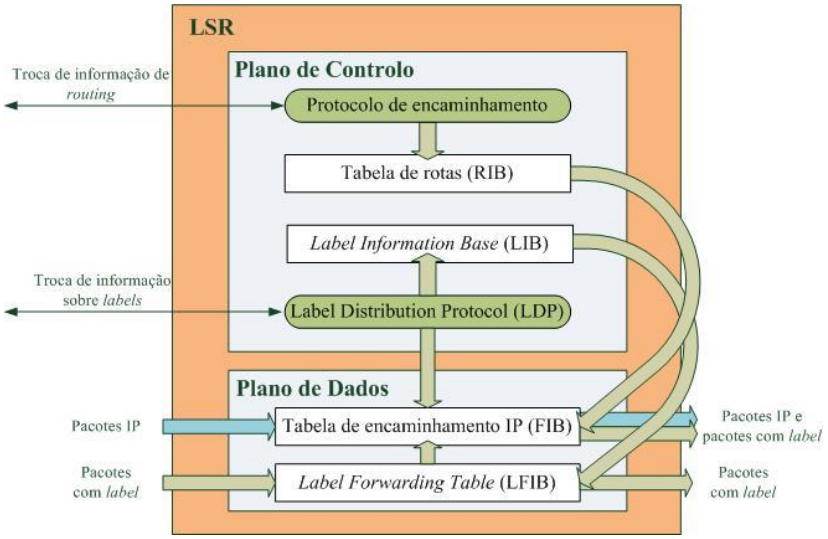


Figura 2.6 – Encaminhamento dentro de um LSR.

A Figura 2.6 mostra o diagrama de blocos usado pelos LSRs ao realizar o encaminhamento dos pacotes. No plano de controlo o protocolo de encaminhamento troca informação de encaminhamento e guarda-a na tabela RIB. Não esquecer que a tabela RIB poderá também conter rotas estáticas. O LDP por sua vez troca informação sobre etiquetas e guarda-a na tabela LIB. De seguida, toda esta informação é usada ao nível do plano de dados permitindo que as etiquetas fornecidas pelo plano de controlo sejam armazenadas na tabela *Forwarding Information Base* (FIB), mapeando um prefixo IP com uma etiqueta de saída, e as etiquetas fornecidas pelo plano de controlo sejam armazenadas na tabela LFIB e mapeadas para uma etiqueta de saída.

Com este esquema de encaminhamento passam a ser possíveis as seguintes situações dentro dum LSR:

- Um pacote IP de entrada é encaminhado, usando a tabela FIB, para a interface de saída como um pacote IP (encaminhamento IP normal).
- Um pacote IP de entrada é encaminhado, usando a tabela FIB, para a interface de saída com etiqueta (se existir alguma etiqueta associada à rede IP de destino).
- Um pacote de entrada com etiqueta é encaminhado, usando a tabela LFIB, para a interface de saída com etiqueta.
- Um pacote de entrada com etiqueta que veja a sua etiqueta removida, é encaminhado, usando a tabela FIB, para a interface de saída como um pacote IP.

2.5.2. *Label Space*

O protocolo LDP especifica um conjunto de procedimentos e mensagens que permitem aos LSRs estabelecerem LSPs através da rede. O LDP associa a cada LSP criado uma classe (FEC) para definição dos pacotes que serão mapeados nesse LSP.

Os LSRs que utilizam LDP para trocar informação sobre etiquetas e FECs são denominados por pares LDP e, para este fim, estabelecem entre si uma sessão LDP. A comunicação entre pares LDP é realizada através da troca de mensagens. Para a identificação do LSR gerador da mensagem, utilizam-se os Identificadores LDP, que são compostos pelo Identificador do LSR, globalmente único, mais o *Label Space*, que é definido dentro do âmbito do LSR.

O conceito de *Label Space* é importante na atribuição e distribuição de etiquetas. Existem dois tipos de *Label Spaces*: por interface e por plataforma [4].

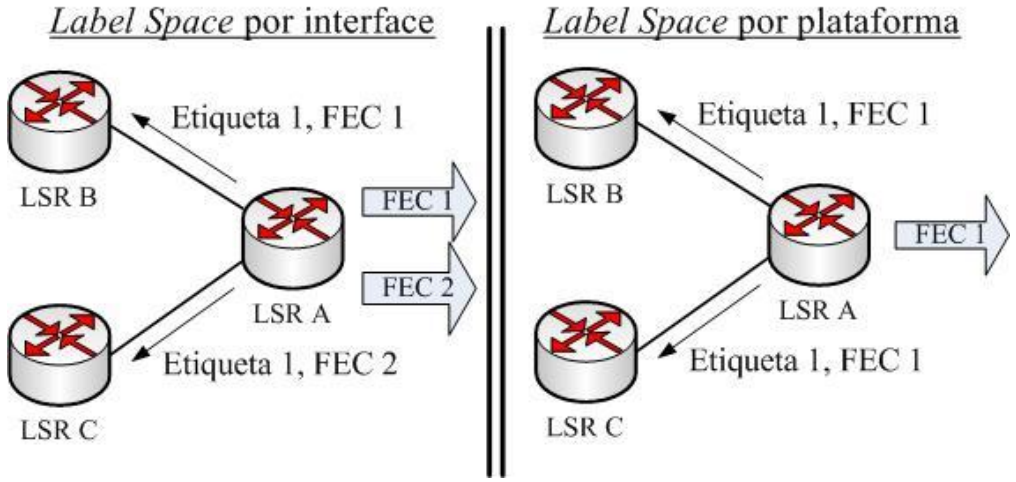


Figura 2.7 - *Label Space* por interface e por plataforma.

Do lado esquerdo da Figura 2.7 o LSR A pode anunciar a etiqueta 1 para o FEC 1 ao LSR B e, a mesma etiqueta 1 para o FEC 2 ao LSR C, mas apenas se, posteriormente, o LSR A conseguir distinguir de qual LSR o pacote com a etiqueta 1 foi recebido. Neste caso, a etiqueta 1 é única por interface, sendo o *Label Space* por interface. Neste caso, o pacote não é só encaminhado apenas com base na etiqueta, mas também com base na interface de entrada [4]**.**

A outra possibilidade é a etiqueta não ser única por interface, mas global ao LSR. Este caso é designado por *Label Space* por plataforma. Neste caso, o LSR A distribui o FEC 1 com a etiqueta 1 aos LSR B e C, como mostra o lado direito da Figura 2.7. Quando o LSR A distribuir uma etiqueta para o FEC 2, esta por sua vez deve ser diferente da etiqueta 1. No *Label Space* por plataforma, os

pacotes são encaminhados apenas com base na *etiqueta*, independentemente da interface de entrada [4].

Dentro de um LSR o *Label Space* é um número atribuído por interface. Se este número for igual para todas as interfaces do LSR, então este estará a usar *Label Space* por plataforma. Se este número for diferente em pelo menos uma das interfaces, então o LSR estará a usar *Label Space* por interface. Uma sessão LDP entre dois LSRs apenas pode ser estabelecida sobre um *Label Space* comum.

2.5.3. Estabelecimento de sessões LDP

Para estabelecer um LSP cada LSR deve: descobrir outros vizinhos LDP na rede; estabelecer sessões LDP com cada vizinho; enviar mensagem de solicitação de etiqueta para um determinado FEC; e responder à solicitação com o mapeamento etiqueta/FEC correspondente. Quando todos os LSRs do LSP obtiverem o mapeamento para o FEC, o LSP estará estabelecido. O LDP é assim, na sua essência, um conjunto de troca de mensagens que permitem aos LSRs obter as etiquetas usadas pelos seus LSRs vizinhos.

A especificação LDP define quatro categorias de mensagens que são trocadas entre os LSRs [5]:

- **Mensagens de descoberta**: utilizadas para anunciar e manter a presença de um LSR na rede;
- **Mensagens de sessão**: usadas para estabelecer, manter e terminar uma sessão entre pares LDP;
- **Mensagens de divulgação**: utilizadas para criar, alterar e remover mapeamentos de etiquetas para FECs;
- **Mensagens de notificação**: usadas para trocar informação e sinalizar erros.

Antes que dois LSRs adjacentes possam iniciar a troca de mensagens para distribuição de etiquetas, eles devem estabelecer uma relação de vizinhança. Para descobrir se um LSR vizinho suporta o protocolo LDP, o LSR envia periodicamente mensagens de descoberta (*Hello*) aos seus vizinhos. Estas mensagens são enviadas como pacotes *User Datagram Protocol* (UDP), no porto 646 (LDP *well-know discovery port*), normalmente para o endereço de grupo *multicast* 224.0.0.2, caracterizando o mecanismo de descoberta básico do LDP [4].

Dois LSRs que não sejam vizinhos (não estejam ligado directamente um ao outro) podem igualmente trocar mensagens de *Hello*, caracterizando o mecanismo de descoberta estendido do LDP. Neste caso, as mensagens de *Hello*, denominadas por "*Targeted Hello*", são direccionadas

para um endereço específico, via comunicação *unicast*. Tal como no mecanismo de descoberta básico, as mensagens "*Targeted Hellos*" são enviadas em pacotes UDP para o porto 646.

Quando um LSR decide estabelecer uma sessão com outro LSR descoberto pelas mensagens de *Hello*, é iniciada uma sessão LDP sobre *Transmission Control Protocol* (TCP). Assim que este procedimento termina com sucesso, os dois LSRs passam a ser pares LDP, e podem trocar mensagens de divulgação [5].

Numa sessão LDP, um dos LSRs inicia uma ligação TCP – porto TCP 646 – com o outro LSR. Se esta ligação TCP for estabelecida com sucesso, ambos os LSRs negociam parâmetros da sessão LDP trocando mensagens de inicialização. Entre estes parâmetros estão, por exemplo, os temporizadores e método de distribuição de etiquetas configurado [4].

Se ambos os LSRs de uma sessão LDP concordarem com os parâmetros propostos pelo outro, a sessão TCP é mantida. Caso contrário, uma nova sessão TCP é tentada. Depois de a sessão ter sido estabelecida com sucesso, a sessão é mantida através ou do envio de pacotes LDP ou de mensagens periódicas de *keepalive*. Por cada vez que um LSR recebe um pacote LDP ou uma mensagem de *keepalive*, é reinicializado um temporizador para esse mesmo par LDP [4].

As mensagens de divulgação servem para cumprir com o principal objectivo do LDP: divulgar mapeamentos de etiquetas. Existem diversos modos que os LSRs podem utilizar para divulgar etiquetas, referidos na secção seguinte. Apenas existem dois tipos destas mensagens: as mensagens de pedido e de mapeamento de etiqueta. As mensagens de pedido de etiqueta são enviadas pelos LSRs de *upstream* para os LSRs de *downstream*, com o objectivo de solicitar aos LSRs de *downstream* os seus mapeamentos. Como é óbvio, as mensagens de mapeamento de etiqueta são usadas pelos LSRs de *downstream* para anunciar os seus mapeamentos aos seus LSRs de *upstream*.

As mensagens de notificação são necessárias para "limpeza" das sessões LDP. São elas que notificam eventos entre os pares LDP. Estes eventos podem ser erros fatais (*Notification Error*) ou simples informações (*Advisory Notifications*). Se acontecer algum erro fatal entre dois LSRs de um par LDP, ambos devem terminar a sessão LDP. Os eventos seguintes podem ser assinalados através destas mensagens de notificação [5]:

- Mensagens mal construídas
- Temporizador *keepalive* expirado
- Sessão terminada num dos LSR
- Eventos das mensagens de inicialização
- Eventos resultantes de outras mensagens
- Erros internos ao LSR

- Detecção de *loops*
- Diversos eventos

O Anexo I apresenta um resumo do formato das várias mensagens LDP definidas por [5].

2.5.4. Distribuição e gestão de etiquetas

A arquitectura MPLS permite que um LSR distribua etiquetas para um FEC em resposta a uma solicitação explícita de outro LSR. Este método é denominado por *Downstream On Demand*, e permite que determinado LSR solicite uma etiqueta ao seu vizinho LSR de *downstream*. Existe ainda o método *Downstream Unsolicited*, em que o LSR pode distribuir etiquetas sem que tenha ocorrido uma solicitação prévia. Neste método, um LSR recebe todos os mapeamentos de etiquetas de cada um dos LSR adjacentes. Estes dois métodos de distribuição podem ser usados simultaneamente, contudo, para uma determinada sessão LDP, cada LSR tem de conhecer o método de distribuição usado pelo seu par LDP [5].

O controlo de distribuição de etiquetas pode ser independente - em que um LSR pode gerar e divulgar etiquetas para os seus vizinhos a qualquer momento - ou ordenado – em que um LSR apenas pode iniciar a transmissão de mapeamento de etiquetas para um FEC sobre o qual ele já possua mapeamento ou quando for o LER de saída. Neste último modo, se não existir qualquer mapeamento para uma determinada FEC, num LSR que não seja LER de saída, este deve esperar até receber uma etiqueta do LSR de *downstream* e enviar a etiqueta correspondente para os LSRs de *upstream* [5].

A vantagem do controlo de distribuição independente é o rápido estabelecimento dos LSPs. Como cada LSR processa os seus mapeamentos independentemente dos outros LSRs, a distribuição de etiquetas é realizada mais rapidamente. A desvantagem do controlo de distribuição independente é o facto de alguns LSRs poderem imediatamente iniciar o encaminhamento de pacotes MPLS, antes de o LSP estar estabelecido extremo-a-extremo. Nestas situações, os pacotes poderão não receber em todo o lado o encaminhamento devido ou poderão mesmo ser descartados.

No LDP existe também um modo de retenção de etiquetas que define se o LSR deve ou não manter um mapeamento para um FEC, obtido de um LSR que não é o próximo salto para o destino associado a esse FEC. Este modo pode ser conservador ou liberal. No modo conservador é retido apenas o mapeamento do próximo salto de acordo com o encaminhamento. No modo liberal são mantidos todos os mapeamentos obtidos dos parceiros LDP [5].

O modo liberal fornece uma adaptação rápida às alterações de encaminhamento, uma vez que todos os mapeamentos de etiqueta se encontram em memória, reagindo assim mais rapidamente a quebras de ligações ou nós. O modo conservador guarda menos etiquetas permitindo assim uma melhor utilização da memória livre.

2.6. MPLS-TE

O modelo de encaminhamento IP baseia-se sempre na escolha da rota que oferece menos custos. Os pacotes IP são encaminhados em cada nó com base no endereço IP de destino e independentemente do caminho pelo qual os pacotes atingiram esse nó. Além disso, o encaminhamento IP não tem em conta a largura de banda utilizada de cada ligação, que muitas vezes difere significativamente do custo que está atribuído a essa ligação. Assim, um router pode continuar a encaminhar tráfego IP por uma determinada ligação, mesmo que ela esteja a descartar tráfego devido à falta de largura de banda. A consequência desta situação no encaminhamento IP é que algumas ligações poderão ficar sobreutilizadas, em contraste com outras subutilizadas. Adicionar mais largura de banda a estas ligações sobreutilizadas necessita de tempo e planeamento. Devido aos padrões de tráfego mudarem repentinamente e não serem sempre permanentes numa rede, a Engenharia de Tráfego (TE) oferece uma solução desviando o tráfego, ou parte dele, das ligações sobreutilizadas [6].

As redes MPLS podem usar os mecanismos TE nativos de forma a minimizar a congestão e melhorar o seu desempenho. Os algoritmos TE modificam os percursos de encaminhamento com o objectivo de fornecer um mapeamento mais eficiente dos fluxos de tráfego aos recursos da rede. Este mapeamento eficiente pode reduzir a eventualidade de congestão e melhorar a qualidade de serviço em termos de latência, *jitter* e atraso. Historicamente, as redes IP apoiam-se na optimização da infra-estrutura das camadas inferiores e nos protocolos IGP para assim realizarem o TE. Em vez disto, o MPLS adiciona extensões aos protocolos IP já existentes e faz uso das capacidades de encaminhamento MPLS para assim fornecer TE nativo. Assim, tendo em conta a largura de banda configurada de cada ligação e os atributos de cada uma (atraso, *jitter*, …), o MPLS-TE permite uma distribuição mais eficiente do tráfego por toda a rede, evitando assim ligações subutilizadas e sobreutilizadas [7].

O MPLS-TE tem também a capacidade de realizar encaminhamento com base no endereço de origem. Isto é possível porque o MPLS faz o encaminhamento no plano de dados com base na etiqueta de entrada. A Figura 2.8 ilustra esta característica do MPLS-TE.

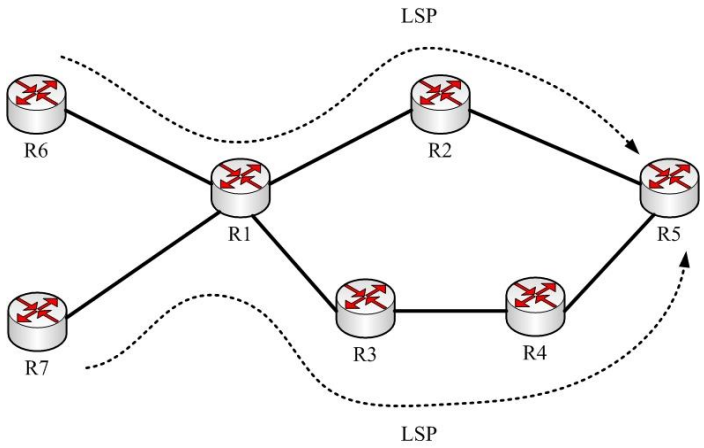


Figura 2.8 - Rede com MPLS-TE.

Nesta figura, os routers R6 e R7 enviam tráfego para uma rede servida pelo router R5. Se a rede estiver apenas a realizar encaminhamento IP, este tráfego irá seguir apenas o caminho R1-R2-R5 porque o encaminhamento de pacotes IP é feito independentemente em cada nó da rede. Assim, R1, que não conhece nada acerca de R6 e R7, encaminha o tráfego de acordo com sua tabela de encaminhamento IP, baseada no caminho com menor custo. No entanto, R6 e R7 podem ter diferentes políticas de encaminhamento e R6 poderá querer enviar o tráfego pelo caminho R6-R1-R2-R5, enquanto R7 poderá querer enviar por R7-R1-R3-R4-R5. Isto é impossível de realizar numa rede IP. Numa rede MPLS, podem ser estabelecidos estes dois caminhos através de dois LSPs, usando deste modo etiquetas diferentes para cada um dos caminhos. No router R1, as diferentes etiquetas de entrada indicam a qual LSP o pacote pertence, encaminhado assim o pacote para um dos dois LSPs. Esta característica do MPLS permite assim escolher qual o caminho que os pacotes devem seguir, em contraste com o encaminhamento IP onde cada router realiza o encaminhamento com base no endereço IP de destino dos pacotes.

2.6.1. Funcionamento do MPLS-TE

O funcionamento do MPLS-TE está dividido em quatro etapas: distribuição do estado de cada ligação, cálculo do caminho, sinalização do LSP e selecção de tráfego [7]. De seguida, serão descritas cada uma destas etapas.

2.6.1.1. Distribuição do estado da ligação

O MPLS-TE acrescenta extensões aos protocolos de encaminhamento IP usados, de forma a poder distribuir informação não só sobre a topologia da rede mas também sobre o seu estado. Um LSR necessita de informação detalhada sobre a rede para poder desempenhar encaminhamento baseado em restrições. Esta informação é distribuída pelos protocolos de encaminhamento capazes difundir o estado de cada ligação (OSPF e IS-IS são os mais conhecidos). Estes protocolos pertencem aos protocolos de encaminhamento do tipo *Link State*. Desta forma, cada LSR utiliza esta informação para construir uma base de dados contendo todos os caminhos possíveis para chegar a um determinado destino. Além desta base de dados, um LSR necessita também de ter toda a informação sobre as restrições de cada ligação. Esta informação é definida num conjunto de atributos associados com o TE e que são transportados pelos protocolos de encaminhamento referidos, usando extensões. Estes atributos de TE são os seguintes: métrica TE, largura de banda máxima, largura de banda máxima reservável, largura de banda não reservada e grupos administrativos [6].

A métrica TE é um parâmetro que pode ser usado para construir uma topologia TE diferente da topologia IP. Como tal, a métrica TE de uma ligação pode ser diferente da métrica do protocolo de encaminhamento IP. A largura de banda máxima é a largura de banda total de uma determinada ligação. A largura de banda máxima reservável é a largura de banda que pode ser reservada para TE, ao passo que a largura de banda não reservada é a restante largura de banda disponível para TE. Esta restante largura de banda pode ser dividida em oito partes correspondendo aos oito níveis de prioridade que um LSP TE pode ter. Estas oito prioridades podem ser definidas no campo EXP do cabeçalho MPLS. Assim, cada prioridade poderá ter associada uma determinada largura de banda. O grupo administrativo é um campo de 32 bits sem significado especial. O operador da rede pode usar este campo para diferentes objectivos e, habitualmente, é usado para definir regras de inclusão ou exclusão de ligações. Todos estes atributos podem ser definidos pelo operador da rede dotando assim a rede de características TE, que lhe permite optimizar o seu encaminhamento [6].

2.6.1.2. Cálculo do caminho

Antes do estabelecimento dum LSP usando TE, cada LSR precisa de determinar o caminho para o estabelecimento desse mesmo LSP. Este processo é realizado recorrendo à base de dados com informação sobre a topologia da rede e também às restrições configuradas para cada ligação. Neste processo é usado uma extensão ao algoritmo *Shortest Path First* (SPF) (usado no OSPF), designado por *Constraint-based Shortest Path First* (CSPF). Este algoritmo determina o caminho

para LSP TE de acordo com os requisitos especificados para cada ligação. Este algoritmo pode usar a métrica do protocolo IGP ou a métrica TE com o objectivo de determinar o caminho mais curto [7]. A Figura 2.9 mostra um exemplo simplificado dos cálculos realizados para determinar um caminho usando o CSPF.

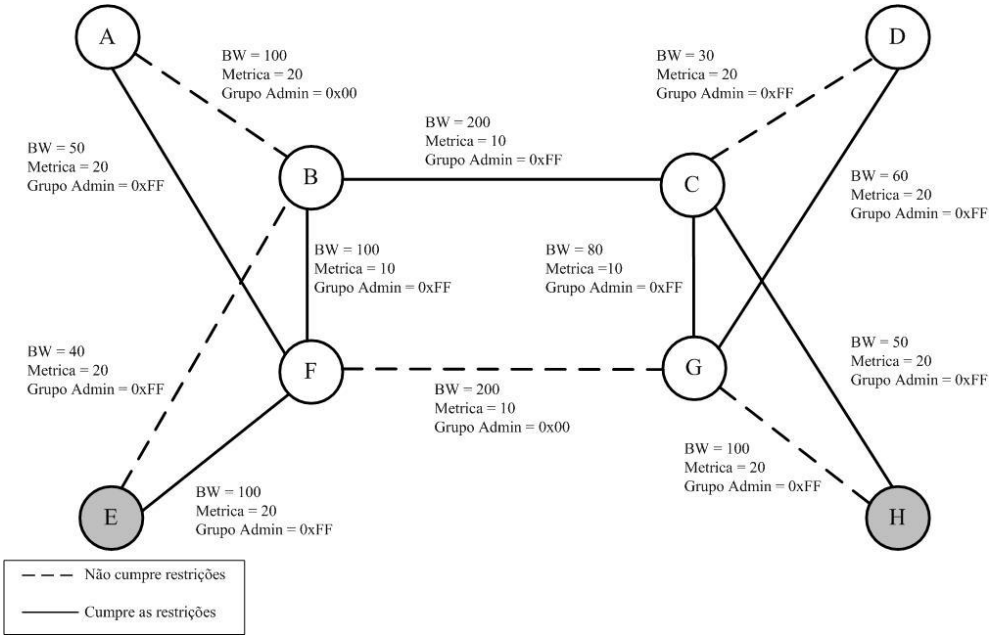


Figura 2.9 - Cálculo do caminho usando o algoritmo CSPF [7].

No caso da figura anterior, o nó E quer calcular o caminho mais curto para chegar ao nó H com os seguintes requisitos: apenas ligações com pelo menos 50 unidades de largura de banda disponível e um valor de grupo administrativo de 0xFF. O Nó E consulta a base de dados TE e descarta todas as ligações com largura de banda insuficiente ou com grupo administrativo diferente de 0xFF. Na figura as linhas a tracejado representam as ligações que foram descartadas pelo CSPF. De seguida, sobre esta topologia mais reduzida, o nó E calcula o caminho mais curto com base nos valores de métrica em cada ligação. Neste caso, o caminho mais curto é E-F-B-C-H. Com este resultado, o nó E pode iniciar a sinalização do LSP TE.

2.6.1.3. Sinalização RSVP-TE

Depois de descoberto o caminho, é necessário um protocolo de sinalização para que o LSP seja estabelecido e as etiquetas possam ser distribuídas. Para o MPLS-TE foi adoptado o RSVP, o qual foi melhorado de forma a poder suportar túneis TE. O RSVP básico é usado para reservar determinadas recursos da rede com o objectivo de garantir QoS para uma aplicação ou fluxo particular. Sendo o único protocolo, normalizado pelo IETF, que providencia sinalização QoS para

as redes IP, o RSVP foi estendido para suportar o estabelecimento e manutenção de LSPs TE, passando a ser designado por RSVP-TE [7].

No inicio da especificação do MPLS-TE, o IETF adoptou o LDP com extensões como protocolo de sinalização de LSPs TE. Estas extensões eram designadas como *Constraint-based routed* LDP (CR-LDP). Durante algum tempo, as especificações do CR-LDP e do RSVP-TE evoluíram simultaneamente. Em 2002, o IETF decidiu não prosseguir com os desenvolvimentos para o CR-LDP e, em vez disso, focou-se no RSVP-TE como principal protocolo para o MPLS-TE [6].

O RSVP utiliza mensagens designadas por PATH e RESV para sinalizar um caminho. O RSVP-TE acrescentou novos objectos dentro destas mensagens, de forma a poder estabelecer LSPs. Quando determinado LSR necessita de estabelecer um LSP, ele envia uma mensagem PATH com o objecto "Solicitação de Etiqueta" – em inglês, *Label Request*. Quando o último router do túnel TE recebe esta mensagem, atribui uma etiqueta a este LSP TE e anuncia esta atribuição ao router de *upstream* através de uma mensagem RESV com o objecto "Etiqueta". Esta etiqueta é a etiqueta de entrada na LFIB do último router. O router de *upstream* recebe esta etiqueta e coloca-a como etiqueta de saída na LFIB para este LSP. Este router associa uma nova etiqueta para este LSP e envia-a numa mensagem de RESV para o router de *upstream*. Este processo continua até chegar ao LSP que solicitou o estabelecimento do LSP [7]. Este processo é semelhante ao LDP; a diferença consiste na escolha das ligações sobre as quais o LSP é estabelecido. No MPLS-TE esta escolha é feita com base nos pressupostos referidos anteriormente, ao passo que no LDP o caminho é escolhido com base no protocolo de encaminhamento IP.

Além dos objectos "Solicitação de Etiqueta" e "Etiqueta", existem outros objectos adicionados às mensagens RSVP pelo MPLS-TE. A lista de objectos adicionados ao MPLS-TE é a que está descrita na tabela seguinte [7].

Objecto RSVP	**Mensagem RSVP**
Solicitação de Etiqueta	PATH
Etiqueta	RESV
Rota Explicita	PATH
Registo de Rota	PATH, RESV
Atributos da Sessão	PATH

Tabela 2.1 - Novos objectos RSVP para suportar MPLS-TE [7].

Todos os objectos da tabela anterior estão incluídos nas mensagens de PATH e RESV, usadas no estabelecimento do LSP. Os objectos "Solicitação de Etiqueta" e "Etiqueta" são usados para a distribuição de etiquetas, usando o modo *Downstream-on-Demand*. O objecto "Rota Explicita" contém uma lista dos saltos que definem a rota sobre a qual a sinalização irá seguir. O objecto "Registo de Rota" reúne todos os saltos e etiquetas registados durante o estabelecimento do LSP. Por fim, o objecto "Atributos da sessão" contém uma lista dos requisitos para o LSP (prioridades, protecção, etc).

2.6.1.4. Selecção de tráfego

O MPLS-TE separa o estabelecimento do LSP do processo de selecção de tráfego que irá usar o LSP TE. O critério de selecção poderá ser estático ou dinâmico. Poderá também depender do tipo de encapsulamento (por exemplo, IP ou Ethernet) ou do conteúdo dos pacotes (por exemplo, classe de serviço). Uma rede MPLS pode usar vários mecanismos de selecção de tráfego dependendo dos serviços em uso.

Um dos mecanismos usado pelo MPLS-TE para a selecção de tráfego é o *DiffServ*-TE (DS-TE), uma extensão ao *DiffServ* usado nas redes IP. O DS-TE fornece um controlo mais granular para minimizar a possibilidade de congestão e melhora o desempenho da rede. Este mecanismo usa o mesmo princípio de operação do MPLS-TE, adicionando apenas extensões para suportar o conceito de múltiplas classes de tráfego, tornando possível o encaminhamento baseado em restrições por classes. Estas melhorias ajudam no controlo da quantidade de tráfego de diferentes classes que circula em cada ligação de uma rede MPLS [7].

Numa determinada ligação, ambos os mecanismos DS-TE e *DiffServ* dividem a largura de banda disponível por classes. O DS-TE actua como um mecanismo do plano de controlo, enquanto que o *DiffServ* actua no plano de dados. Desta forma, estes dois mecanismos podem usar quantidades diferentes de classes e diferentes larguras de banda com o objectivo de satisfazer os requisitos de uma rede particular. No *DiffServ* a distribuição de largura de banda é realizada com base priorização de cada classe de serviço e na largura de banda máxima disponível. No DS-TE a distribuição de largura de banda é realizada com base nas classes de serviço e na largura de banda reservada para cada uma dessas classes [7].

2.6.2. Fast Reroute

O MPLS-TE suporta protecção local de LSPs usando uma técnica designada por *Fast Reroute* (FRR). A protecção do tráfego, no caso de ocorrer uma falha na rede, é crítica quando se trata de serviços em tempo real ou de qualquer outro tráfego com restrições em termos de perdas. O FRR utiliza uma aproximação de protecção local, que se baseia na pré-sinalização dum LSP de reserva para comutar o tráfego em cada de falha. O nó que se encontra imediatamente antes da falha é responsável por comutar o tráfego. Evita-se assim o atraso provocado pela propagação da informação de falha ao LER origem de cada LSP, o atraso necessário para calcular um novo caminho e também de sinalizar um novo LSP para comutar o tráfego. Após a ocorrência duma falha, o FRR consegue comutar o tráfego em dezenas de milissegundos [7]. Esta técnica chega a ser mais preferida que os mecanismos de protecção da camada 1 – como por exemplo o *Automatic Protection Switching* (APS), usado na protecção de ligações ópticas – uma vez que dispensam ligações de protecção inactivas, usadas apenas quando ocorrem falhas.

As especificações do FRR oferecem duas técnicas de protecção: protecção simples e protecção um para um. A protecção simples faz uso da *stack* de etiquetas para proteger múltiplos LSPs utilizando apenas um único LSP de protecção. A protecção um para um não usa *stack* de etiquetas e cada LSP protegido necessita de um LSP de protecção dedicado [7]. De seguida, apenas se aborda a protecção simples por ser a técnica mais usada e de maior escalabilidade.

A Figura 2.10 apresenta um exemplo de actuação do FRR, técnica de protecção simples.

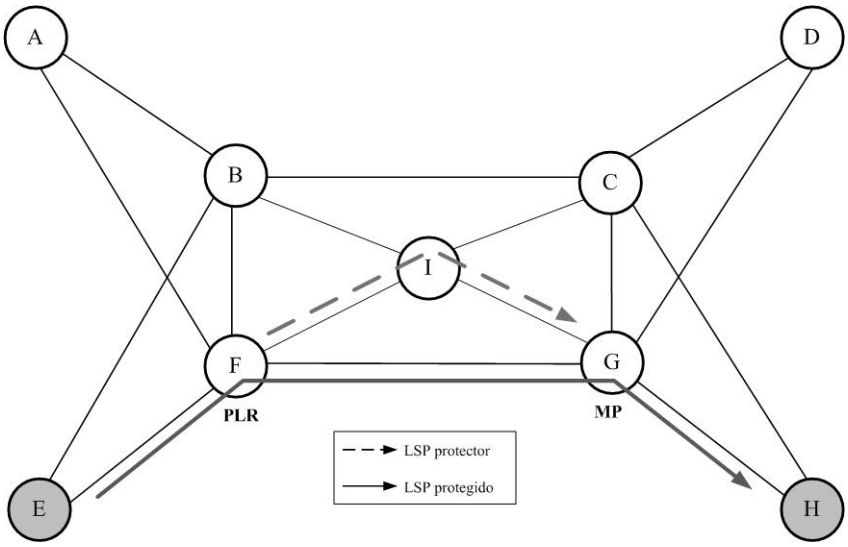


Figura 2.10 - *Fast Reroute* numa rede MPLS [7].

Nesta figura, o nó E sinaliza um LSP TE em direcção ao nó H. A rede protege este LSP contra falhas nas ligações entre o nó F e G. Devido à característica de protecção local do FRR, o nó F é

responsável por comutar o tráfego para o LSP protector no caso da ligação entre F e G falhar. Este papel faz do ponto F o Ponto de Reparação Local (PRL). Este ponto pré-sinalizou um LSP protector através do nó I em direcção ao nó G, protegendo assim a potencial falha. O nó G recebe a designação de Ponto de Convergência (PC) e é o ponto onde o tráfego sairá do LSP protector durante a falha, retomando o caminho original do LSP protegido. Como se pode observar o LSP protector já se encontra estabelecido mesmo antes da falha ocorrer, minimizando assim as perdas devido à falha e comutação do tráfego.

O FFR introduz um novo objectos nas mensagens RSVP-TE, com o fim de sinalizar o LSP protector. Este objecto é designado por "*Fast Reroute*" e especifica a técnica a ser usada (protecção simples ou um para um), e as características desejadas (prioridades, largura de banda, atributos, etc) do LSP protector.

O FRR pode actuar também para protecção de nós e não apenas para protecção de ligações.

2.7. MPLS VPN

O MPLS VPN é uma das mais populares e generalizadas implementações na tecnologia MPLS. Usar o MPLS para implementar VPNs é uma das soluções cada vez mais usadas pelos grandes operadores quando estes pretendem estruturar a sua rede em redes independentes mais pequenas, partilhando a mesma infra-estrutura [8].

Uma VPN é uma rede privada construída sobre uma infra-estrutura de recursos partilhados. Esta infra-estrutura pode conter diversas VPNs completamente independentes. Além disto, as VPNs construídas na camada IP podem exigir conectividade entre elas e também ligação à Internet. O MPLS VPN suporta todas estas características, dado que, ao contrário do IP, o MPLS dissocia o plano de controlo do plano de dados [2].

Existem actualmente três tipos de VPNs que podem ser suportados pelo MPLS: as VPNs MPLS/BGP (de camada 3), as VPNs da camada 2 do tipo ponto-a-ponto (L2VPN) e as VPNs da camada 2 do tipo ponto-multiponto mais conhecidas por *Virtual Private LAN Service* (VPLS) [9]. De seguida será abordado cada um destes tipos de VPNs.

2.7.1. VPN BGP/MPLS

As VPNs de camada 3, construídas sobre redes MPLS, são completamente transparentes para o cliente, não necessitando de configurações adicionais, endereçamento ou novos equipamentos [9]. Estas VPNs beneficiam de todas as vantagens associadas à tecnologia MPLS.

A arquitectura duma MPLS VPN é descrita pela RFC 2547, que tem o seu próprio vocabulário para identificar cada um dos elementos de rede. Nesta arquitectura, a VPN apenas existe na fronteira da rede do operador. Os routers do núcleo da rede não participam nas VPNs e apenas realizam o encaminhamento sobre os vários LSPs [9]. A Figura 2.11 apresenta essa arquitectura, bem como a designação dada a cada um dos seus elementos.

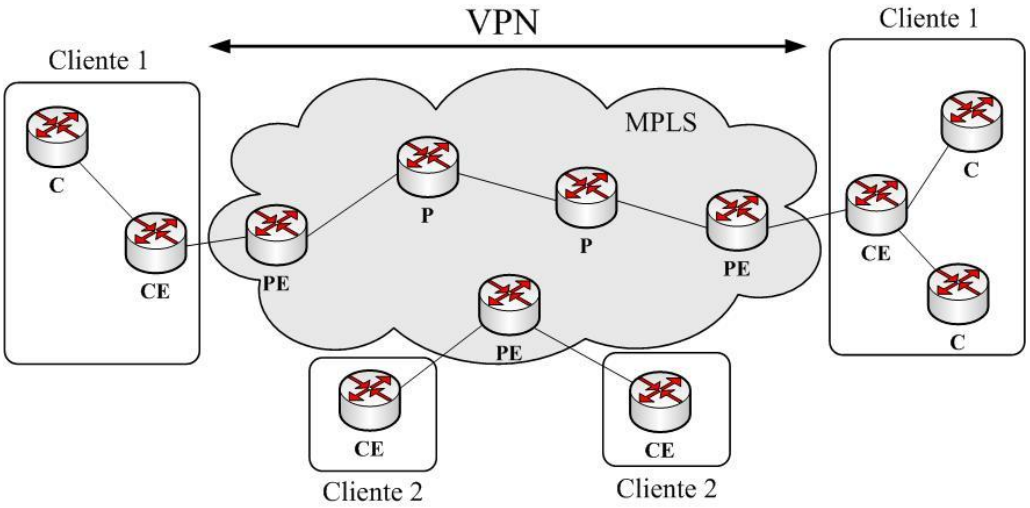


Figura 2.11 - Topologia de uma rede BGP/MPLS VPN [8].

Nesta figura, a VPN usada pelo cliente 1 atravessa a rede MPLS. Cada router nesta VPN tem a sua função específica. O router PE, que significa *Provider Edge*, é o router que inicia e termina a VPN na fronteira do MPLS. De acordo com a RFC 2547, estes routers são os responsáveis por todas as funções de estabelecimento, manutenção e operação destas VPNs. Um router PE está ligado directamente ao router do cliente, designado por *Costumer Edge* (CE). Esta ligação é realizada ao nível da camada 3, sendo responsabilidade do cliente a escolha do protocolo de encaminhamento a usar (RIP, OSPF, BGP, rotas estáticas, etc) para estabelecimento das rotas. Note-se que o router CE desconhece a existência do MPLS ou mesmo da VPN. Dentro da rede do cliente podem existir ainda os routers C, de *Costumer*, que não têm uma ligação directa com o PE e que não realizam MPLS [9]. Tipicamente, um PE está ligado a múltiplos routers CE suportando diferentes clientes.

Os routers P, de *Provider*, são os routers do núcleo da rede MPLS. Estes routers são os LSRs referidos na arquitectura do MPLS. Estes routers encaminham pacotes MPLS sobre os LSPs estabelecidos e, tal como os routers CE, estão completamente alheios à existência da VPN [9].

Os routers PE participam no encaminhamento e nos esquemas de endereçamento IP de todos os clientes directamente ligados. Muitos destes esquemas de endereçamento poderão sobrepor-se quando são usados prefixos privados (por exemplo, a rede 10.x.x.x é frequentemente usada). Os routers PE terão de garantir que, por exemplo, o tráfego destinado à rede 10.x.x.x do cliente A não será entregue inadvertidamente à rede 10.x.x.x do cliente B. Para conseguir isto, os routers PE têm tabelas de encaminhamento individuais para cada cliente. Estas tabelas são designadas por tabelas de *VPN Routing and Forwarding* (VRF) [9]. Os métodos de implementação destas tabelas dependem do fornecedor. Estas tabelas são muito importantes na medida em que asseguram a apropriada segmentação do tráfego de dados e controlo.

Cada VPN irá ter a sua própria tabela VRF. A interface entre os routers PE e CE apenas pode pertencer a uma única VRF [2]. A informação de encaminhamento terá assim de ser isolada individualmente para cada VRF, evitando assim que esta informação se misture entre as várias VPNs. Isto é conseguido adicionando um identificador de 8 bytes a cada rota, de forma a identificar a VRF e consequentemente a VPN a que pertence [9].

Os identificadores de rota, presentes nas tabelas VRFs, podem ter os seguintes formatos: *ASN:nn* ou *Endereço IP: nn*, onde *nn* representa um número. O formato mais utilizado pelos operadores de rede é o *ASN:nn*, onde ASN significa *Autonomous System Number*, ou seja, é um identificador que a entidade *Internet Assigned Numbers Authority* (IANA) atribui a cada operador. O *nn* é um número, usado pelo operador da rede, para identificar a VRF. Desta forma, o identificador de cada VRF em conjunto com o IP de destino da rota permite distinguir todas as VPNs, evitando a sobreposição de endereços [8].

Os routers PE trocam informação de encaminhamento entre si de modo a estabelecer as VPNs. Para realizar esta tarefa, os routers PE correm uma extensão do protocolo de encaminhamento BGP. Todos os routers PE comunicam entre si através do *Interior* BGP (iBGP) com extensões *Multi-Protocol* (MP-BGP).

A Figura 2.12 mostra as várias etapas de propagação de uma rota para estabelecimento duma VPN. Na etapa 1, o router CE envia a rota para o router PE através de um protocolo de encaminhamento IGP ou eBGP. Se nenhum destes protocolos for usado e o encaminhamento for estático, então o router PE irá ter uma rota para a rede do site A através do router CE. De seguida, o router PE associa a essa rota o identificador de VPN referido anteriormente e insere-a na tabela VRF, completando assim a etapa 2. Na etapa 3, o router PE envia estas rotas alteradas para o router PE de

destino através do protocolo MP-BGP. O PE de destino desempenha a etapa 4 identificando a que VPN pertence a rota, com base no identificador da VRF. Na última etapa (a etapa 5), o router PE envia a rota para o cliente certo através de um dos protocolos de encaminhamento já referidos.

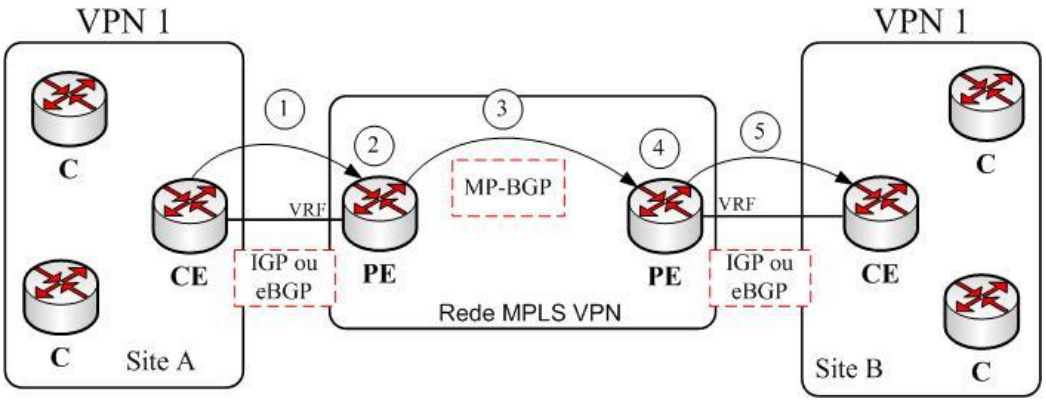


Figura 2.12 - Propagação de rotas numa rede MPLS VPN [8].

Estas VPNs são estabelecidas sobre uma rede MPLS. Como tal, no router PE são inseridas etiquetas nos pacotes e estes são encaminhados para o LSP apropriado. Poderão existir múltiplos LSPs entre dois PE. A única diferença entre os pacotes usados nestas VPNs e qualquer outro pacote MPLS é que os destas VPNs são transportados com duas etiquetas [9].

As regras básicas do MPLS permitem o uso de etiquetas múltiplas (também designadas por pilhas de etiquetas). Todas as decisões de encaminhamento são realizadas com base na etiqueta do topo da pilha. Esta etiqueta é trocada à medida que o pacote é encaminhado na rede MPLS. Nas VPNs MPLS, esta etiqueta é usada para encaminhar o pacote para o router PE de saída [9].

Assim que o pacote chega ao PE de saída, a etiqueta MPLS é removida e o pacote encaminhado para o seu destino com base nos seus endereços IP. No entanto, como referido anteriormente, numa implementação de VPNs MPLS, os endereços IP dos clientes poderão sobrepor-se. Assim, o router PE necessita de informação adicional para identificar o pacote e encaminhá-lo para o cliente apropriado. A segunda etiqueta fornece esta informação, identificando a que VPN o pacote pertence. Em resumo, a primeira etiqueta é usada para encaminhar o pacote no domínio MPLS para o router de saída e a segunda etiqueta para entregar o pacote ao cliente correcto [9].

2.7.2. L2VPN

As L2VPN são VPNs criadas ao nível da camada 2 e transportadas sobre redes de pacotes através de *pseudowires* (PW). Os PWs transportam tráfego de camada 2 sobre redes *Packet-Switched Network* (PSN). No MPLS, o PW é um túnel entre dois routers PE que transporta tramas de camada

2. Estas tramas são encapsuladas em pacotes e adicionadas etiquetas para que possam ser encaminhadas pela rede MPLS. O resultado é um serviço de camada 2 – seu funcionamento e características – a ser transportado sobre uma rede PSN [2]. Algumas das tecnologias de camada 2 que podem ser transportadas pelo MPLS são, por exemplo, o *Ethernet*, o *High-Level Data Link* (HDLC), o *Point-to-Point Protocol* (PPP), o ATM, ou o *Frame Relay*. Estes PWs são também conhecidos por *Virtual Private Wire Service* (VPWS).

A arquitectura das L2VPNs é idêntica à das VPNs BGP/MPLS. Neste caso, o router CE estabelece comunicação com outros routers CE através duma tecnologia de camada 2, desconhecendo que pelo meio possa estar uma rede MPLS. Na fronteira desta rede estão os routers PE que realizam os PWs e no núcleo estão os routers P que realizam o encaminhamento MPLS, desconhecendo a existência dos PWs [8].

Tal como nas VPNs BGP/MPLS, os pacotes MPLS que realizam o PW contém também duas etiquetas. A primeira etiqueta é usada para estabelecer o LSP entre os dois PE. Depois, de modo a suportar múltiplos PWs num LSP, é usada uma outra etiqueta que identifica o PW. Esta última etiqueta é designada por etiqueta VC ou PW, e é sempre a última na pilha de etiquetas dos pacotes [8].

Para estabelecer o PW, cada par de routers PE deve estabelecer uma sessão LDP *targeted* entre eles. Esta sessão LDP permite trocar informação sobre as características do PW e anuncia a etiqueta VC a ser usada. Estes routers PE estabelecem também uma sessão LDP com cada router P vizinho, para anunciar a primeira etiqueta que constrói o LSP. Depois do LDP estabelecer o LSP e anunciar a etiqueta VC, todos os routers estão prontos para encaminhar tráfego. A Figura 2.13 apresenta um exemplo desse encaminhamento, bem como, ao nível do plano de controlo, o facto dum LSP poder conter mais de um PW.

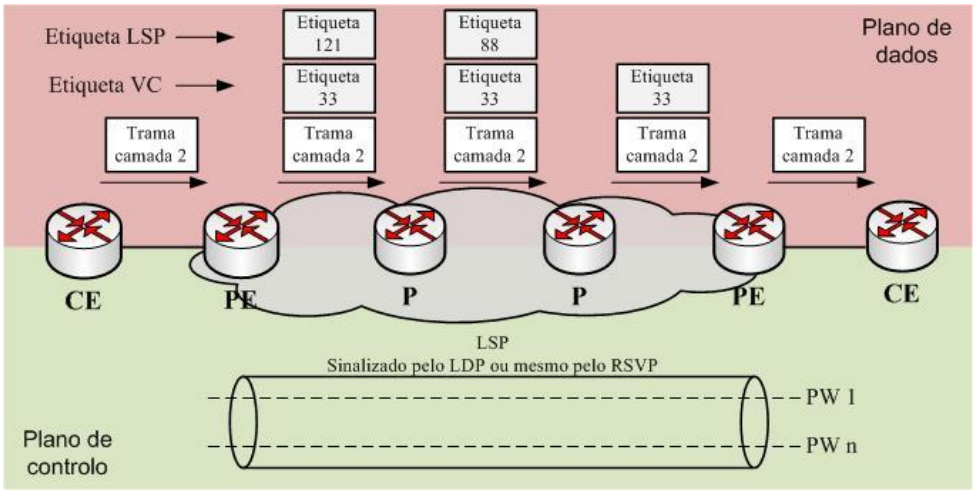


Figura 2.13 - Encaminhamento duma trama de camada 2 na rede MPLS, adaptado de [8].

Na figura, o router PE de entrada insere a etiqueta VC (etiqueta 33) na trama. Depois insere a etiqueta do LSP. Esta etiqueta é a etiqueta associada à rota, anunciada pelos protocolos de encaminhamento. De seguida, o pacote MPLS é encaminhado ao longo do LSP, nó a nó, até chegar ao PE de saída. Como se pode observar, os routers P nunca analisaram a etiqueta VC, pelo que são completamente alheios ao PW. De notar que, quando o pacote alcança o PE de saída, a etiqueta do LSP já foi removida pelo LSR anterior. Esta funcionalidade designa-se por *Penultimate hop Popping* (PHP).

O PHP é uma funcionalidade que permite aliviar a carga nos LERs de saída. Quando o penúltimo LSR de um determinado LSP tem na sua tabela LFIB a etiqueta reservada 3 (designada como etiqueta implícita), retira a primeira etiqueta de todos os pacotes do LSP, enviando de seguida todos os pacotes para o LER de saída. Assim, os pacotes chegam já ao LER sem uma etiqueta. Esta funcionalidade é útil quando os pacotes chegam ao LER de saída com várias etiquetas em *stack*. Se o penúltimo LSR retirar uma etiqueta, evita assim que o LER de saída tenha de fazer mais uma pesquisa nas suas tabelas [8].

2.7.3. VPLS

O VPLS é um serviço que permite construir uma *Local Area Network* (LAN) sobre uma rede MPLS, usando para esse fim os PW descritos na secção anterior. Usando o VPLS para interligar vários sites Ethernet sobre uma rede MPLS é semelhante a ter todos esses sites interligados por um *switch* Ethernet. Uma LAN Ethernet é uma tecnologia da camada 2. Como tal, as tramas Ethernet terão de ser transportadas pela rede MPLS. Sendo um serviço ponto-multiponto, o VPLS suporta também o transporte de tramas *broadcast* e *multicast* [8].

Para que o VPLS se assemelhe a um *switch* Ethernet, terá de ter as seguintes características: encaminhamento de tramas Ethernet, encaminhamento de tramas *unicast* com endereço *Media Access Control* (MAC) de destino desconhecido, replicação de tramas *broadcast* e *multicast* para mais do que uma porta, prevenção de *loops* e aprendizagem dinâmica de endereços MAC. Para suportar estas características, os routers PE estabelecem PWs entre eles para transportar as tramas Ethernet. Cada PW é constituído por dois LSPs, um por cada direcção [8].

O VPLS é configurado associando, no router PE, uma determinada instância VPLS a um porto ou a uma *Virtual LAN* (VLAN). Desta forma, por exemplo, as tramas com endereço MAC de destino desconhecido serão encaminhadas para todos os portos e PWs que pertencem a essa instância [2].

Tal como no L2VPN, no VPLS a trama Ethernet é encapsulada com duas etiquetas no MPLS. A primeira etiqueta identifica o LSP e a segunda (etiqueta VC) refere-se ao porto Ethernet ou à interface VLAN para a qual a trama Ethernet deve ser encaminhada [8].

Pelo que foi dito anteriormente, o VPLS requer obviamente múltiplos PWs entre routers PE para cada instância VPLS. Na configuração de um router PE devem ser especificados todos os routers PE remotos numa determinada instância VPLS. Com base nesta configuração, os routers PE estabelecem sessões LDP *targeted*, anunciando as etiquetas para cada PW [8].

2.8. Diferenciação de serviços (*DiffServ*)

A entidade IETF designou duas formas de implementar QoS numa rede IP: Serviços Integrados (*IntServ*) e Serviços Diferenciados (*DiffServ*). O *IntServ* usa o protocolo de sinalização RSVP. O *DiffServ* faz uso dos bits de tipo de serviço – em inglês, *Type of Service* (TOS) – existentes no cabeçalho IP para qualificar o pacote IP de uma determinada QoS. Os routers olham para estes bits para priorizar os pacotes. Uma das diferenças fundamentais entre o *DiffServ* e o *IntServ* é o facto de o *DiffServ* não usar nenhum protocolo de sinalização.

De acordo com [10], os seis bits mais significativos do campo TOS do cabeçalho IP são usados para realizar a diferenciação de serviços. Como se pode observar pela Figura 2.14, estes seis bits são designados por *Differentiated Services Codepoint* (DSCP). Os dois bits menos significativos não são usados – *Current Unused* (CU). Dentro dos bits DSCP, os três bits mais significativos definam a classe de serviço, os dois bits a seguir definem o nível de descarte e o último bit é reservado. Quanto maior for o nível de descarte dentro de uma classe de serviço, maior é a probabilidade de o pacote ser descartado, em relação a outros pacotes com inferior nível de descarte em caso de congestão.

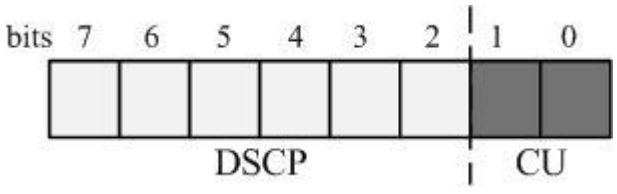


Figura 2.14 - Estruturação do byte TOS para diferenciação de serviços [10].

No cabeçalho MPLS existe o campo EXP (ver Figura 2.1). Este campo pode ser usado para implementar QoS nas redes MPLS. Estes bits podem ser usados da mesma forma que os três bits que definem a classe de serviço do DSPC. Se for este o método usado para implementar QoS no MPLS, os LSPs podem ser designados por E-LSPs [11].

No entanto, no MPLS existe outra forma de implementar QoS. A etiqueta usada para encaminhar tráfego pode ser usada para realizar diferenciação de serviços. Ou seja, um LSP teria associada várias etiquetas, uma para cada classe de serviço. Assim, o protocolo de sinalização teria de ser capaz de sinalizar etiquetas diferentes para o mesmo LSP. Este LSP é designado por L-LSP, indicando que a etiqueta faz parte do processo de QoS [11].

Capítulo 3: Requisitos para o *backhaul* de redes móveis

O *backhaul* de redes móveis é a rede de transporte que fornece conectividade entre o acesso rádio (*cell sites*) e os seus elementos controladores correspondentes, localizados no interior do núcleo da rede [12]. A Figura 3.1 apresenta a topologia habitual do *backhaul* de uma rede móvel.

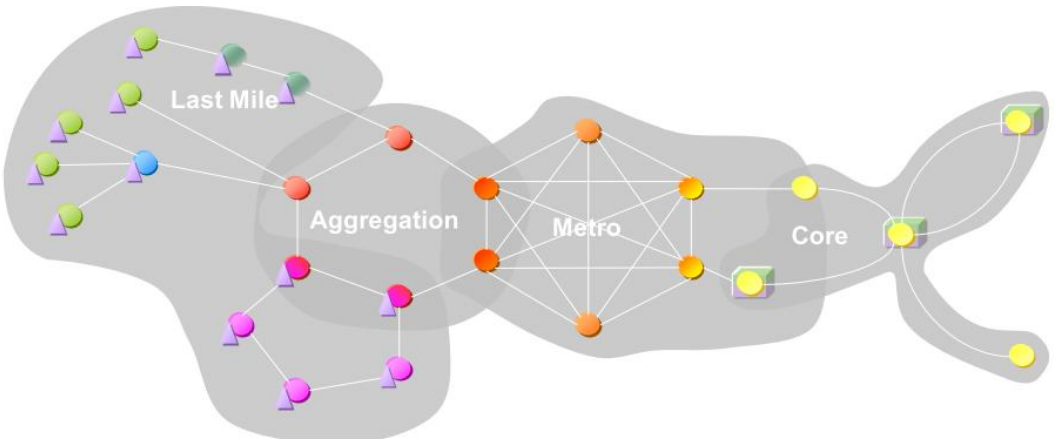


Figura 3.1 - Topologia do *backhaul* de uma rede móvel [12].

Nesta figura, a rede de *backhaul* estende-se desde o acesso rádio, passando pelo domínio "*Last Mile*", agregação (*Aggregation*), metro e terminando à entrada do domínio do núcleo (*Core*) da rede. Entre cada domínio residem os nós de transporte que fornecem capacidades de gestão de tráfego, bem como de comutação e monitorização do desempenho. A rede de *backhaul* pode ser constituída por uma variedade de tecnologias de transmissão físicas incluindo a fibra óptica, micro-ondas rádio, cabo *Digital Subscriber Line* (DSL) e, ocasionalmente, satélite. Existe mais variedade de meios de transmissão nos domínios "*Last Mile*" e agregação do que nas redes de metro e núcleo, que predominantemente utilizam transmissões ópticas de alta capacidade como o *Wavelength-Division Multiplexing* (WDM) [12].

As redes de metro e agregação têm também diferentes tipologias. Hoje em dia, as redes metro transportam elevados volumes de tráfego agregado e, naturalmente, precisam de mais capacidade

de transmissão. As redes de metro estão mais preparadas para a adopção de tipologias mistas e tecnologias por pacotes. Por outro lado, as redes de agregação têm mais diversidade. Estas redes são constituídas por uma mistura de ATM/*Synchronous Digital Hierarchy* (SDH) e Ethernet, suportando topologias em anel e serviços multi-camada [12]. Como veremos neste capítulo, a evolução das redes móveis trarão algumas alterações a esta topologia.

3.1. Evolução e convergência das redes móveis

Os operadores de redes móveis ambicionam fornecer cada vez mais serviços de banda larga, como a televisão ou vídeo, com a mesma qualidade que os operadores de redes fixas. Actualmente muitos dos operadores móveis fornecem serviços centrados na voz (2G/3G). Apesar de estes serviços continuarem a crescer firmemente, as receitas por eles gerados não crescem na mesma proporção. Desta forma, muitos operadores estão a planear o crescimento das suas receitas com base em serviços de dados, ao mesmo tempo que continuam a fornecer os seus serviços de voz [13].

O desenvolvimento de tais serviços de banda larga terá, também, um impacto no *backhaul* das redes. Os actuais serviços 2G e 3G usam as tecnologias TDM e ATM, respectivamente. O tráfego TDM gerado pelas *Base Transceiver Stations* (BTSs) 2G e o tráfego ATM dos Nós B 3G são multiplexados e transportados em redes de *backhaul* TDM, i.e., a plataforma de transporte TDM está a ser usada como plataforma multi-serviço para serviços de voz. Por outro lado, os serviços de dados estão a ser desenvolvidos sobre plataformas de transporte por pacotes, usando o Ethernet, MPLS e o IP, porque os serviços de banda larga são baseados em tecnologias de pacotes, que fornecem uma solução para os problemas emergentes de custo e dimensionamento. Isto significa que a instalação destes serviços de banda larga requer novas redes de *backhaul* baseadas em pacotes. Como tal, os novos *backhauls* necessitam uma nova plataforma flexível que transporte pacotes e também TDM [13].

A Figura 3.2 mostra um *roadmap* para a disponibilização das várias tecnologias nas redes móveis. As tecnologias 2G estão já em operação mas, no entanto, a quantidade dos seus utilizadores tem tendência para diminuir à medida que outros serviços vão sendo oferecidos. Contudo, as tecnologias 2G deverão continuar a ser suportadas até todos os seus utilizadores mudarem para o 3G ou tecnologias posteriores. As tecnologias 3G já estão também disponíveis e gradualmente a evoluir para uma base mais estável. As tecnologias WiMAX estão já disponíveis para instalação. O *Ultra Mobile Broadband* (UMB) – uma evolução do *Code Division Multiple Access* 2000 (CDMA2000) – e o LTE iniciarão a sua instalação em 2010. Num futuro próximo, o 3G baseado em IP, o WiMAX, o LTE e o UMB crescerão e tornar-se-ão as tecnologias dominantes [13].

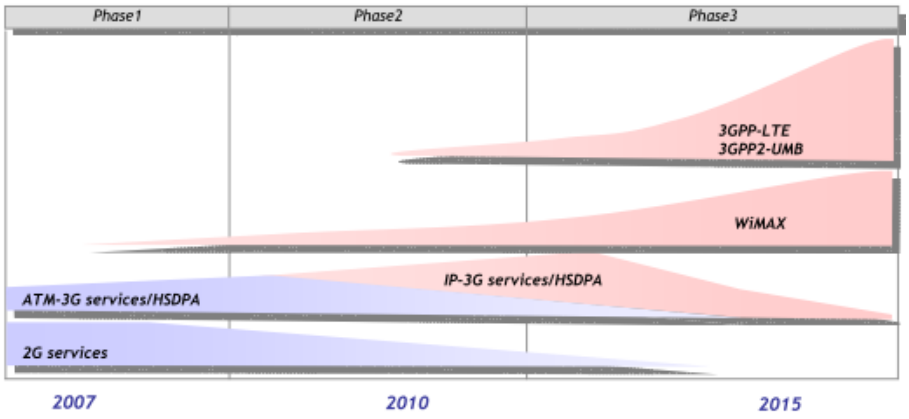


Figura 3.2 - *Roadmap* para a disponibilização das tecnologias móveis [13].

A Figura 3.2 mostra também que o *roadmap* está dividido em 3 fases. Na fase 1 (*Phase 1*) são fornecidos o 2G/3G. Na fase 2 (*Phase 2*) estão disponíveis os recentes serviços 3G/HSDPA baseados em IP. Na fase 3 (*Phase 3*), o LTE e o UMB estão a crescer.

Em linha com o *roadmap* apresentado por [13], está a evolução do *backhaul*, que transitará do TDM para as redes de pacotes. A Figura 3.3 mostra a evolução do *backhaul* ao longo das três fases.

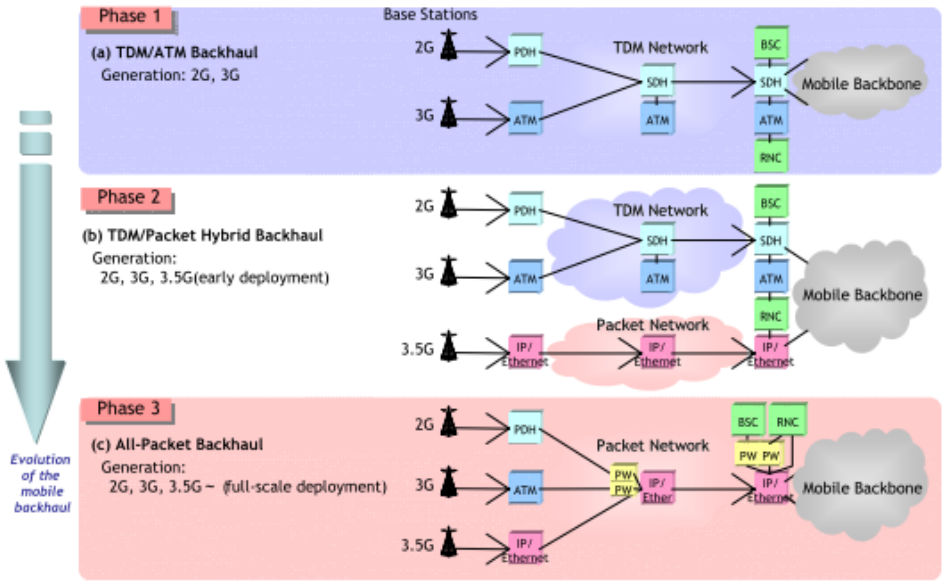


Figura 3.3 – Evolução da arquitectura do *backhaul* de redes móveis [13].

Na fase 1 é utilizado o *backhaul* TDM que suporta as actuais tecnologias 2G/3G (Figura 3.3 (a)). Estes *backhauls* TDM foram largamente adoptados para suportar tanto o TDM como o ATM.

Na fase 2, os *backhauls* são alterados para uma versão híbrida TDM/pacotes (Figura 3.3 (b)). Este *backhaul* é a transição para um *backhaul* puramente de pacotes. À medida que os serviços de banda

larga vão crescendo, maiores serão os *backhauls* de pacotes, deixando para trás os TDM. Nesta fase, a actual plataforma TDM será gradualmente mudada para a nova plataforma de pacotes.

Na fase 3, quando os serviços de banda larga baseados em pacotes se tornarem dominantes, o *backhaul* terá uma arquitectura puramente baseada em pacotes (Figura 3.3 (c)). Nesta arquitectura, os serviços TDM/ATM e os serviços de pacotes serão agregados numa única plataforma de pacotes. Na figura pode-se observar que a passagem de TDM para pacotes é realizada recorrendo ao estabelecimento de PWs. Esta parte será abordada nas secções seguintes.

Os novos *backhauls* terão assim de cumprir com três grandes objectivos: deverá ser flexível (para suportar ambos os serviços descontinuados, como o TDM, e o IP), escalável, (para suportar as tecnologias emergentes) e de baixo custo (para compensar os aumentos de tráfego) [14].

3.2. IP/MPLS em redes móveis

Os operadores de redes móveis reconhecem cada vez mais o IP/MPLS como a melhor solução estratégica para os seus *backhauls*. Apenas o IP/MPLS tem a combinação de custo, escalabilidade e flexibilidade que os operadores necessitam para, por um lado continuar a aproveitar os investimentos existentes, por outro aumentar a capacidade das suas redes para colmatar o aumento de tráfego de dados [15]. As características que tornam o IP/MPLS uma solução desejável são [16]:

- Suporta o transporte de uma elevada gama de serviços de camada 2 e 3, incluindo o TDM, ATM, HDLC e IP, sendo assim possível a migração das habituais redes TDM e ATM para redes baseadas em IP.
- Permite recuperação de falhas e disponibiliza funções *Operation, Administration and Maintenance* (OAM) que podem ser usadas para garantir a segurança da rede de *backhaul*.
- Fornece engenharia de tráfego (TE) e capacidades de QoS que permitem uma melhor gestão dos recursos na rede de transporte, maximizando a utilização da infra-estrutura da rede.
- Permite que fornecedores possam disponibilizar serviços de acesso/agregação a um conjunto diferente de operadores móveis, usando tecnologias antigas e emergentes sobre uma rede convergente.
- Pode funcionar sobre várias redes de transporte, incluindo o SDH, *Plesiochronous Digital Hierarchy* (PDH) e Ethernet.
- Contém um plano de controlo que facilita o encaminhamento do tráfego.

- Fornece um conjunto abrangente de mecanismos de protecção e restabelecimento.

Além disso, o IP/MPLS é uma plataforma preparada para receber a migração para as novas tecnologias como o HSPA e o 4G/LTE. A transição para o IP nos *backhaul* está a ser realizada para suportar estas tecnologias emergentes. Desta forma, a mudança para o IP/MPLS é apenas uma extensão lógica de uma tecnologia já em uso em muitas redes (como é o IP) [15].

De notar também que o IP/MPLS pode atender à necessidade de se ter múltiplas tecnologias no acesso rádio. Pela sua natureza, o MPLS é uma tecnologia agregadora, permitindo que diversas tecnologias, como o TDM, ATM e Ethernet, possam ser transportados numa única infra-estrutura. A informação TDM – que é predominantemente tráfego de voz – pode ser transportada sobre o IP/MPLS utilizando PWs [15] (como referido no capítulo anterior, os PWs emulam circuitos sobre uma infra-estrutura baseada em pacotes).

As redes SONET/SDH, até agora usadas no transporte de redes móveis, são conhecidas pela sua elevada segurança e rápida recuperação em caso de falha. Como tal, o IP/MPLS terá de garantir o mesmo nível de recuperação destas redes. Características como o FRR, que utiliza o protocolo RSVP-TE, permitem esse nível de recuperação nas redes MPLS. Quando ocorre uma falha, a rede, usando o FRR, pode rapidamente comutar para o caminho protector, evitando o atraso associado com a actualização das tabelas de encaminhamento. A combinação do FRR com o cuidadoso planeamento dos LSPs primários e secundários permite a uma rede IP/MPLS recuperações da ordem das dezenas de milissegundos [15].

A Figura 3.4 ilustra uma abordagem ao transporte integrado das diversas tecnologias móveis usando o IP/MPLS. Nesta abordagem, uma única rede IP/MPLS transporta voz e dados simultaneamente. O tráfego de voz é transmitido desde o acesso rádio através de PWs e depois convertido novamente para TDM/ATM na central de agregação. O tráfego 4G, que já vem em IP, é transportado mesma rede MPLS. Esta abordagem permite assim que todo o tráfego – de voz e dados – aproveite as vantagens do IP/MPLS e ainda preserva os investimentos realizados no acesso rádio e núcleo da rede.

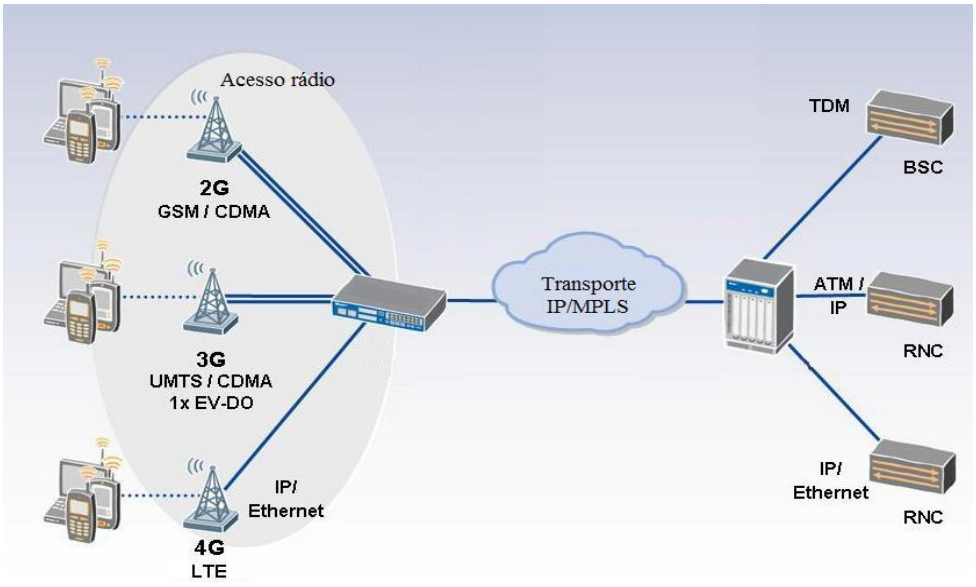


Figura 3.4 - Agregação das diversas tecnologias usando o IP/MPLS, adaptado de [15]

3.3. Requisitos para redes móveis

O *Broadband Forum* é um consórcio global que visa a elaboração de especificações para a próxima geração de redes IP. Um dos grandes objectivos deste consórcio passa por conduzir a evolução das próximas gerações de redes, possibilitando a convergência das redes fixas e móveis [17]. Uma das suas iniciativas é designada por *MPLS Mobile Backhaul Initiative* (MMBI), iniciativa esta que tem como objectivo propor um conjunto de ferramentas para o uso da tecnologia MPLS nas novas gerações de redes móveis [14].

O MMBI define vários cenários de desenvolvimento e fornece recomendações para a implementação do MPLS em cada um destes cenários. As arquitecturas de rede sugeridas são definidas para várias redes de transporte e gerações de redes móveis. Estes cenários são agrupados de acordo com a Tabela 3.1 e divididos em duas categorias básicas: descontinuados (TDM, ATM, HDLC) e futuros (IP/Ethernet) [14].

Tecnologia	Rede de transporte	Débitos (aproximados)
GSM/GPRS (2G/2.5G)	TDM	56 – 114 Kbps
EDGE (2.5G)	TDM	236.8 – 473.6 Kbps
UMTS/HSDPA (3G)	ATM	~384 Kbps (*uplink*) ~2 a 3.1 Mbps (*downlink*)

CDMA 1xRTT (2.5G)	HDLC ou TDM	144 Kbps
CDMA 1x EV-DO (3G)	IP	~1.8 Mbps (*uplink*) ~3.1 Mbps (*downlink*)
WiMAX	IP	50 Mbps
LTE (4G)	IP	> 50 Mbps (*uplink*) > 100 Mbps (*downlink*)

Tabela 3.1 - Tecnologias rádio, rede de transporte e débitos [14].

De acordo com a tabela anterior, os serviços descontinuados têm a sua rede de transporte baseada em TDM ou ATM sobre interfaces T1/E1. Por sua vez, o CDMA 1xRTT faz uso do HDLC. No futuro, os serviços LTE e WiMAX serão suportados por IP.

Para o transporte de todas as tecnologias anteriores, o MMBI define requisitos gerais, necessários para a implementação do MPLS em redes móveis. Estes requisitos foram o ponto de partida para o trabalho desta dissertação e são eles [16]:

- A rede de transporte deve ser capaz de suportar MPLS baseado em redes de comutação por pacotes IP.
- A rede IP/MPLS deve suportar características de QoS/TE.
- Os equipamentos PE devem suportar sinalização PW e capacidades de encapsulamento.
- Os LSPs podem ser estabelecidos manualmente ou através de sinalização, usando o RSVP-TE ou LDP.
- O LSP pode opcionalmente suportar capacidades de OAM.

Antes de estabelecer os LSPs e realizar o encaminhamento dos pacotes com base em comutação de etiquetas, as redes IP/MPLS devem ser essencialmente IP. Além das vantagens MPLS que possuem, estas redes deverão ter todas as propriedades do encaminhamento IP.

De acordo com [16], as redes MPLS (e, consequentemente, os equipamentos que a constituem) têm de cumprir com as especificações RFC 3031 e RFC 3032 para os túneis MPLS (LSPs), e RFC 3985 para o estabelecimento de PWs. Para os casos do TDM, ATM ou HDLC, o transporte deverá ser feito com recurso às L2VPNs, como as VPWSs e as VPLSs. Para o caso do IP/Ethernet, o transporte deverá ser feito também através das L2VPNs ou então recorrendo às MPLS/BGP VPNs de camada 3.

Os túneis MPLS podem ser estabelecidos manualmente ou dinamicamente através da sinalização. A sinalização deverá estar de acordo com a RFC 4447 para estabelecer e gerir PWs. Para os túneis

MPLS sem TE, os routers MPLS deverão suportar o LDP. O uso da sinalização RSVP-TE é opcional [16].

Na generalidade, se acontecer uma falha, é preciso detectá-la, diagnosticá-la, localizá-la, notificar as entidades apropriadas e tomar as acções correctivas apropriadas ao tipo de falha. No MPLS, o OAM deverá operar *in-band* com o objectivo de detectar essas falhas nos LSPs, quer o plano de controlo seja dinâmico ou não. Para os LSPs, os routers MPLS deverão suportar o LSP Ping de acordo com RFC 4379. Para os PWs, deverá ser suportado o *Virtual Circuit Connection Verification-Ping* (VCCV-Ping) de acordo com a RFC 5085 [16]. Estas duas funcionalidades servem para testar as ligações usadas nos LSPs, permitindo descobrir eventuais falhas que o plano de controlo não encontre.

As redes MPLS também deverão suportar *DiffServ*, permitindo a distinção do tráfego por classes e habilitando assim o QoS nestas redes [16].

No caso de ser usado o RSVP-TE como protocolo de sinalização, o mecanismo de protecção de LSPs a ser usado será o FRR [16].

3.4. Requisitos do software

O objectivo deste trabalho passa pela avaliação duma solução MPLS para utilização em equipamentos da PT Inovação. A PT Inovação, por sua vez, já possui meios para que o plano de dados do MPLS possa ser implementado, quer ao nível das plataformas *MPLS-Transport Profile* (MPLS-TP) já desenvolvidas, quer ao nível de processadores de rede, como os *chips* da *Wintegra* actualmente usados nas plataformas ATM da PT Inovação.

A *Wintegra* é um fornecedor de semicondutores de processamento e software para infra-estruturas de telecomunicações. Um dos produtos deste fornecedor é o *chip Winpath* que, na sua terceira versão, suporta diversos protocolos da camada 2 e 3, tais como o Ethernet, IP, MPLS, etc [18]. Este fabricante possui também um módulo de software, designado por *Data Path Software* (DPS), que permite gerir e configurar o *Winpath*. Tendo em conta o MPLS, o *Winpath* permite [19]:

- Funções de LSR:
    - Encaminhamento de pacotes com base na etiqueta de entrada.
    - Comutação de etiquetas.
    - Verificação e alteração do campo TTL do cabeçalho MPLS.
    - Inserir e retirar novas etiquetas na *stack*.
    - Suporte do campo EXP para fornecer classes de serviço.

- Funções LER:
    - Suporte do encaminhamento de camada 3 para determinação do FEC.
    - Suporte de mapeamento directo de camada 2 para determinar o FEC (por exemplo, FEC para endereço VPI/VCI no ATM, endereço MAC no Ethernet, VLAN *tag*).
    - Introdução de etiquetas e encaminhamento com base na FEC.
    - Remoção de etiquetas.
    - Criação/remoção do campo TTL no cabeçalho MPLS.

Estas funcionalidades estão relacionadas com o plano de dados do MPLS. São elas que permitem suportar as tabelas FIB e LFIB referidas anteriormente. As associações etiqueta-FEC terão assim de ser configuradas manualmente. Desta forma, a PT Inovação procura uma forma de suportar nos seus equipamentos a gestão e distribuição automática de etiquetas, ou seja, software que possa realizar a sinalização MPLS incorporada no plano de controlo.

Para analisar as alternativas existentes no mercado, foram definidos requisitos básicos que a solução de software teria cumprir. Estes requisitos foram definidos com base nos requisitos para o *backhaul* de redes móveis e nos objectivos da PT Inovação. Além dos requisitos definidos na Tabela 3.2, o software deverá ser capaz de correr em plataformas Linux, sendo este o sistema operativo utilizado pela PT Inovação na maioria dos seus equipamentos.

Requisitos para a solução MPLS em Linux	
Sinalização MPLS	Encaminhamento IP
Sinalização LDP	Encaminhamento dinâmico IGP (OSPF e IS-IS)
MPLS-TE (RSVP-TE)	Encaminhamento dinâmico EGP (BGPv4, MP-BGP)
Funções MLPS LER	OSPF-TE
Protecção MPLS 1:1 LSP	IPv6
L2VPN (VPWS + VPLS)	
MPLS/BGP VPN	
MPLS FRR	
MPLS OAM	

Tabela 3.2 - Requisitos para o software.

Os requisitos definidos na tabela anterior estão divididos em dois grupos que se complementam: os requisitos para a sinalização MPLS e os requisitos para o encaminhamento IP. Uma vez que as etiquetas são distribuídas dinamicamente através dos protocolos de sinalização MPLS, faz sentido que a solução suporte também encaminhamento IP dinâmico.

Dentro dos requisitos para a sinalização MPLS estão os dois protocolos de distribuição de etiquetas: o LDP e o RSVP-TE. Para o LDP poder funcionar, o software terá de suportar um ou mais protocolos de encaminhamento dinâmico IP, tal como o OSPF, IS-IS e BGPv4. O OSPF-TE também é necessário para suportar as extensões exigidas pelo RSVP-TE. A protecção do LSP 1:1 e o FRR estão relacionadas com o RSVP-TE e têm de ser suportadas pelo software.

As funções MPLS LER são todas as funções realizadas por um equipamento que esteja na fronteira de uma rede MPLS, ou seja, o software tem de suportar a transição de uma rede MPLS para uma rede não MPLS, e vice-versa.

O estabelecimento de PWs também tem de ser suportado, através das VPWSs e VPLS. Para suportar as MPLS/BGP VPNs, o software terá também de correr o MP-BGP.

Por último, as funções OAM são importantes na medida em que possibilitam detectar mais facilmente as falhas na rede. Uma das funções OAM que o software deverá suportar deverá ser o LSP *ping*. O IPv6 deverá ser uma opção a ter em conta na escolha de software, visto ser o futuro das redes IP. Ou seja, todos os protocolos anteriores terão de suportar o IPv6.

3.5. Análise das *stacks* MPLS disponíveis

De acordo com os requisitos definidos anteriormente, foi realizada uma pesquisa das *stacks* de software MPLS disponíveis no mercado. Esta pesquisa foi realizada com base na informação disponibilizada na Internet pelos fornecedores.

3.5.1. Projecto “MPLS for Linux”

O projecto “MPLS for Linux” tem como objectivo desenvolver um conjunto de protocolos de sinalização e um plano de encaminhamento para o MPLS em código aberto - sob a licença *General Public License* (GPL) - e a ser suportado em sistemas Linux. À data da realização da pesquisa, o software disponível já suporta um plano de encaminhamento MPLS para o *kernel* 2.6.x, bem como uma implementação do LDP [20].

O plano de encaminhamento suportado por este software realiza todas as funções associadas ao plano de dados do MPLS. Entre essas funções estão [20]:

- Interfaces Ethernet;
- Interfaces PPP;
- *Label Spaces* por interface e por plataforma;
- *Stacking* de etiquetas (associado ao LDP permite ter VPNs);
- Procura recursiva de etiquetas;
- Uma entrada na tabela de encaminhamento do Linux (FEC) pode ter uma etiqueta associada;
- Integração com o modelo de QoS do Linux;
- Diferenciação de serviços (*DiffServ*) – E-LSPs e L-LSPs;
- Ethernet sobre MPLS (usando as *ebtabels*);
- *Penultimate Hop Popping* (PHP).

Estas funcionalidades estão integradas dentro do *kernel* do Linux e utilizam alguns dos seus recursos. O suporte de interfaces Ethernet e PPP, a integração com o QoS do Linux, o *Diffserv* e o Ethernet sobre MPLS (que permite as VPLS) são funcionalidades já existentes dentro do *kernel* e adaptadas ao MPLS. Além disso, este software possui também *stack* de etiquetas para as VPNs e procura recursiva de etiquetas para pesquisas rápidas. Suporta também a possibilidade de se associar uma etiqueta a cada rota que exista na tabela de encaminhamento do Linux.

A implementação LDP, suportada também no "MPLS for Linux", tem actualmente as seguintes características [20]:

- Modos de distribuição de *downstream on demand* e *unsolicited*;
- Descoberta básica e extensiva;
- Distribuição de etiquetas controlada por policiamento;
- Integrado na plataforma de encaminhamento *Quagga*.
- *Application Programming Interface* (API) de configuração flexível.

As características básicas do LDP são suportadas por este software. Além disso, está integrado numa plataforma de software em código aberto, designada por *Quagga*, que suporta vários protocolos de encaminhamento dinâmico IP. Esta implementação permite a troca de mensagens LDP com outros LSRs para construir os LSPs. As etiquetas são depois entregues ao módulo de *kernel*, referido anteriormente, para os cruzamentos MPLS.

De acordo com os objectivos da PT Inovação, interessa apenas avaliar o módulo LDP em conjunto com o *Quagga*. Embora muitos dos requisitos inicialmente definidos não sejam suportados, esta

implementação permite a distribuição de rotas IP e etiquetas MPLS. Desta forma, dado que é um software de utilização grátis, é um bom ponto de partida para avaliar a exequibilidade da solução pretendida pela PT Inovação.

3.5.2. Metaswitch

Metaswitch é um fornecedor de sistemas de transporte e soluções de software para arquitecturas baseadas em comutação por pacotes. Este fornecedor possui um produto designado por plano de controlo integrado que pode ser usado para implementar o plano de controlo do MPLS e encaminhamento IP numa vasta gama de equipamentos de rede [21]. A Figura 3.5 mostra a arquitectura do plano de controlo integrado.

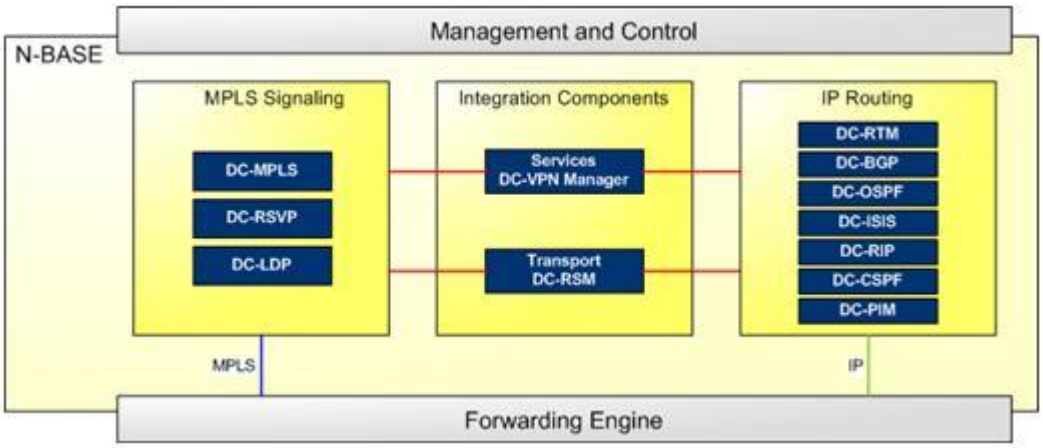


Figura 3.5 - Arquitectura do plano de controlo integrado da *Metaswitch* [21].

Esta arquitectura é constituída por dois componentes principais: o DC-MPLS e o IP *routing*. O módulo DC-MPLS contém todos os protocolos de sinalização MPLS e, o IP *routing* suporta todos os protocolos mais importantes de encaminhamentos IP. Para além destes componentes, o módulo DC-RSM (*Route Selection Manager*) é usado no transporte do MPLS e estabelece a ligação entre o DC-MPLS e os protocolos de encaminhamento (DC-OSPF, DC-ISIS, …) permitindo o cálculo automático de uma rota através da rede. O DC-RSM suporta muitas aplicações para comutação por pacotes, incluindo o tradicional IP/MPLS, MPLS-TP e OTN (*Optical Transport Network*).

O plano de controlo integrado possui também um plano de controlo para MPLS/BGP VPNs, designado por DC-VPN *Manager*. Este módulo é uma extensão ao DC-BGP, constituído por software de encaminhamento e encaminhamento VPN, que possibilita a implementação de VPNs com MP-BGP/MPLS.

O DC-MPLS contém vários módulos de software que implementam os vários protocolos de sinalização. O diagrama de blocos da Figura 3.6 mostra a arquitectura de software de alto nível deste componente.

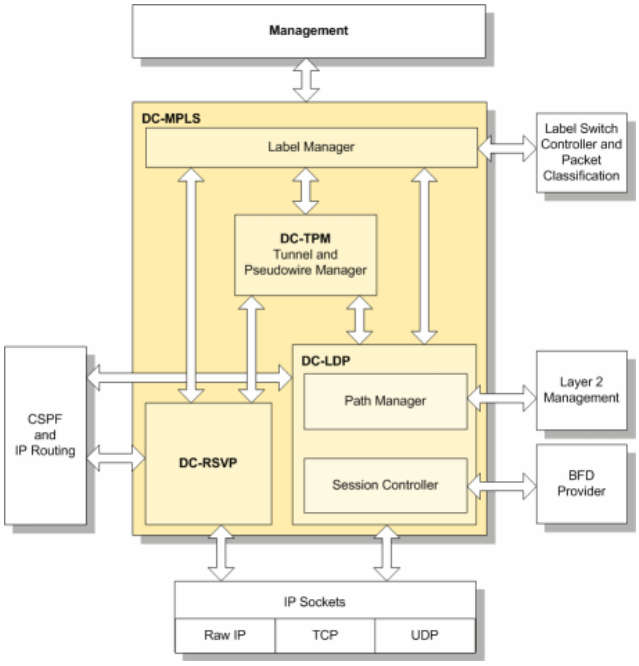


Figura 3.6 - Arquitectura do DC-MPLS da *Metaswitch* [21].

Na figura anterior são identificáveis os módulos DC-LDP, DC-RSVP e DC-TPM. Os módulos DC-LDP e DC-RSVP encontram-se ao mesmo nível e trocam mensagens com outros LSRs através de *sockets* IP. Estes dois módulos trocam informação com o módulo *Label Manager* para gerir as etiquetas transmitidas e recebidas. O módulo DC-TPM permite gerir as etiquetas para o estabelecimento de PWs. As características de alguns dos módulos disponíveis são as seguintes.

DC-LDP: inclui as funções do protocolo LDP.

- LDP;
- Suporta IPv4 e IPv6;
- Suporta todas as combinações dos modos de gestão de etiquetas:
- Modos de distribuição *downstream unsolicited* e *downstream on demand*;
- Modos de retenção liberal e conservativo;
- Controlo ordenado e independente;
- Detecção de *loops*;
- L2VPNs: arquitectura PWE3, VPWS, VPLS;
- MPLS/BGP VPNs;
- LSP ping.

DC-RSVP: inclui as funções do protocolo RSVP-TE.

- Suporte total do RSVP e as suas extensões de TE;
- Protecção de *links*;
- FRR;
- Suporta IPv4 e IPv6;
- *Diffserv*;

DC-TPM: gestão do MPLS-TE, túneis e PWs, inclui uma interface de gestão para estabelecer LSPs em pontos de entrada e para responder a pedidos de estabelecimentos de LSP em pontos de saída.

Este fornecedor também tem disponíveis módulos de software que implementam os vários protocolos encaminhamento IP: DC-BGP, DC-ISIS, DC-OSPF, DC-RIP, DC-CSPF. Além disto, tem também o módulo DC-RTM que agrupa toda a informação proveniente destes protocolos de encaminhamento com o objectivo de gerar uma tabela integrada de encaminhamento IPv4 e IPv6.

Esta solução suporta todos os requisitos definidos.

3.5.3. IP Infusion

IP Infusion é um fornecedor de soluções inteligentes de software para serviços IP. Este fornecedor tem disponível uma plataforma de software designada por *ZebOS Network Platform*, que suporta soluções em software dos protocolos de camada 2 e 3. Este software tem uma arquitectura modular, em que todos os módulos podem ser independentemente licenciados, instalados e actualizados. Esta plataforma é independente do hardware ou sistema operativo usado.

A Figura 3.7 apresenta o diagrama de blocos da arquitectura de software ZebOS, onde se insere a plataforma *ZebOS Network Platform*.

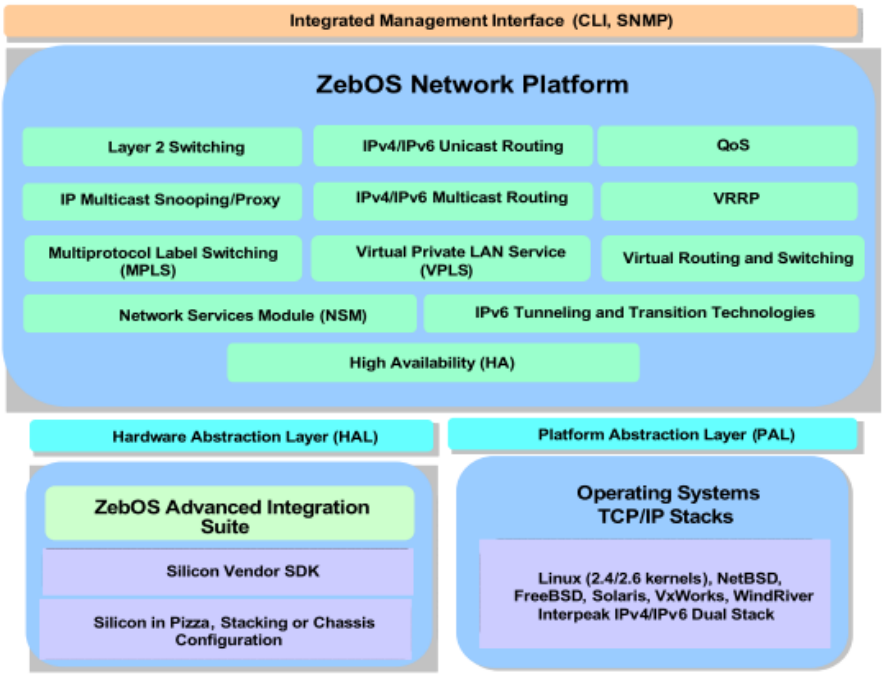


Figura 3.7 - Diagrama de blocos da *ZebOS Network Plataform* da *IP Infusion* [22].

Dentro da plataforma *ZebOS Network Platform* cada módulo é construído sobre o *Network Services Module* (NSM), o módulo base que simultaneamente e independentemente comunica com todos os processos ZebOS de *routing* e *switching*. O NSM gere tanto a tabela de encaminhamento como cada um dos protocolos activos; executa tarefas de conversão e redistribuição de rotas; e gere o estado de cada interface, policiamento de rotas e tarefas de filtragem. O NSM comunica também através do módulo *Plataform Abstraction Layer* (PAL) com as camadas inferiores, como o sistema operativo, ou através do módulo *Hardware Abstraction Layer* (HAL) para comunicar com o processador de rede para actualizar tabelas de encaminhamento.

Dentro desta plataforma existe um bloco que agrega vários módulos de software, que suportam os vários protocolos de sinalização MPLS. Os módulos disponíveis e as suas características são [22]:

- **ZebOS LDP**: suporta todas as funcionalidades do LDP, incluindo a criação e manutenção de PWs Ethernet. Todas as entradas de encaminhamento MPLS criadas por este módulo são fornecidas ao NSM. Este, por sua vez, comunica com o encaminhador MPLS através de APIs predefinidas, as quais podem ser adaptadas para qualquer hardware. Em conjunto com o ZebOS BGP-4 é possível suportar MPLS/BGP VPNs.
- **LDP6:** suporta grande parte das funcionalidades do LDP para redes IPv6.
- **RSVP-TE:** fornece informação de sinalização para TE em redes MPLS. Isto é realizado pelo protocolo RSVP-TE. Algumas das características deste módulo são: *RSVP-TE Hellos*,

Explicit Route Object, *Record Route Object*, detecção de *loops*, FRR e TE para LSPs, prioridades e caminhos *Pre-emption*, protecção LSP 1:1.

- **RSVP6-TE:** suporta grande parte das funcionalidades do RSVP-TE para redes IPv6.
- **ATM/TDM over MPLS:** suporte de ATM sobre MPLS e TDM sobre MPLS através do estabelecimento de L2VPNs. Para funcionar necessita do módulo LDP.
- **MPLS Layer 2 Virtual Circuit (VC):** permite a extensão de LANs sobre uma rede MPLS. As tramas Ethernet são inicialmente encapsuladas com uma etiqueta de circuito virtual e enviadas depois sobre um túnel LSP. Para funcionar necessita do módulo LDP.
- **VPLS:** este módulo corre sobre o MPLS Layer 2 VC de modo a fornecer uma solução flexível e escalável multiponto-a-multiponto L2VPN. O módulo VPLS vem aumentar as possibilidades do MPLS Layer 2 VC, adicionando o suporte de VPN de topologia mista. Este módulo em conjunto com o MPLS Layer 2 VC permite que vários sites possam ser ligados sobre uma rede MPLS, partilhando um domínio comum. Para funcionar necessita do módulo LDP.

Além destes módulos, este fornecedor suporta também os protocolos de encaminhamento, integrados igualmente dentro do *ZebOS Network Platform*. Os protocolos suportados são: BGP-4+, IS-IS, OSPF V2, OSPF V3, RIP e RIPng.

Esta solução suporta todos os requisitos definidos.

3.6. Stack MPLS seleccionada

A *stack* de software escolhida foi a "MPLS for Linux". Apesar de esta solução não cumprir com alguns dos requisitos definidos, foi a solução escolhida devido aos seguintes aspectos:

- Cumpre os requisitos mínimos para estabelecer LSPs através do LDP;
- Suporta alguns dos protocolos de encaminhamento IP mais usados;
- *Driver* MPLS incorporado no *kernel* que permite testar os cruzamentos MPLS efectuados;
- Software grátis, em código aberto e em constante actualização;
- Suporte do RSVP-TE previsto no *roadmap*;
- Pode ser instalado em qualquer máquina Linux;
- Módulo LDP comunica com o *kernel* para estabelecer os cruzamentos MPLS, mas está preparado para comunicar com qualquer outra entidade.

Este último motivo é muito importante, uma vez que permite integrar o módulo LDP com o software usado nos equipamentos da PT Inovação.

Como referido anteriormente, esta solução é composta por dois módulos distintos: o plano de dados do MPLS incorporado no *kernel* e o plano de controlo LDP incorporado na plataforma *Quagga*. A interacção entre estes dois módulos está exemplificada pela Figura 3.8.

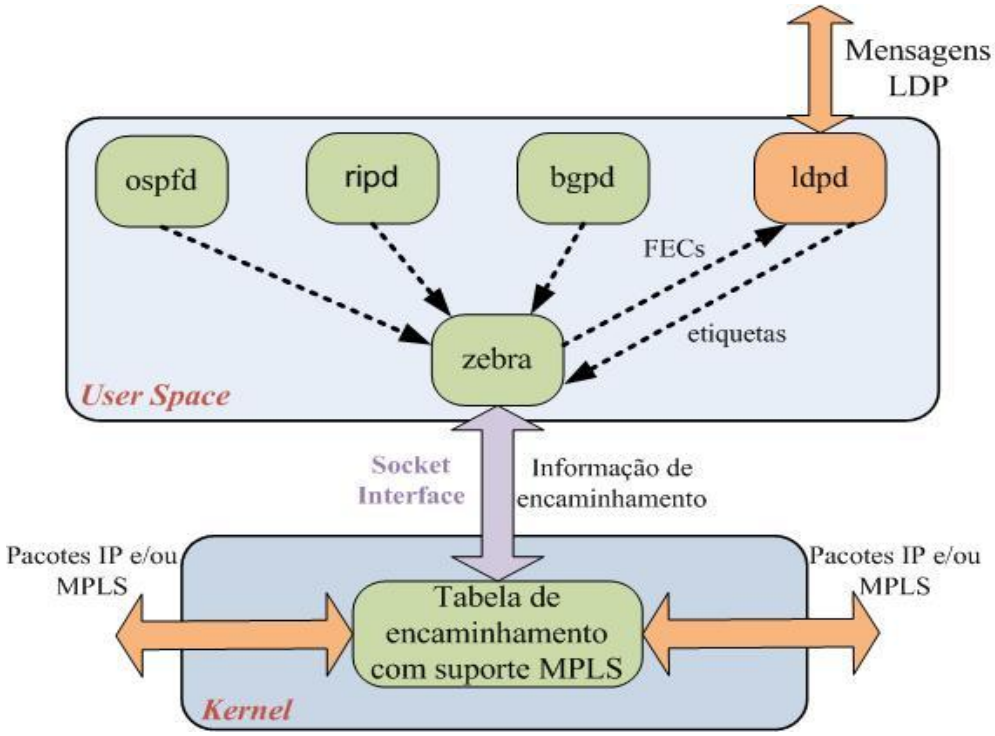


Figura 3.8 – Interação entre os dois módulos da solução "MPLS for Linux".

A plataforma *Quagga* foi desenvolvida para correr em sistemas Unix, no qual se insere o Linux. Uma máquina computacional que tenha o *Quagga* instalado pode comportar-se como um router dedicado, trocando informação de encaminhamento com outros routers que usem protocolos de encaminhamento. O *Quagga* usa esta informação para actualizar a tabela de encaminhamento do *kernel* [23]. Além dos protocolos de encaminhamento, o *Quagga* pode configurar interfaces, endereços e rotas estáticas. O RIP, OSPF e BGP são os protocolos de encaminhamento *unicast* suportados actualmente pelo *Quagga* [23].

O *Quagga* é constituído por um conjunto de processos (designados por *daemons*) que trabalham juntos para construir a tabela de encaminhamento. A Figura 3.8 mostra alguns dos *daemons* a correr em *user space*. Todos estes *daemons* comunicam com um *daemon* principal que tem a função de gerir a tabela de encaminhamento do *kernel*. Para isso, troca informação com o *kernel* através de *sockets* em software. Este *daemon* é designado por *zebra* [23].

Na figura, o *daemon ripd* implementa o protocolo RIP, por outro lado o *ospfd* é um *daemon* que suporta o protocolo OSPF na versão 2. O *bgpd* suporta o protocolo BGP-v4. Todos estes *daemons* trocam informação com os routers vizinhos. As rotas descobertas por eles são depois entregues ao

daemon zebra que realiza a gestão das mesmas. Com esta arquitectura de software, apenas é preciso correr o *daemon* relativo ao protocolo que se deseja [23].

O projecto "MPLS for Linux", aproveitando a arquitectura de software do *Quagga*, implementou o *daemon ldpd* que suporta o protocolo LDP. Este, por sua vez, comunica com o *zebra* para obter as rotas descobertas pelos outros *daemons* e atribui a cada uma delas uma etiqueta, trocando depois esta informação com os LSRs vizinhos. As etiquetas trocadas são depois entregues ao *zebra* que configura a tabela de encaminhamento do *kernel* [23].

Cada *daemon* tem o seu próprio ficheiro de configuração e um terminal de configuração. Por exemplo, uma rota estática deverá ser configurada no *zebra*, enquanto que a configuração de uma rede BGP deverá ser feita no bgpd. Para resolver isto, a plataforma *Quagga* fornece uma interface de configuração integrada designada por *vtysh*. A interface *vtysh* liga-se a cada *daemon* por meio de *sockets*, trabalhando com um *proxy* para quem está a configurar [23].

A Figura 3.8 ilustra também a função do *kernel* neste software. O *kernel* contém uma tabela de encaminhamento modificada pelo projecto "MPLS for Linux" para incorporar as etiquetas. Quando chega um pacote, esta tabela é consultada, e é dado o encaminhamento correcto ao pacote.

O Anexo II descreve todo o procedimento de instalação do "MPLS for Linux". Foram instalados o *kernel* MPLS e o módulo LDP. A versão do *kernel* utilizada foi a 2.6.27 e a do *Quagga* foi a 0.99.6.

3.7. Preparação dos testes

Os dois capítulos seguintes descrevem os testes efectuados que permitiram validar a solução IP/MPLS para os equipamentos da PT Inovação. Os testes foram realizados nos laboratórios desta empresa utilizando para o efeito uma plataforma de teste (*Agilent N2X*) e equipamentos de outros fabricantes (Cisco) com o objectivo de garantir, não só a funcionalidade e conformidade da solução, mas também a interoperabilidade da mesma com outras implementações. Após estas validações foram realizados igualmente testes de desempenho e de escalabilidade, que caracterizam a robustez e a capacidade de funcionamento da solução.

A solução escolhida "MPLS for Linux" correu sobre uma plataforma computacional genérica. Esta plataforma é caracterizada por um CPU da Intel, modelo Celeron de 3.6GHz, com 256KB de Cache e 1GB de RAM. Tem também disponíveis duas interfaces *Fast Ethernet* (RJ45) - uma embutida na *motherboard* e outra numa placa de rede - para comunicação exterior (Figura 3.9). O equipamento

Cisco disponível é um router da série 2800, com versão de software 12.4(4)T e duas interfaces Gigabit Ethernet do tipo RJ45.

Figura 3.9 - Interfaces da plataforma onde correrá a solução.

O equipamento de testes Agilent N2X é constituído por uma plataforma de processamento central e por um ou vários *chassis* que podem suportar vários tipos de cartas. A plataforma de processamento permite um controlo centralizado, fornecendo uma interface gráfica que controla as aplicações que correm nas cartas de teste. Por sua vez, as cartas de teste realizam o plano de dados implementando os diferentes protocolos. A carta usada no âmbito dos testes foi uma carta com quatro interfaces Ethernet do tipo 10/100/1000 Base-T. A Figura 3.10 mostra a carta do N2X com as suas interfaces assinaladas. Todos os equipamentos usados suportam emulação MPLS e o protocolo LDP.

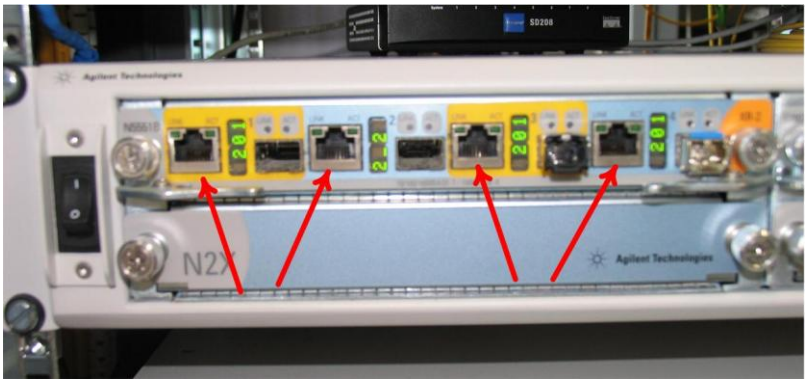


Figura 3.10 - Carta do N2X usada para realização dos testes.

Como referido anteriormente, o projecto "MPLS for Linux" suporta tanto o plano de dados como o plano de controlo. No entanto, os testes incidiram apenas sobre o plano de controlo, dado que os equipamentos da PT Inovação implementam o plano de dados recorrendo a um processador de rede dedicado. Assim, nesta dissertação foi usado o plano de dados (embutido no *kernel* do Linux)

apenas para emulação e encaminhamento dos pacotes MPLS. Nos testes teve-se também em conta as limitações que a plataforma computacional impõe, nomeadamente ao nível da capacidade de processamento e memória disponível, capacidades estas, que são diferentes das dos equipamentos da PT Inovação devido à plataforma de processamento dedicada.

Foi dada especial ênfase à funcionalidade LDP, dado ser o único protocolo de sinalização MPLS suportado pela solução testada.

Capítulo 4: Testes funcionais

4.1. Procedimentos

Numa primeira fase, os testes incidiram principalmente nos aspectos funcionais do protocolo LDP. Nesta fase é necessário dispensar especial atenção às especificações, de forma a garantir que o sistema testado consegue lidar com as situações positivas (cumprindo as especificações) e também com as negativas (pacotes mal formados, estados incorrectos, sequência incorrecta de mensagens) [24]. É importante ter testes positivos para testar o quanto o sistema funciona em condições normais, mas é extremamente importante ter também testes negativos, para assegurar uma operação robusta da solução quando ligada em rede com outros equipamentos.

Numa segunda fase foram realizados testes de desempenho e escalabilidade. Estes testes são importantes na medida em que determinam a capacidade e o desempenho da solução quando inserida numa rede de telecomunicações real. Os testes relativos à primeira fase são apresentados e discutidos neste capítulo. O capítulo seguinte apresentará os testes relativos à segunda fase.

4.2. Modos de funcionamento de um LSR

A arquitectura do MPLS [2] define vários modos de funcionamento para um LSR. Estes modos definem as acções efectuados pelos LSRs quando distribuem associações FEC-Etiqueta (doravante designadas por mapeamentos de etiquetas). Algumas destas acções são efectuadas pelos LSRs de *downstream*, como por exemplo distribuição e remoção de etiquetas. Outras acções são efectuadas pelos LSRs de *upstream*, como por exemplo, o pedido, libertação, disponibilidade e utilização de etiquetas. Conforme definido em [2], existem diversas combinações destas acções que são suportadas. Não é objectivo deste trabalho detalhar estas acções, mas é extremamente importante conhecer todos os modos de funcionamento possíveis de um LSR, para assim poder realizar testes mais completos. Através destes modos, consegue-se prever as acções tomadas pelos LSRs na distribuição de etiquetas.

As duas tabelas seguintes contêm as várias combinações possíveis para os diferentes modos de funcionamento de um LSR. Este modos (apresentados num dos capítulos anteriores) dividem-se em:

- Modo de distribuição: *Downstream on Demand* ou *Downstream Unsolicited*.
- Modo de retenção: Conservativo ou Liberal.
- Controlo de distribuição: Independente ou Ordenado.

A juntar a estes modos, está a detecção de *loops*. Esta funcionalidade, estando activa, em algumas situações pode obrigar o LSR a transmitir mensagens de mapeamentos e pedidos de etiqueta quando não precisaria de o fazer se estivesse desactiva. Por esta razão a detecção de *loops* também é incluída nas tabelas seguintes.

Dependendo se um LSR suporte ou não *merge* de etiquetas, existem alguns modos que não são suportados. A Tabela 4.1 contém todas as combinações para os modos de funcionamento de um LSR quando este suporta *merge* de etiquetas, definidos em [2].

Configuração	Modo De Distribuição		Modo de Retenção		Controlo de Distribuição		Detecção de *Loops*
	Downstream-on-Demand	Downstream Unsolicited	Conservativo	Liberal	Independente	Ordenado	
1	■		■			■	Opcional
2		■	■			■	Opcional
3		■		■	■		Sim
4		■		■	■		Não
5		■		■		■	Opcional
6	■		■		■		Sim
7	■		■		■		Não

Tabela 4.1 – Modos de funcionamento com suporte de *merge* de etiquetas, baseado em [2].

A Tabela 4.2 contém os modos de funcionamento possíveis para um LSR que não suporta *merge* de etiquetas. Aqui apenas são possíveis o modo de distribuição *On Demand* e retenção Conservativa, como referido em [2]. Os LSRs que não suportam *merge* de etiquetas usam múltiplas etiquetas para um mesmo FEC pelo que têm de fazer múltiplos pedidos de etiquetas para esse FEC, possível

apenas no modo *On Demand*. Como estes LSRs podem acomodar muitas etiquetas é aconselhável usar o modo de retenção Conservativo, reduzindo assim as etiquetas alocadas por LSR.

Configuração	Modo De Distribuição		Modo de Retenção		Controlo de Distribuição		Detecção de *Loops*
	Downstream-on-Demand	Downstream Unsolicited	Conservativo	Liberal	Independente	Ordenado	
8							Opcional
9							Não
10							Sim

Tabela 4.2 - Modos de funcionamento sem suporte de *merge* de etiquetas, baseado em [2].

De referir que o modo de distribuição *On Demand* e modo de retenção Conservativo só fazem sentido se forem configurados simultaneamente. No modo *On Demand*, cada LSR faz pedidos de etiquetas apenas ao LSR do próximo salto, de acordo com a tabela de encaminhamento. Como o modo Conservativo só retém etiquetas que sejam necessárias para realizar o encaminhamento dos pacotes (e aqui só se incluem as etiquetas do próximo salto), não fará sentido utilizar o modo *On Demand* em conjunto com o modo Liberal, pois os mapeamentos de etiquetas que existem em cada LSR são apenas os mapeamentos do LSR do próximo salto [5].

Os *loops* podem ser detectados com auxílio às mensagens LDP de pedido e mapeamento de Etiqueta, como referido anteriormente. Se os LSR estiverem configurados com controlo Independente, estas mensagens poderão ser enviadas antes de chegarem as mensagens dos seus LSRs vizinhos, multiplicando assim o envio destas mensagens se a detecção de *loops* estiver activada. Dado este facto, [5] recomenda que a detecção de *loops* seja usada em conjunto com controlo Ordenado, minimizando assim a quantidade de mensagens de mapeamento e pedido enviadas. No entanto, foram considerados os modos de configuração com controlo Independente em simultâneo com a detecção de *loops*. Com controlo Ordenado a detecção de *loops* é opcional, pois o modo de funcionamento do LSR não é alterado.

4.3. Descrição dos testes

Estes testes permitem aferir a conformidade da solução com as normas respectivas. Foram realizados diversos testes tendo em vista a validação do protocolo LDP. Assim, começou-se por validar a conformidade das mensagens LDP enviadas pela solução testada, bem como a recepção das mesmas. De seguida, testou-se o processo de descoberta de pares LDP e o estabelecimento de sessões por parte da solução. Foi também testada a detecção de *loops* e avaliado o comportamento da solução quando esta realiza funções de LER ou LSR, nos diferentes modos de funcionamento. Para estes testes foram usados todos os protocolos de encaminhamento IP disponíveis, verificando se o comportamento do LDP é igual para qualquer um deles. Por fim, foram testados os dois modos de *Label Space*, por plataforma e por interface.

A plataforma, que correu a solução testada, passará a designar-se por *Device Under Test* (DUT). O protocolo de encaminhamento usado em todos os cenários é o OSPF (excepto na detecção de *loops*) e a gama de etiquetas de 16 a 1048575 (a gama de 0 a 15 encontra-se reservada pela norma [5]). Todos os cenários de teste descritos neste capítulo, à excepção do último teste onde é testado também o *Label Space* por interface, usam *Label Space* por plataforma por ser o modo mais simples de funcionamento, onde uma determinada etiqueta só pode estar a atribuída a um FEC numa interface, e também por ser a mais utilizada pelos operadores. No final desta secção foram realizados testes simples aos dois tipos de *Label Space*.

Os testes funcionais foram realizados com base em [24] e na especificação [5]. O documento [24] descreve, em formato de tabela, vários testes funcionais ao protocolo LDP que os fornecedores de serviços poderão utilizar quando implementam os seus equipamentos MPLS. No entanto, nesta dissertação foram também incluídos alguns testes não mencionados em [24], tais como, aqueles que permitem diferenciar o funcionamento de um LSR de um LER, os testes ao *Label Space* e aos modos de funcionamento. Desta forma, conseguimos ter testes mais abrangentes, permitindo validar todos os detalhes do protocolo LDP. Existem ainda alguns testes especificados em [24] que não foram aqui considerados, nomeadamente os testes relacionados com a descoberta extensiva, uma vez que este tipo de descoberta permite estabelecer LSPs entre dois pontos específicos, utilizado na construção de *pseudowires* [26]. Como o DUT não suporta estabelecimento de *pseudowires*, a descoberta extensiva não foi incluída nos testes.

4.3.1. Mensagens LDP

As mensagens LDP são usadas pelos LSRs para descobrir os seus vizinhos na rede e para trocar informações sobre etiquetas. Nesta secção pretende-se testar o envio e a recepção destas mensagens por parte do DUT. Nestes testes é usado como referência o cenário da Figura 4.1. Neste cenário, constituído pelo DUT, por um router Cisco e pelo N2X, o LDP está activo nas interfaces que interligam o Cisco e o N2X ao DUT, existindo apenas troca de mensagens LDP nestas duas ligações. O N2X pode funcionar como *sniffer* de pacotes, permitindo capturar as mensagens LDP enviadas pelo DUT para posterior análise. No N2X há também a possibilidade de formatar as mensagens LDP que se querem enviar, aumentando assim o leque de testes negativos (mensagens mal formadas, sequências incorrectas) realizados.

Em cada equipamento MPLS é configurada uma interface *Dummy* com um IP de *Host*. Como cada router pode ter múltiplas interfaces configuradas, estes IPs de *Host* são usados para identificar cada router, tanto no processo OSPF como no LDP.

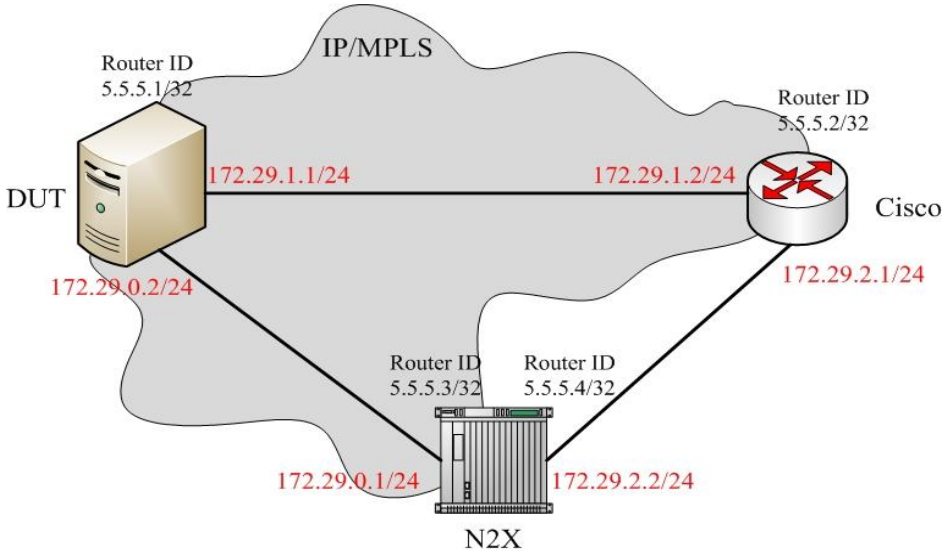


Figura 4.1 - Cenário de referência para o teste às mensagens LDP.

Na figura anterior a interface ETH0 do DUT está ligada directamente ao N2X. A ligação ao Cisco é realizada pelo DUT através da sua interface ETH1. Os *daemons* lançados foram o *zebra*, *ospfd* e *ldpd*. Sabendo isto, as configurações realizadas no DUT para comunicação com o N2X e Cisco foram as seguintes:

```
!Endereço IP na interface ETH0 e ETH1
int eth0
ip address 172.29.0.2/24
int eth1
ip address 172.29.1.1/24
!Endereço IP de Host para o router ID
int dummy0
ip address 5.5.5.1/32
!Activar OSPF e encaminhamento na rede IP
router ospf
network 172.29.0.0/24 area 0
network 172.29.1.0/24 area 0
!Definir label space
int eth0
mpls labelspace 0
int eth0
mpls labelspace 0
!Activar a funcionalidade LDP
mpls ldp
!Activar LDP na interface ETH0 e ETH1
int eth0
mpls ip
int eth1
mpls ip
```

As configurações do DUT são simples e dividem-se em três fases: configurar endereços das interfaces, configurar rede OSPF e configurar LDP.

A Figura 4.2 exemplifica uma troca de mensagens que pode ocorrer entre o N2X e o DUT. A situação 1 corresponde ao mecanismo de descoberta e estabelecimento da sessão LDP. Os LSRs para descobrir potenciais pares LDP enviam mensagens de *hello*. De seguida, é estabelecida uma ligação de transporte (TCP); neste caso não são enviadas mensagens LDP. Após estabelecida esta ligação são trocadas entre os dois LSRs mensagens de inicialização e de endereço. As primeiras servem para negociar parâmetros da sessão LDP, enquanto as mensagens de endereço anunciam os endereços configurados nas interfaces de cada LSR. As mensagens de *keepalive* são usadas para manter a sessão LDP. Na situação 2 são distribuídos mapeamentos de etiqueta através das mensagens de pedido e mapeamento. Nesta situação os LSRs encontram-se configurados no modo *Downstream on Demand*, dado que é enviada a mensagem de pedido de etiqueta. Se o LSR, que enviou o pedido, quiser abortar este mesmo pedido, irá enviar a mensagem abortar pedido de etiqueta, ilustrada pela situação 3. Na situação 4, a mensagem de remoção de endereço é enviada quando é desactivada uma interface, e a situação 5 ocorre quando um dos LSRs já não necessita do mapeamento FEC-Etiqueta. Nesta última situação são enviadas as mensagens de remover etiqueta e libertar etiqueta.

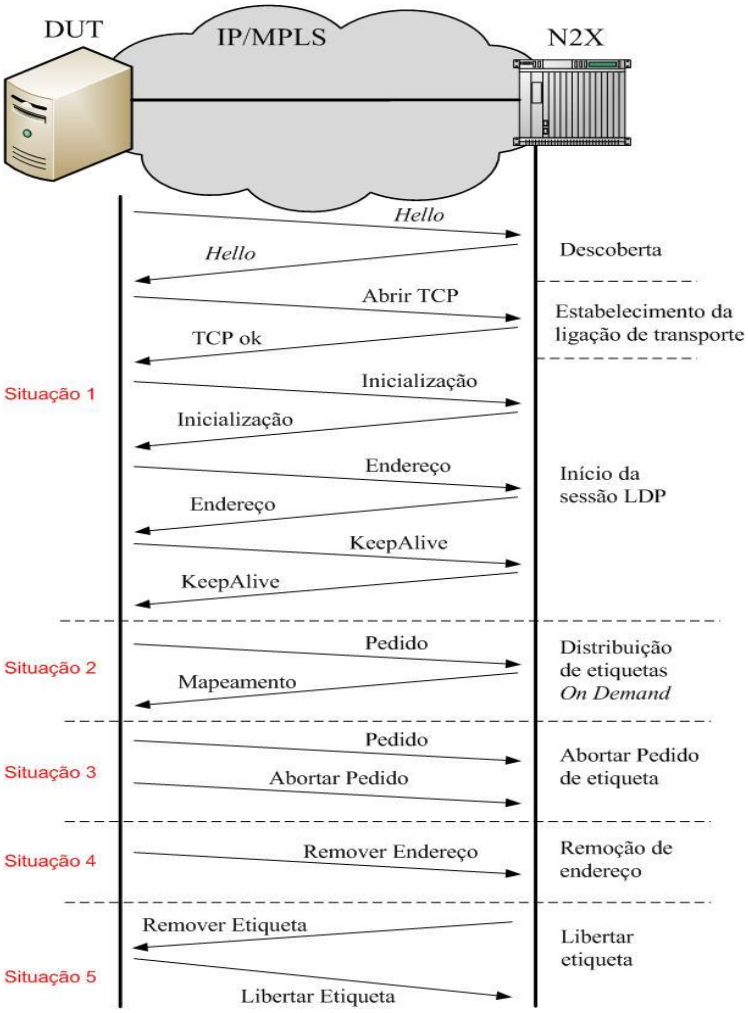


Figura 4.2 - Exemplo de troca de mensagens LDP entre dois LSRs.

O formato geral das mensagens LDP encontra-se descrito no Anexo I. De seguida serão detalhados os testes efectuados para cada uma das mensagens LDP.

4.3.1.1. Hello

As mensagens de *hello* (MSG ID 0x0100) fazem parte do mecanismo de descoberta do LDP. Quando um LSR recebe *hellos* de outro LSR guarda uma referência para cada um desses hellos. Para cada uma dessas referências é associado um temporizador, que reinicia sempre que é recebida uma mensagem de *hello* que corresponda a essa referência. Se o valor deste temporizador se anular, a referência associada será descartada [5]. A [5] recomenda que o intervalo de tempo entre retransmissões de *hellos* seja pelo menos um terço do valor do temporizador.

Inicialmente, neste teste foram enviadas várias mensagens de *hello* para o DUT com diferentes valores de temporizadores. A aceitação destas mensagens e dos seus respectivos temporizadores

podem ser vistos no DUT através do comando "**show ldp session**". Se o valor do temporizador se esgotar, este deve descartar a referência à mensagem de *hello*, desaparecendo assim da tabela de descobertas.

Sempre que um LSR recebe uma mensagem de *hello* verifica o valor do seu temporizador, compara esse valor com o seu (definido pelo utilizador) e utiliza o menor dos dois [5]. O DUT permite modificar o valor do seu temporizador, alterando a periodicidade do envio de *hellos*. O valor do temporizador será sempre três vezes maior que esta periodicidade. Sendo assim, isto foi testado enviando mensagens de *hello* do N2X com diferentes valores de temporizadores. O DUT terá de ajustar o temporizador da referência *hello* se o temporizador recebido tiver menor valor. Caso contrário, o valor usado será o definido pelo utilizador.

Quando um LSR recebe *hellos* com o valor de temporizador igual a zero, este deve usar o valor por omissão de 15 segundos [5]. Assim, enviando *hellos* do N2X com um temporizador nulo, o DUT deverá usar o valor de 15 segundos.

As mensagens de *hello* são compostas por TLV obrigatórios e opcionais. Dentro dos TLV obrigatórios existe um designado por *Common Hello Parameters* TLV. Este TLV é constituído por vários campos entre os quais o campo *Reserved*, que deve ser enviado preenchido com zeros e ignorado na recepção [5]. Neste teste foram enviadas mensagens *hello* com o campo *Reserved* preenchido com valor diferente de zero, as quais o DUT deverá aceitar, ignorando o campo *Reserved*. Também foi verificado o envio de *hellos* por parte do DUT, em que o campo *Reserved* deverá ir a zero.

4.3.1.2. Inicialização

As mensagens LDP de inicialização (MSG ID 0x0200) são trocadas entre dois LSRs que estabelecem uma ligação de transporte. São usadas para negociar vários parâmetros da sessão LDP [5]. Um destes parâmetros é o tipo de distribuição de etiquetas, que pode ser *Downstream Unsolicited* ou *Downstream On Demand*. Se um LSR propuser *Downstream Unsolicited* e o outro propuser *Downstream on Demand*, o modo *Downstream Unsolicited* deverá ser utilizado [5]. Este ponto foi testado através do envio de mensagens de inicialização com um modo diferente do que está configurado no DUT. Em qualquer um dos casos o DUT deverá sempre adoptar o modo *Downstream Unsolicited*. O DUT apenas deverá adoptar o modo *Downstream on Demand* se estiver configurado para tal e se receber uma mensagem de inicialização com o mesmo modo. O comando para alterar o modo de distribuição no DUT é o "**distribution-mode [**dod | du**]**" dentro do menu "**mpls ip**".

O campo *Loop Detection* existente nestas mensagens, que indica se a detecção de *loops* está activa, foi testada capturando as mensagens enviadas pelo DUT e comparando este campo ao que está configurado. Outro campo existente nestas mensagens é o *Path Vector Limit* que deverá ser preenchido e enviado com zeros se a detecção de *loops* não estiver activa no DUT. Se esta funcionalidade estiver activa este campo deverá ir com o valor configurado no equipamento, configurável no DUT através do comando "**max-path-vector** <1-255>" no menu "**mpls ip**".

Tal como as mensagens de *hello*, as mensagens de inicialização contêm um campo reservado que deverá ser preenchido com zeros na transmissão e ignorado na recepção. Outro campo existente é o temporizador *keepalive*, referido na secção seguinte.

4.3.1.3. Keepalive

O LDP utiliza a recepção regular de mensagens como forma de monitorizar a integridade de uma sessão LDP. Um LSR mantém um temporizador *keepalive* por cada sessão que reinicia sempre que recebe uma mensagem LDP. Se este temporizador expirar porque não recebeu nenhuma mensagem LDP, o LSR conclui que a ligação de transporte está deteriorada ou que o seu par falhou, terminando assim a sessão LDP fechando a ligação de transporte TCP. Qualquer mensagem LDP recebida numa sessão reinicia o seu temporizador. No entanto, em circunstâncias onde nenhuma mensagem seja recebida, num determinado período de tempo, deverão ser usadas as mensagens de *keepalive* (MSG ID 0x0201) [5].

No DUT foi testado o envio e a recepção destas mensagens. O valor do temporizador *keepalive* é definido na recepção das mensagens de inicialização, referidas anteriormente. Aqui também se aplica o princípio das mensagens de *hello*, ou seja, o valor de temporizador menor (recebido ou configurado localmente) será o valor usado. No DUT não se consegue definir o valor local deste temporizador, mas consegue-se definir a periodicidade de envio destas mensagens através do comando "**keepalive-interval** <1-60>" no menu "**mpls ip**". Este valor nunca deverá exceder o valor do temporizador.

4.3.1.4. Endereço

Quando uma sessão LDP é inicializada, e antes de serem enviadas mensagens de pedido e mapeamento de etiqueta, um LSR deve anunciar todos os seus endereços IP configurados através das mensagens LDP de endereço (MSG ID 0x0300). Estes endereços são usados pelos LSRs para manter uma base de dados que contém associações entre LSR IDs e endereços do próximo salto

[5]. Estas relações ajudam o LSR a determinar se algum dos endereços anunciados é o próximo salto para determinado destino, para assim poder realizar o encaminhamento.

Neste teste foi verificado se o DUT transmite mensagens de endereço antes de enviar qualquer mensagem de pedido ou mapeamento de etiqueta. Foi verificado também se o DUT anuncia nestas mensagens todos os endereços configurados nele próprio. Todos os endereços recebidos pelo DUT podem ser visualizados através do comando "**show ldp neighbor**", debaixo do ponto "**Addresses bound to peer:**"

Dentro do TLV da mensagem de endereço deve ser especificada a família de endereços, de acordo com [27]. Neste caso, o DUT deverá enviar este campo preenchido com o valor 1, que significa família de endereços IPv4. Se o DUT receber alguma mensagem com família de endereços não suportada (diferente de 1), deverá enviar uma notificação (*Unsupported Address Family*) de volta para o equipamento que originou a mensagem de endereço.

4.3.1.5. Remoção de endereço

Um LSR transmite uma mensagem de remoção de endereço (MSG ID 0x0301) para um par LDP com o objectivo de remover endereços anunciados previamente [5]. Sendo assim, ao desactivar qualquer interface no DUT, este deverá enviar uma mensagem de remoção de endereço para avisar o seu par LDP.

4.3.1.6. Mapeamento de etiquetas

As mensagens LDP de mapeamento de etiquetas (MSG ID 0x0400) são usadas por um determinado LSR quando quer anunciar os seus mapeamentos Etiqueta-FEC ao seu par LDP [5]. As condições em que são enviadas estas mensagens dependem muito do modo de distribuição e controlo de etiquetas configurado nos LSRs. Desta forma é fácil concluir que a quantidade de combinações de teste é muito elevada.

Um LSR, que opere no modo de controlo Independente, envia uma mensagem de mapeamento de etiqueta sempre que as seguintes condições se verifiquem [5]:

- Um novo FEC é reconhecido via tabela de encaminhamento e o modo de distribuição é *Unsolicited*.
- Recebe um pedido de etiqueta do LSR de *upstream* para um determinado FEC presente na tabela de encaminhamento.

- O próximo salto é alterado para um determinado FEC e a detecção de *loops* está activada.
- Os atributos do mapeamento são alterados.
- Recebe um mapeamento de etiqueta do LSR de *downstream* e (i) ainda não há mapeamento local ou (ii) a detecção de *loops* está activada ou (iii) os atributos do mapeamento foram alterados.

Quando um LSR se encontra a funcionar no modo de controlo Ordenado, as condições necessárias para enviar uma mensagem de mapeamento de etiqueta são as seguintes [5]:

- Um novo FEC é reconhecido via tabela de encaminhamento e o LSR é o LER de saída para esse FEC.
- Recebe um pedido de etiqueta do LSR de *upstream* para um determinado FEC presente na tabela de encaminhamento e o LSR é o LER de saída para esse FEC ou já possui o mapeamento do LSR de *downstream*.
- O próximo salto é alterado para um determinado FEC e a detecção de *loops* está activada.
- Os atributos do mapeamento são alterados.
- Recebe um mapeamento de Etiqueta do LSR de *downstream* e (i) ainda não há mapeamento local ou (ii) a detecção de *loops* está activada ou (iii) os atributos do mapeamento foram alterados.

De referir que a detecção de *loops* é uma funcionalidade que pode levar ao envio de mapeamentos, que em situações onde não estivesse activada não seriam enviados. Com base nos pontos anteriores foi construída a Tabela 4.3 que resume os testes a efectuar. Nesta tabela é associado o teste (Causa) com a resposta a ser dada pelo DUT (Consequência) e os modos configurados. Na tabela os termos mapeamento e pedido referem-se às mensagens LDP mapeamento e pedido de etiqueta, respectivamente.

Teste	Causa	Consequência	Modos de funcionamento
1	Recebeu um pedido	Inclui o TLV "*Request Message ID*" na mensagem de mapeamento	1, 2, 5, 6, 7, 8, 9, 10
2	Recebeu um pedido para um FEC para o qual já tinha sido fornecido mapeamento e esse pedido não é duplicado	Envia mensagem de mapeamento	1, 6, 7, 8, 9, 10
3	Recebeu um pedido para um FEC para o qual é um LER de saída	Envia mensagem de mapeamento	1, 2, 5, 6, 7, 8, 9, 10

4	Recebeu um pedido para um FEC para o qual ainda não tem mapeamento do LSR de *downstream*	Envia mensagem de mapeamento	6, 7, 9, 10
5	Recebeu um mapeamento de *downstream*, o qual já tinha sido fornecido, mas este tem atributos – *Hop Count* e *Path Vector* – diferentes	Envia mensagem de mapeamento	1, 2, 3, 5, 6, 8, 10
6	Reconheceu um novo FEC e não é o LER de saída para esse FEC	Envia mensagem de mapeamento	3, 4
7	Reconheceu um novo FEC e é o LER de saída para esse FEC	Envia mensagem de mapeamento	2, 3, 4, 5
8	Recebeu um mapeamento do LSR de *downstream* e tem um pedido pendente para esse FEC	Envia mensagem de mapeamento	1, 2, 5, 8
9	Reconheceu um novo FEC e já tem o mapeamento do LSR de *downstream*	Envia mensagem de mapeamento	2, 5
10	Recebeu um mapeamento de *downstream*, o qual já tinha sido fornecido com os mesmos atributos – *Hop Count* e *Path Vector*	Não envia mensagem de mapeamento	1, 2, 3, 5, 6, 8, 10
11	Não tem o mapeamento para um determinado FEC do seu LSR de *downstream*	Não envia mensagem de mapeamento	2, 5
12	Recebeu um mapeamento com um elemento do TLV FEC do tipo "*Wildcard*"	Envia notificação ou mensagem Libertar Etiqueta	Todos

Tabela 4.3 - Testes para a mensagem Mapeamento de Etiqueta.

O teste 1 só é possível quando os LSR estão configurados em todos os modos excepto o *Downstream Unsolicited* Independente. Neste modo, os LSR poderão enviar mensagens de mapeamento sem que estas sejam respostas a pedidos. Desta forma, sendo o TLV "*Request Message ID*" opcional e igual ao TLV que chega na mensagem de pedido, este não será incluído quando os LSRs operam no modo *Downstream Unsolicited* Independente. Este princípio também se aplica ao teste 3, onde a obrigatoriedade de transmitir mapeamentos como forma de responder a pedidos não se aplica a este modo.

4.3.1.7. Pedido etiqueta

A mensagem LDP de pedido de etiqueta (MSG ID 0x0401) é usada pelos LSR de *upstream* para solicitar explicitamente ao LSR de *downstream* que atribua e anuncie uma etiqueta para um

determinado FEC [5]. Um LSR poderá transmitir um pedido de etiqueta sob qualquer uma das seguintes condições [5]:

- Quando reconhece um novo FEC através da tabela de encaminhamento, o próximo salto é um par LSR, o qual ainda não forneceu a etiqueta para este FEC.
- O próximo salto para o FEC mudou e ainda não forneceu a etiqueta para este FEC.
- Recebe um Pedido de Etiqueta do LSR de *upstream*, o próximo salto é um par LSR, o qual ainda não forneceu a etiqueta para este FEC.

De acordo com os pontos anteriores, foi construída a Tabela 4.4 que resume os testes a efectuar para esta mensagem.

Teste	Causa	Consequência	Modos de funcionamento
1	Detectou uma alteração no próximo salto para um FEC e ainda não foi fornecido mapeamento de etiqueta desse LSR	Envia mensagem de pedido para o próximo salto	1, 2, 6, 7, 8, 9, 10
2	Surgiu um novo FEC para o qual ainda não tem mapeamento de etiqueta do LSR de *downstream*	Envia mensagem de pedido	1, 2, 6, 7, 8, 9, 10
3	Recebeu um pedido para um FEC para o qual não tem rota na tabela de encaminhamento	Envia mensagem notificação (*No Route*)	1, 2, 6, 7, 8, 9, 10
4	Surgiu um novo FEC para o qual ainda não tem mapeamento de etiqueta do próximo salto	Não envia mensagem de pedido	3, 4, 5
5	Recebeu uma notificação (*No Resources*) como resposta a um dos seus pedidos	Não envia mais qualquer mensagem de pedido para o par LSR	1, 2, 6, 7, 8, 9, 10
6	Recebeu pedidos que não são duplicados e não tem *merge* de etiquetas	Poderá enviar múltiplas mensagens de pedido para o mesmo FEC	8, 9, 10
7	Recebeu pedidos que não são duplicados e tem *merge* de etiquetas	Não envia múltiplas mensagens de pedido para o mesmo FEC	1, 6, 7
8	Surgiu um novo FEC para o qual é o LER de entrada	Envia mensagem de pedido	1, 6, 7, 8, 9, 10
9	Recebeu uma notificação (*Resources Available*) de um LSR que anteriormente tinha enviado outra notificação (*No Resources*)	Envia mensagem de pedido	1, 2, 6, 7, 8, 9, 10

10	Recebeu um pedido com um elemento do TLV FEC do tipo "*Wildcard*"	Envia notificação	1, 2, 5, 6, 7, 8, 9, 10

Tabela 4.4 - Testes para a mensagem Pedido de Etiqueta.

De referir que os modos de configuração dos testes 1, 2, 3, 4, 5 e 9 não incluem a retenção Liberal, pois neste modo em conjunto com a distribuição *Unsolicited* não são enviadas mensagens de pedido de Etiqueta. Este facto deve ser comprovado pelo teste 4.

Os testes 6 e 7 pretendem distinguir entre os LSRs que suportam *merge* de etiquetas, que só enviam uma etiqueta por FEC; e os LSRs que não suportam *merge* de etiquetas e que podem enviar múltiplas etiquetas num FEC.

O teste 8 só deverá acontecer para a distribuição *On Demand*. No modo *Unsolicited* não deverão ser enviados pedidos de etiquetas.

4.3.1.8. Abortar pedido de etiqueta

Um LSR poderá enviar uma mensagem LDP de abortar pedido de etiqueta (0x0404) para abortar um pedido de etiqueta que esteja pendente (ainda não recebeu o mapeamento correspondente) nas seguintes circunstâncias [5]:

- O próximo salto para o FEC em questão já não é o mesmo.
- Não suporta *merge* de etiquetas, não é o LER de entrada para o FEC e recebeu uma mensagem abortar pedido dum LSR de *upstream*.
- Suporta *merge* de etiquetas, não é o LER de entrada para o FEC e recebeu uma mensagem abortar pedido dum LSR de *upstream*, o qual é o último LSR de *upstream*.

Se um LSR receber uma mensagem de abortar pedido de etiqueta depois de já ter respondido ao pedido de etiqueta com um mapeamento ou notificação, esse LSR deverá ignorar essa mensagem. No cenário de testes foram enviadas estas mensagens para o DUT, depois de se ter estabelecido os LSPs. Neste caso, o DUT deverá descartá-las. Caso contrário, ou seja, se tiver algum pedido de etiqueta pendente, então deve abortá-lo e enviar uma notificação (*Label Request Aborted*). No caso, de receber uma mensagem em que o TLV "*Message ID*" não iguale o mesmo campo no pedido de etiqueta recebido, então o DUT não deverá responder com a notificação.

Quando um pedido de etiqueta é abortado com sucesso, deverá ser enviada uma notificação, como referido anteriormente. No entanto, esta notificação deverá ser enviada com o mesmo "*Message ID*" que a mensagem de abortar pedido de etiqueta recebida.

Se o DUT estiver configurado para não ter *merge* de etiquetas e modo de retenção Conservativo, então se tiver enviado múltiplos pedidos de etiqueta para um FEC e detectar uma alteração no seu próximo salto, deverá enviar múltiplas mensagens de abortar pedido de etiqueta.

O DUT não deverá enviar mensagens de abortar pedido se tiver *merge* de etiquetas, se tiver outros pedidos pendentes para o mesmo FEC (recebidos do LSR de *upstream*) e se já tiver enviado um pedido para o seu LSR de *downstream* [24].

4.3.1.9. Remover etiqueta

Um LSR envia uma mensagem LDP de remover etiqueta (MSG ID 0x0402) para sinalizar ao seu par LDP para não continuar a usar o mapeamento FEC-Etiqueta, anunciado previamente. Esta mensagem remove a ligação entre o FEC e a etiqueta. Estas mensagens são transmitidas quando um LSR já não reconhece um FEC para o qual anunciou previamente uma etiqueta ou porque o operador da rede decidiu eliminar esse mapeamento por configuração no LSR. Um LSR que recebe uma mensagem de remover etiqueta deverá responder com uma mensagem de libertar etiqueta [5].

Se o DUT estiver configurado no modo de controlo Ordenado, deverá enviar uma mensagem de remover etiqueta para o seu LSR de *upstream*, se receber esta mensagem do seu LSR de *downstream*. O DUT deverá também enviar esta mensagem se detectar alteração no próximo salto e já tiver distribuído etiquetas para os seus vizinhos.

4.3.1.10. Libertar etiqueta

Um LSR envia uma mensagem LDP de libertar etiqueta (MSG ID 0x0403) para sinalizar a um par LDP que já não precisa do mapeamento FEC-Etiqueta anunciado previamente por esse par. Estas mensagens deverão ser transmitidas sempre que [5]:

- O LSR que enviou o mapeamento de etiqueta já não é o próximo salto para o FEC em questão. O LSR está configurado no modo Conservativo.
- O LSR recebe um mapeamento de etiqueta dum LSR que não é o próximo salto para o FEC em questão. O LSR está configurado no modo Conservativo.
- O LSR recebeu uma mensagem de remover etiqueta.

Se o DUT estiver configurado no modo de controlo Liberal, obviamente não deverá transmitir mensagens de libertar etiqueta nos dois primeiros pontos anteriores. Neste caso, o DUT deverá manter a etiqueta em questão, para que possa ser usada imediatamente se o LSR de *downstream* se tornar o próximo salto.

O DUT deverá também enviar esta mensagem se detectar um *loop* através dos campos *Hop Count* e *Path Vector*, e se tiver recebido uma mensagem mapeamento para o qual já tinha etiqueta associada mas esta etiqueta não corresponde à já existente.

No caso de o DUT receber esta mensagem deverá remover a etiqueta associada das suas tabelas de encaminhamento.

4.3.1.11. Notificações

Se um LSR encontrar determinada condição que requer notificar o seu par LDP, com informação de erro ou apenas aviso, o LSR irá enviar para o ser par uma mensagem de notificação (MSG ID 0x0100) contendo um TLV que codifica a causa da notificação [5]. Foi testado o envio de mensagens de notificação por parte de DUT quando recebe mensagens LDP defeituosas, nomeadamente com TLVs desconhecidos ou defeituosos, bem como em situações mais genéricas.

A Tabela 4.5 contém todas as acções a ser tomadas pelo DUT, quando este recebe as seguintes mensagens LDP defeituosas: Inicialização, Pedido de Etiqueta, Mapeamento de Etiqueta, Remover Etiqueta, Libertar Etiqueta, Abortar Pedido de Etiqueta, Endereço e Remover Endereço. A tabela contém o tipo de defeito da mensagem enviada e a respectiva resposta em formato de mensagem de notificação.

Teste	Mensagem de Notificação enviada pelo DUT
Identificador LDP desconhecido	*Bad LDP Identifier*
Versão do protocolo LDP desconhecido	*Bad Protocol Version*
Tamanho de PDU superior (> 4096 bytes) ao máximo ou inferior ao mínimo (< 14 bytes)	*Bad PDU Length*
Tipo de mensagem desconhecido (<0x8000)	*Unknown Message Type*
Tamanho de mensagem inválido	*Bad Message Length*
Mensagem sem alguns parâmetros obrigatórios	*Missing Message Parameters*

Tabela 4.5 - Envio de notificações como resposta a mensagens LDP inválidas.

Em relação às mensagens com TLVs desconhecidos ou defeituosos, a Tabela 4.6 apresenta os testes efectuados. As mensagens LDP enviadas foram as mesmas que as anteriores. A única diferença é que agora estas mensagens foram enviadas com erros nos TLVs.

Teste	Mensagem de Notificação enviada pelo DUT
Tamanho do TLV demasiado grande	*Bad TLV Length*
Tipo de TLV desconhecido (<0x8000)	*Unknown TLV*
Valor do TLV errado	*Malformed TLV Value*
Tipo de FEC errado	*Unknown FEC*

Tabela 4.6 - Envio de notificações como resposta a TLVs desconhecidos ou defeituosos.

Também foi testado o envio de notificações por parte do DUT, devido a situações mais genéricas:

- Verificar que o DUT retransmite uma notificação para o LSR de *upstream*, quando recebe do LSR de *downstream* uma notificação com o bit 'F' a 1.
- Verificar que o DUT não responde quando recebe do LSR de *downstream* uma notificação com o bit 'F' a 0.
- Verificar que o DUT remove todas as etiquetas descobertas sobre uma sessão se a ligação TCP cair.

4.3.2. Descoberta e estabelecimento de sessões LDP

Estes testes permitem validar o procedimento de descoberta de potenciais vizinhos LDP e o estabelecimento de sessões LDP com os mesmos.

4.3.2.1. Descoberta LDP básica

O mecanismo de descoberta de vizinhos LDP é realizado com recurso a mensagens de *hello*, como referido anteriormente. O DUT apenas suporta o modo de descoberta básico. Este teste foi realizado com recurso ao cenário da Figura 4.3. e pretende averiguar a recepção e transmissão destas mensagens por parte do DUT. A recepção de mensagens foi analisada com recurso ao comando "**show ldp discovery**", que apresenta uma tabela com todas as descobertas realizadas. Nesta tabela poderão ser confirmados os vários campos enviados na mensagem de *hello* pelo N2X, tais como, tipo de descoberta (básica ou extensiva) e endereço de transporte IP. O tempo de espera

recebido na mensagem será o tempo máximo pelo qual o DUT irá descartar a descoberta se não receber entretanto mais nenhuma mensagem de *hello*.

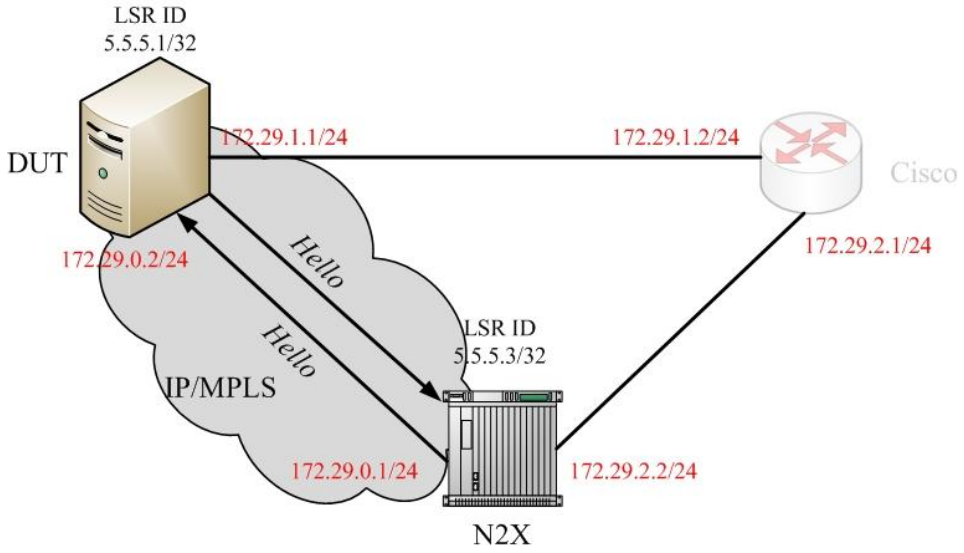


Figura 4.3 - Teste de descoberta de pares LDP.

Se o DUT receber mensagens de *hello* inválidas, terá de descartar essas mensagens e não assinalar a descoberta em "**show ldp discovery**". Foram analisadas as situações em que o DUT recebe mensagens com um tamanho TLV superior ao especificado no tamanho da mensagem e também mensagens com versão do protocolo LDP não suportada pelo DUT.

A transmissão das mensagens *hello* por parte do DUT foi analisada no N2X e foi tido em conta o conteúdo das mesmas e também o tempo de retransmissões destas mensagens.

4.3.2.2. Estabelecimento da ligação de transporte

Concluída a descoberta de vizinhos LDP através da troca de *hellos*, são estabelecidas ligações TCP para o suporte das respectivas sessões LDP. Nestas ligações é atribuído o papel activo ao LSR que possuir maior endereço de transporte. Este endereço é incorporado no TLV de Transporte de Endereço das mensagens de *hello*. Sendo este parâmetro opcional e no caso de não existir, será usado o endereço IP de origem da mensagem de *hello* [5]. Usando o cenário da Figura 4.3, foi verificado o estabelecimento de ligações TCP por parte do DUT através do *trace* de mensagens TCP. Foram transmitidas para o DUT mensagens com e sem TLV de Endereço de Transporte, com o fim de validar o procedimento de escolha de papel activo ou passivo por parte do DUT.

4.3.2.3. Inicialização da sessão LDP

A inicialização da sessão LDP é realizada com a troca de mensagens de inicialização LDP. Esta mensagem é enviada pelo LSR que desempenha o papel activo. O LSR que desempenha o papel passivo espera por esta mensagem para depois serem negociados os parâmetros da sessão (versão do protocolo LDP, método de distribuição de etiqueta, valores de temporizadores) [5]. Neste teste será validada a criação de sessões LDP pelo DUT no modo passivo e activo. As sessões LDP estabelecidas pelo DUT podem ser analisadas em "**show ldp session**".

Foi analisada igualmente a resposta do DUT quando este recebe mensagens de inicialização inválidas. O DUT terá de transmitir uma notificação de "*Session Rejected*" quando está no papel passivo e recebe:

- Uma mensagem de inicialização com uma versão de protocolo LDP incompatível.
- Uma mensagem de inicialização com um LSR ID que não corresponde a nenhum *hello* recebido anteriormente.
- Uma mensagem de inicialização com um tamanho de PDU inválido (maior que o definido na mensagem LDP).

Se o DUT desempenhar o papel activo e as suas mensagens de inicialização estiverem continuamente a serem rejeitadas (por exemplo devido a uma não aceitação dos parâmetros LDP por parte do par LDP), este deve encerrar a sessão LDP. As tentativas seguintes de restabelecimento da sessão devem ser realizadas após 15 segundos pelo menos, e as seguintes tentativas em nunca menos de 120 segundos [5].

4.3.2.4. Máquina de estados da inicialização

O processo de negociação de uma sessão LDP pode ser representado por uma máquina de estados, descrita em [5]. Esta máquina de estados, representada na Figura 4.4, contém todas as acções realizadas pelo LSR ao inicializar uma sessão LDP.

A máquina de estados é composta por 5 estados possíveis. O primeiro desses estados é o estado "Não Existente" em que o LSR pode ou não ter recebido *hellos*, mas ainda não estabeleceu a ligação de transporte. Assim que esta ligação é estabelecida, o LSR transita para o estado de "Inicialização". Neste estado, dependendo se tem o papel activo ou passivo, o LSR terá de enviar uma mensagem de inicialização ou esperar pela recepção desta mensagem, respectivamente. Se receber qualquer outra mensagem, o LSR deve transitar novamente para o estado "Não Existente". No papel activo, depois de enviar a mensagem de inicialização, o LSR transita para o estado

"Aberto a transmitir", no qual espera pela recepção da mensagem de inicialização do seu par LDP. Assim que receber esta mensagem, transita para o estado "Aberto a receber", transmite a mensagem de *keepalive*, e espera pela recepção da mensagem de *keepalive* do seu par LDP para depois transitar para o estado "Operacional".

No papel passivo, depois de receber a mensagem de inicialização do seu par LDP, o LSR transita para o estado "Aberto a receber", transmitindo a sua mensagem inicialização e de *keepalive*. Neste estado, depois de receber a mensagem de *keepalive* do seu par LDP, transita para o estado "Operacional". Este estado é mantido com a recepção regular de mensagens LDP.

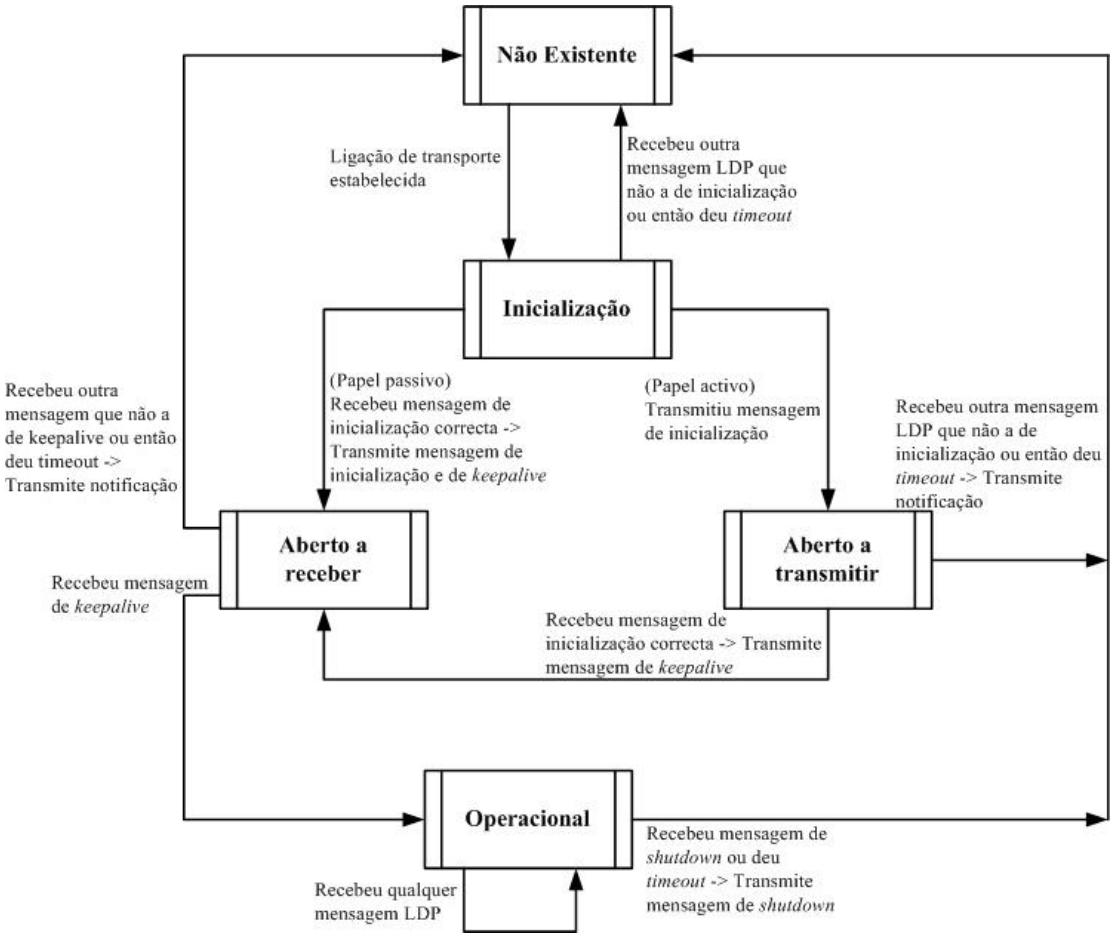


Figura 4.4 - Máquina de estados da inicialização de uma sessão LDP [5].

Com base nesta máquina de estados, foram testadas as acções tomadas pelo DUT em cada estado, com o auxílio de ficheiros de *logs*. Estes *logs* podem ser activados através do comando "**trace all**" no menu "**mpls ldp**". O ficheiro que contém estes *logs* deve ser definido no ficheiro de configurações do processo ldpd, através da sintaxe "**log file <file name>**".

Se no processo de inicialização da sessão LDP alguma coisa correr mal, o DUT deverá enviar uma notificação e fechar a ligação TCP. Com base no esquema da Figura 4.4, as situações que permitem abortar a inicialização da sessão LDP são as seguintes:

- Receber outra mensagem que não a mensagem de inicialização quando está no estado de "Inicialização".
- Receber outra mensagem que não a mensagem de *keepalive* quando está no estado de "Aberto a receber" (estado passivo).
- Receber outra mensagem que não a mensagem de inicialização quando está no estado de "Aberto a transmitir" (estado activo).
- Estiver no estado "Operacional" e receber uma notificação de *Shutdown*.
- Estiver no estado "Operacional" e o temporizador *keepalive* expirar por não receber nenhuma mensagem LDP.

Estas situações foram reproduzidas com o auxílio do N2X, gravando previamente várias sequências de mensagens LDP, que permitem posteriormente validar cada um dos pontos anteriores.

4.3.3. Teste aos modos de funcionamento

Após as sessões LDP estabelecidas, o procedimento de distribuição de etiquetas não será o mesmo para os diferentes modos de funcionamento presentes em cada cenário. Foram testados todos os modos de funcionamento presentes nas tabelas Tabela 4.1 e Tabela 4.2, mantendo sempre o objectivo da conectividade entre as duas interfaces do N2X. De seguida, é realizado um resumo do que acontece para cada modo de configuração. Foi apenas considerado o modo de distribuição (*On Demand* e *Unsolicited*) e controlo de etiquetas (Ordenado e Independente), pois são os únicos que influenciam a distribuição de etiquetas. O modo de retenção (Conservativo ou Liberal) apenas decide se um determinado mapeamento deve ser mantido ou não. Foi usado como exemplo o cenário da Figura 4.1, e um FEC no sentido N2X-Cisco.

- ***Downstream On Demand*, Ordenado (Modos: 1, 8)**

Neste modo, os LSR de *upstream* enviam pedidos de etiquetas aos LSR de *downstream*. Por sua vez, estes respondem com os seus mapeamentos para os LSR de *upstream*. Cada LSR só envia o seu mapeamento depois de receber o pedido do LSR de *upstream* e o mapeamento do LSR de *downstream*. Estes passos estão exemplificados na Figura 4.5, que é uma parte do cenário da Figura 4.1.

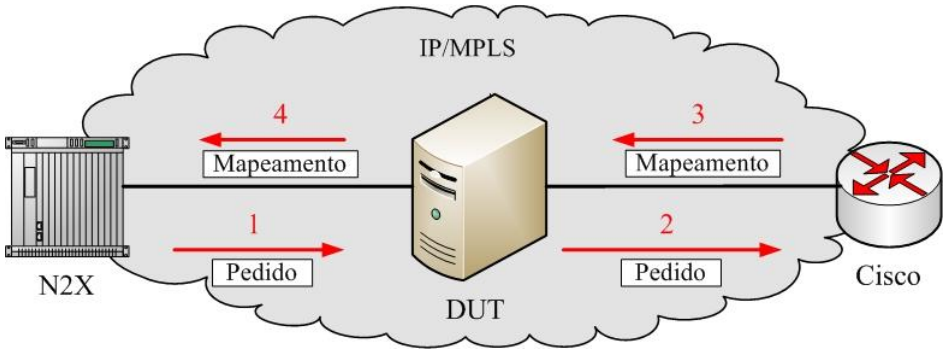


Figura 4.5 – Modo *Downstream On Demand*, Ordenado.

1. O N2X, que neste caso é o LER de entrada, começa por enviar pedidos de etiqueta para o DUT.
2. O DUT recebe estes pedidos e envia novos pedidos para o Cisco.
3. O Cisco, que neste caso é o LER de saída, responde aos pedidos com mapeamentos de etiquetas.
4. O DUT recebe estes mapeamentos e envia para o N2X os seus mapeamentos.

- ***Downstream On Demand*, Independente (Modos: 6, 7, 9, 10)**

A diferença deste modo para o anterior prende-se com o facto do LSR que recebe o pedido poder enviar logo de seguida os seus mapeamentos sem esperar pelos mapeamentos do LSR de *downstream*. A Figura 4.6 exemplifica as acções tomadas por cada LSR neste modo de configuração.

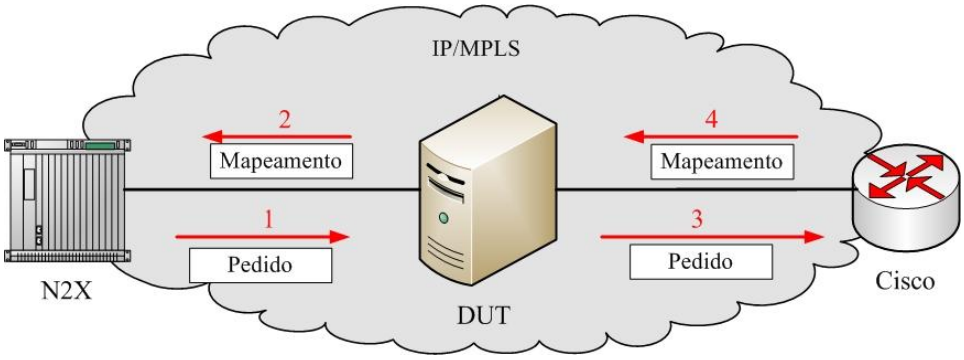


Figura 4.6 - Modo *Downstream On Demand*, Independente.

1. O N2X, que neste caso é o LER de entrada, começa por enviar pedidos de etiqueta para o DUT.
2. O DUT recebe estes pedidos e poderá enviar os seus mapeamentos para o N2X.

3. Simultaneamente com o ponto anterior o DUT envia pedidos de etiqueta para o Cisco.
4. O Cisco, que neste caso é o LER de saída, responde aos pedidos com mapeamentos de etiquetas.

Se a detecção de *loops* estiver activada, após o DUT receber os mapeamentos do Cisco, deve enviar novos mapeamentos para o N2X.

- ***Downstream Unsolicited*, Ordenado (Modos: 2, 5)**

Neste modo, os LSR não necessitam de receber pedidos de etiqueta dos seus LSRs de *upstream* para poderem enviar os seus mapeamentos. No entanto, antes os enviar, precisam antecipadamente de receber os mapeamentos dos seus LSRs de *downstream*. Estes passos estão exemplificados na Figura 4.7.

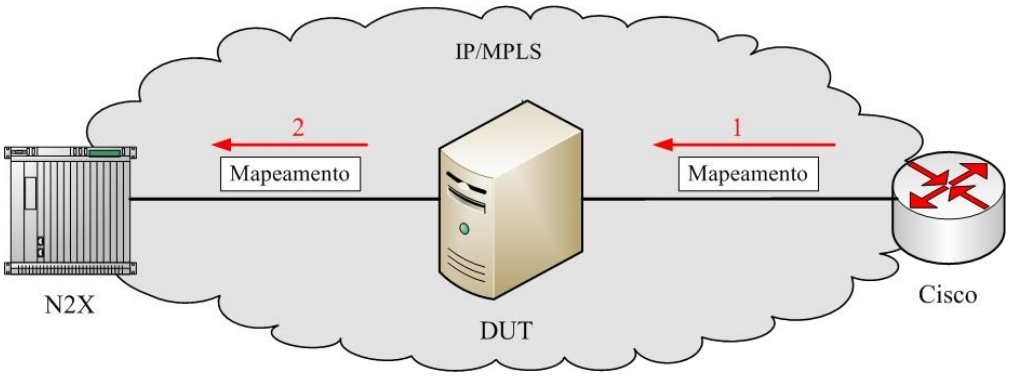


Figura 4.7 - Modo *Unsolicited*, Ordenado.

1. O Cisco, sendo o LER de saída, começa por enviar os seus mapeamentos para o DUT.
2. O DUT recebe estes mapeamentos e poderá enviar os seus mapeamentos para o N2X.

Neste caso, a distribuição de etiquetas é despoletada a partir do Cisco, que é o LER de saída.

- ***Downstream Unsolicited*, Independente (Modos: 3, 4)**

Neste modo, os LSRs funcionam independentemente uns dos outros. Ou seja, a decisão de enviar mapeamentos para os LSRs de *upstream* é realizada individualmente em cada LSR. Assim que um LSR reconheça um FEC poderá atribuir-lhe uma etiqueta e enviar esse mapeamento para o seu

vizinho. A Figura 4.8 ilustra isso mesmo, não havendo dependências nas acções a tomar por cada LSR.

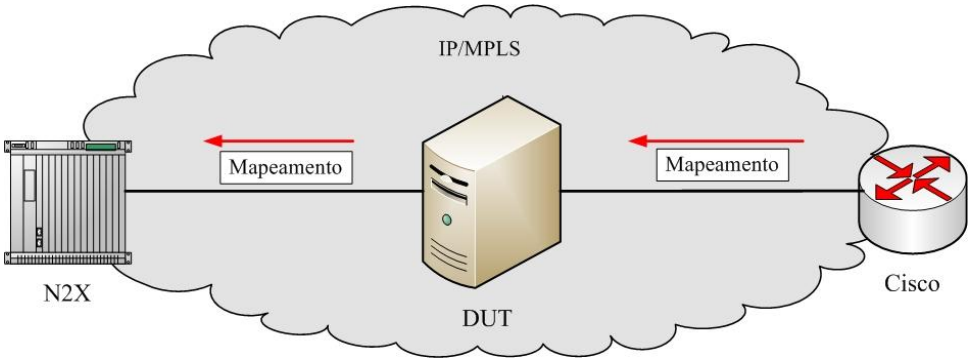


Figura 4.8 - Modo *Unsolicited*, Independente.

Cada uma das situações anteriores foi usada para os testes que serão realizados de seguida.

4.3.3.1. Teste ao LER

O DUT pode desempenhar funções de LER e/ou LSR, dependendo do FEC em questão. Esta secção avalia o comportamento do DUT quando este realiza funções de LER. Na Figura 4.9 o DUT situa-se na fronteira da rede MPLS. Depois dos LSPs estabelecidos, podem circular pacotes MPLS na rede 172.29.0.0/24 onde o DUT realiza funções de LER de entrada e saída. Este teste tem como objectivo avaliar o comportamento do DUT quando tem de distribuir etiquetas na função de LER.

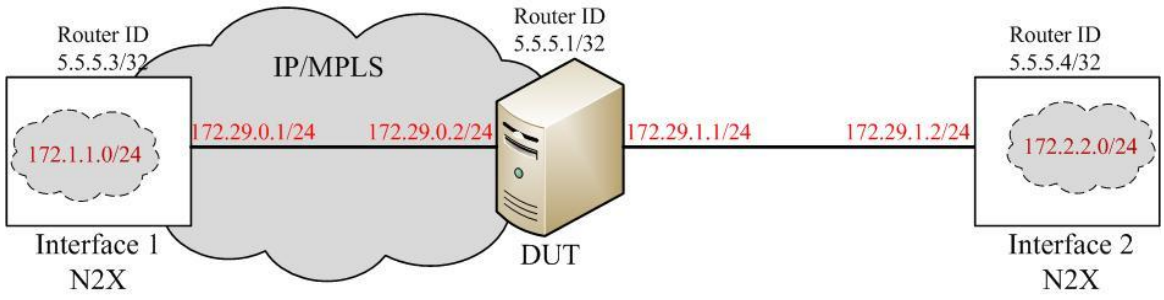


Figura 4.9 - Cenário para teste de um LER.

Neste teste as interfaces 1 e 2 do N2X foram configuradas para anunciar redes externas ao cenário. A interface 1 anuncia a rede 172.1.1.0/24, ou seja, para o DUT o endereço do próximo salto usado para atingir a esta rede é o endereço da interface 1 do N2X (172.29.0.1). A interface anuncia igualmente a rede 172.2.2.0/24. Desta forma, é possível avaliar o estabelecimento de LSPs de redes não directamente ligadas aos routers MPLS.

Neste teste, o DUT estabelece apenas uma sessão LDP com a interface 1 do N2X. Dentro desta sessão LDP são trocadas, entre os dois equipamentos, mensagens de pedido e mapeamento com o objectivo de definir quais as etiquetas associadas a cada FEC. As etiquetas atribuídas a cada FEC, locais e remotas, poderão ser visualizadas no DUT em "**show ldp database**". As etiquetas usadas no encaminhamento dos pacotes MPLS poderão ser consultadas no DUT em "**show mpls forwarding**". O comando "**show ip route**" permite visualizar a tabela de encaminhamento. Se para um dado destino o DUT tiver uma etiqueta de saída associada, ela irá aparecer na tabela de encaminhamento.

Independentemente dos modos de funcionamento configurados, no cenário da Figura 4.9 devem ser estabelecidos os seguintes LSPs:

- LSP no sentido Interface 1 – DUT para o FEC 172.2.2.0/24;
- LSP no sentido Interface 1 – DUT para o FEC 172.29.1.0/24;
- LSP no sentido Interface 1 – DUT para o FEC 172.29.0.0/24;
- LSP no sentido DUT – Interface 1 para a rede 172.1.1.0/24;
- LSP no sentido DUT – Interface 1 para a rede 172.29.0.0/24.

4.3.3.2. Teste ao LSR

A Figura 4.10 apresenta o cenário usado quando se pretende testar o DUT com funções de LSR. Neste caso, o DUT realiza comutação de etiquetas e o equipamento N2X faz a fronteira do domínio MPLS, através das suas duas interfaces.

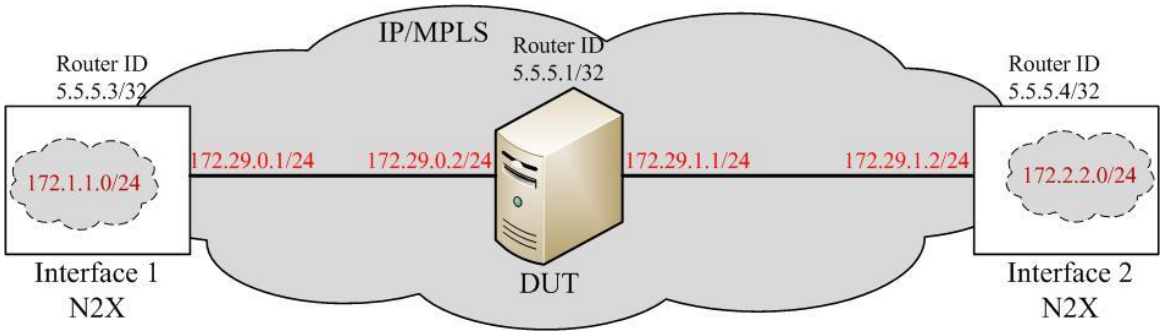


Figura 4.10 - Cenário para teste de um LSR.

No cenário anterior o DUT estabelece duas sessões LDP; uma por cada interface do N2X. Com esta configuração os LSPs que devem ser estabelecidos são os seguintes:

- LSP no sentido Interface 1 – Interface 2 para o FEC 172.2.2.0/24;
- LSP no sentido Interface 1 – DUT para o FEC 172.29.1.0/24;

- LSP no sentido Interface 1 – DUT para o FEC 172.29.0.0/24;
- LSP no sentido Interface 2 – Interface 1 para a rede 172.1.1.0/24;
- LSP no sentido Interface 2 – DUT para a rede 172.29.0.0/24;
- LSP no sentido Interface 2 – DUT para a rede 172.29.1.0/24;

Além de realizar funções de LSR para as redes 172.1.1.0/24 e 172.2.2.0/24, o DUT realiza funções de LER para as redes que lhe estão directamente ligadas. Ou seja, para pacotes com destino para as redes 172.29.0.0/24 e 172.29.1.0/24, o DUT insere e remove etiquetas nestes pacotes.

4.3.3.3. RIP e BGP

O objectivo deste teste é validar a capacidade do DUT em estabelecer LSPs com base em rotas descobertas por vários protocolos de encaminhamento (OSPF, RIP, BGP). Até aqui, todos os testes especificados utilizaram o protocolo OSPF. Nestes testes pretende-se validar o estabelecimento de LSPs quando as rotas são descobertas pelos protocolos RIP e BGP. Além do OSPF, estes são os únicos protocolos suportados pela plataforma *quagga*. Esta plataforma faz parte integrante da solução a testar. É ela que realiza o processamento dos vários protocolos de encaminhamento, sobre os quais corre o LDP. Assim, torna-se extremamente importante validar a habilidade da solução em estabelecer LSPs sobre os vários protocolos.

Os resultados deste teste têm de ser semelhantes aos resultados verificados nos testes aos modos de funcionamento, LER e LSR.

O RIP pertence à classe de protocolos baseados em algoritmos de encaminhamento do tipo vector-distância, que usam um contador de saltos como métrica. É um protocolo do tipo IGP, o que quer dizer que actua apenas dentro dum sistema autónomo [28]. O RIP é ideal para redes pequenas que não tenham mais de 16 saltos.

O *daemon ripd* tem de ser lançado e a forma como é configurado o RIP no *Quagga* é muito semelhante ao OSPF. A configuração básica efectuada no *Quagga* para os cenários das Figura 4.9 e Figura 4.10 é a seguinte:

```
#Activar RIP
router rip
#Activar RIP por interface
network 192.168.0.0/24
network 192.168.1.0/24
```

O BGP é um protocolo do tipo *Exterior Gateway Protocol* (EGP), o que quer dizer que é configurado entre múltiplos sistemas autónomos, trocando informação de encaminhamento e

acessibilidade com outros sistemas BGP [28]. Como exemplifica a Figura 4.11, o BGP pode ser usado para interligar vários sistemas autónomos. Estes sistemas podem correr outro protocolo de encaminhamento ou mesmo o próprio BGP. Nesta figura os *Core Router* utilizam o BGP para encaminhar o tráfego entre os vários sistemas autónomos.

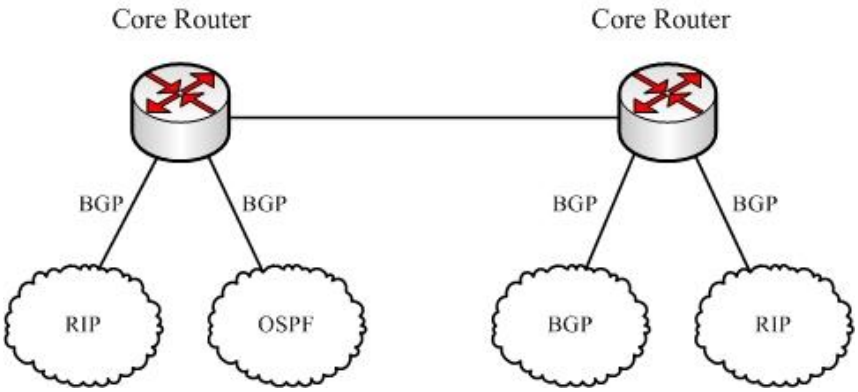


Figura 4.11 - Exemplo da utilização do BGP [28].

Como qualquer protocolo de encaminhamento, o BGP gere as tabelas de encaminhamento, transmite actualizações de encaminhamento, e baseia as suas decisões de encaminhamento nas métricas. A função primária dum sistema BGP é trocar informação de acessibilidade, incluindo a informação da lista de caminhos dos sistemas autónomos, com outros sistemas BGP [28].

Para o BGP, a configuração no *Quagga* passa pela definição dos sistemas autónomos a que pertence o DUT e os seus vizinhos. Neste caso, o DUT foi configurado para pertencer ao sistema autónomo 1 (**router bgp 1**). A interface 1 do N2X pertence ao sistema autónomo 2 (**neighbor 172.29.0.1 remote-as 2**) e a interface 2 pertence ao sistema 3 (**neighbor 172.29.1.2 remote-as 3**). O DUT deve ser capaz de estabelecer LSPs sobre este cenário.

```
#Activar BGP
router bgp 1
#Activar BGP por interface
neighbor 172.29.0.1 remote-as 2
neighbor 172.29.1.2 remote-as 3
```

4.3.4. Detecção de loops

O método de detecção de *loops* no LDP está incorporado nas mensagens de pedido de etiqueta e mapeamento de etiqueta. Nestas mensagens podem ser usados os campos *Path Vector* e *Hop Count* para detecção de *loops* [5]. Com base nisto, foram realizados testes para estas quatro combinações:

pedido de etiqueta usando o campo *Path Vector* ou o campo *Hop Count*, e mapeamento de etiqueta usando o campo *Path Vector* ou o campo *Hop Count*.

O cenário usado é mostrado na Figura 4.12. O objectivo deste cenário é enviar, através do N2X, mensagens de pedido e mapeamento de etiqueta especiais para o DUT. Estas mensagens são construídas no N2X e quando o DUT as recebe é levado a crer que está numa situação de *loop*. Isto é conseguido preenchendo o campo *Path Vector* com o LSR ID do DUT (5.5.5.1) ou o campo *Hop Count* com o valor máximo suportado pelo DUT (255). Assim que o DUT recebe estas mensagens este deve detectar a presença do *loop*, descartar a mensagem e enviar uma notificação de "*loop* detectado" [5].

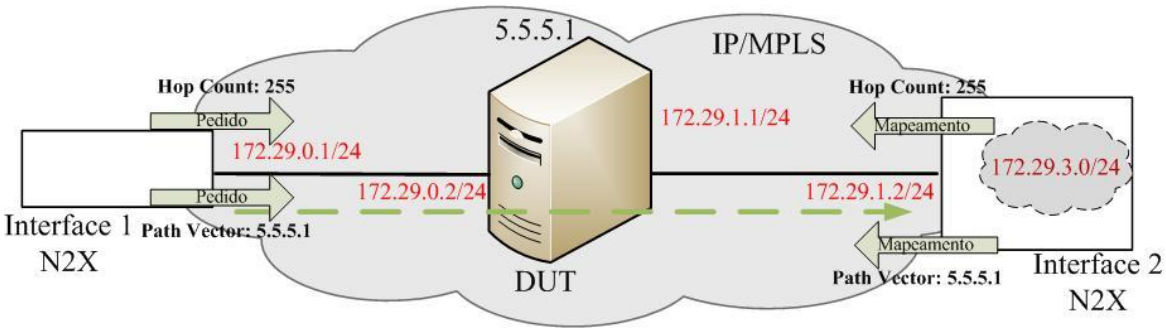


Figura 4.12 - Cenário de teste para detecção de *loops*.

No cenário da figura anterior, o OSPF foi desactivado. No DUT foi configurada uma rota para a rede 172.29.3.0/24 em que o próximo salto é o endereço da interface 2 do N2X (172.29.1.2). Desta forma, os mapeamentos de etiquetas enviados devem incluir o FEC 172.29.3.0/24 no sentido interface 2 – interface 1 do N2X. Igualmente, os pedidos de etiquetas devem incluir o mesmo FEC com o sentido interface 2 – interface 1.

Ambas as interfaces do N2X têm a detecção de *loops* desactivada. No DUT esta funcionalidade está disponível em três formas: detecção de *loops* por *Hop Count*, detecção de *loops* por *Path Vector*, e ambas. O comando "**loop-detection-mode [both | hop | path]**", no menu "**mpls ldp**" permite activar cada uma das opções.

Este teste passa por, para cada uma das três configurações anteriores, avaliar se o DUT é capaz de detectar *loops* através das mensagens de pedido ou mapeamento de etiqueta.

4.3.5. Label Space

A noção de *Label Space* define se um determinado LSR pode atribuir a mesma etiqueta a FECs e interfaces diferentes [2]. No DUT o *Label Space* é um identificador (número) configurável por

interface. Se duas interfaces tiverem o mesmo *Label Space*, nunca poderão atribuir etiquetas iguais. Se pelo contrário duas interfaces do DUT tiverem *Label Spaces* diferentes, poderão atribuir etiquetas iguais. O *Label Space* é configurável no DUT através do comando "**mpls labelspace** <value>" dentro do menu de interfaces. Não se pretende com este teste realizar uma validação extensiva aos dois modos de *Label Space*. Apenas foi realizado um pequeno teste que permite verificar se o DUT é capaz de distinguir estes dois modos.

As duas situações anteriores foram testadas com o auxílio do cenário apresentado na Figura 4.13 e na Figura 4.14. Na Figura 4.13 foi configurado o *Label Space* por plataforma, em que o DUT distribui etiquetas diferentes para os dois FECs existentes – 172.29.0.0/24 e 172.29.1.0/24. Assim, o DUT envia um mapeamento para o N2X, contendo uma etiqueta L1 para o FEC 172.29.1.0/24, e um mapeamento para o Cisco contendo uma etiqueta L2, diferente de L1, para o FEC 172.29.0.0/24.

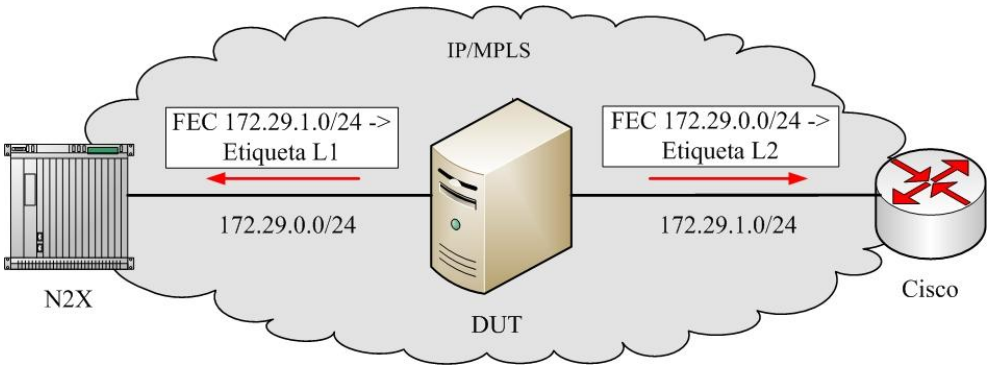


Figura 4.13 - Teste ao *Label Space* por plataforma.

Na Figura 4.14 o DUT distribui etiquetas iguais aos dois FECs existentes. Uma vez que estes FECs são anunciados em interfaces diferentes, e se estas duas interfaces tiverem *Label Spaces* distintos, o DUT poderá atribuir a mesma etiqueta L1 aos dois FECs.

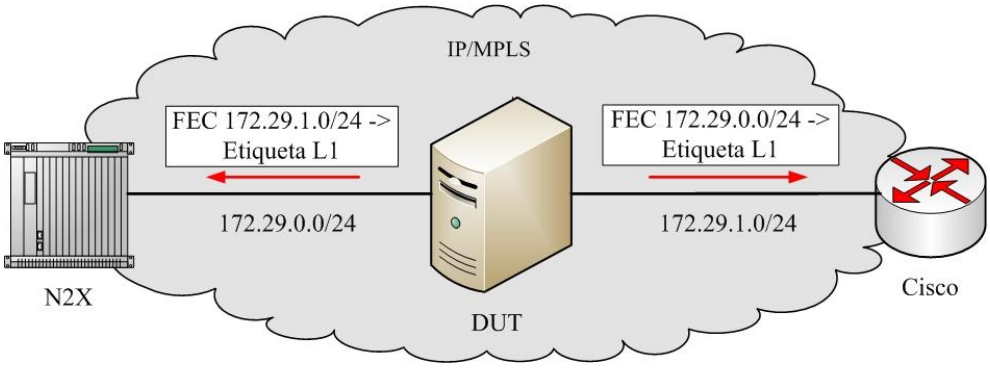


Figura 4.14 - Teste ao *Label Space* por interface.

4.4. Apresentação e discussão de resultados

Os testes funcionais revelaram algumas lacunas existentes na solução LDP do "*MPLS for Linux*". Apesar de esta solução conseguir desempenhar algumas funções básicas, como o estabelecimento de sessões LDP e a distribuição de etiquetas, apresenta algumas não conformidades ou até deficiências na implementação. De seguida, os resultados serão apresentados por tópicos, à semelhança do planeamento de testes realizado anteriormente.

Mensagens LDP:

Os resultados obtidos nos testes às mensagens LDP encontram-se resumidos na tabela do 0. Todas as mensagens são suportadas pela solução, no entanto foram detectados os problemas apresentados de seguida.

A solução apresenta problemas quando esta é configurada para não realizar *merge* de etiquetas, modos de funcionamento 8, 9 e 10. Sempre que um destes modos era configurado o *daemon ldpd* morre, inviabilizando a recepção e transmissão de mensagens LDP. No entanto, estes não são os modos mais aconselhados para o transporte de IP, de acordo com [5]. Como os testes apenas incidiram no transporte de IP, esta lacuna foi considerada não grave.

As mensagens de *hello* não apresentaram nenhuns problemas. O DUT envia e recebe estas mensagens. Nos *hellos* recebidos, o DUT compara o valor dos temporizadores com os seus e adopta o que for menor. Se receber um valor de temporizador nulo, o DUT ajusta-o para 15 segundos. O DUT ignora o campo *Reserved* existente nestas mensagens.

As mensagens de inicialização também não apresentaram qualquer problema. De referir que, quando os modos de distribuição (*on demand* e *unsolicited*) configurados no DUT e N2X não coincidem, o N2X envia uma notificação para encerramento da sessão LDP. Na interface que interliga o DUT ao Cisco isto já não acontece, revelando que o DUT ajusta o modo de distribuição de acordo com o seu par LDP.

As mensagens de *keepalive* são enviadas e processadas pelo DUT de acordo com a periodicidade configurada.

Em relação às mensagens de endereço, apenas há a salientar o facto de o DUT não enviar uma notificação apropriada (*Unsupported Family*) quando recebe uma mensagem de endereço com família de endereços diferente de IPv4. Quanto às mensagens de remoção de endereço, não se verificou qualquer problema. O DUT envia estas mensagens com todos os endereços IP configurados no equipamento.

Na resposta às mensagens de pedido de etiqueta (através das mensagens de mapeamento), o DUT não está a incluir o TLV que permite identificar qual a mensagem pedido correspondente. Este TLV deve ser incluído quando a mensagem de mapeamento é enviada como resposta a um pedido. Como o DUT não o inclui, o N2X descarta a mensagem enviando uma mensagem de libertar etiqueta.

No modo de controlo ordenado, o DUT é incapaz de estabelecer LSPs quando desempenha funções de LER de saída. Nesta função o DUT deve ser o primeiro a enviar os mapeamentos de etiqueta, no entanto apenas envia para as redes que lhe estão directamente ligadas. No modo independente o DUT funciona como esperado.

Quando o DUT recebe um mapeamento de etiqueta com o TLV FEC preenchido com *wildcard*, deveria transmitir uma notificação ou uma mensagem de libertar etiqueta. Em vez disso, o *daemon ldpd* morre, mostrando que esta implementação não está preparada para esta situação. O TLV FEC com *wildcard* deve apenas ser usado nas mensagens de libertar e remover etiqueta. No entanto, como o TLV é o mesmo que o TLV usado nos mapeamentos, o DUT deve estar preparado para estas mensagens.

As mensagens de pedido de etiqueta são as que mais apresentam problemas. No modo de distribuição *on demand*, o DUT quando desempenha funções de LER de entrada deve enviar pedidos de etiqueta quando reconhece um novo FEC. De acordo com os testes realizados, o DUT não corresponde com o esperado, não enviando qualquer mensagem de pedido. A única situação observada, em que o DUT envia mensagens de pedido de etiqueta, é quando este desempenha funções de LSR e reencaminha um pedido que acabou de receber.

As mensagens de abortar pedido de etiqueta apresentam problemas semelhantes às mensagens de pedido de etiqueta. O DUT processa estas mensagens e suspende o estabelecimento do LSP, mas como resposta envia uma notificação de "*fatal error*" que reinicia a sessão LDP.

Quando o DUT recebe uma mensagem de remover etiqueta, indicando qual a etiqueta a remover, este remove-a com sucesso. No entanto, esta mensagem pode chegar com o selo de *wildcard*, indicando que é para remover todas as etiquetas. O *daemon ldpd* morre quando recebe estas mensagens. As mensagens de libertar etiqueta apenas apresentaram problemas no modo *on demand*. Neste modo, o *daemon ldpd* também morre quando o DUT receber uma nova etiqueta para um FEC que já tem LSP estabelecido.

As mensagens de notificação apresentam uma implementação algo deficiente. Algumas mensagens não estão suportadas. Nas diversas situações testadas, o DUT envia quase sempre a mesma notificação de "*fatal error*" reiniciando a sessão LDP.

Descoberta e estabelecimento de sessões LDP:

As várias etapas de estabelecimento de sessões LDP são cumpridas na generalidade pelo DUT. A descoberta básica de pares LDP, a decisão de qual o LSR activo ou passivo, o estabelecimento da ligação de transporte (TCP), a inicialização e a máquina de estados são etapas que a solução LDP cumpre de acordo com [5]. Alguns dos testes não foram possíveis de realizar, uma vez que o N2X não permite o envio de mensagens LDP formatadas antes da sessão LDP estar estabelecida. Deste modo, a recepção de mensagens de inicialização inválidas e os erros que poderão acontecer nas várias etapas da máquina de estados não puderam ser testadas como planeado.

De qualquer forma, o DUT consegue sem problemas descobrir outros LSRs que lhe sejam vizinhos. Consegue decidir, através dos endereços de transporte, qual o LSR activo e passivo, e cumpre com as várias etapas da máquina de estados, estabelecendo com sucesso as sessões LDP.

Teste aos modos de funcionamento:

Os testes aos modos de funcionamento revelaram algumas lacunas, já perceptíveis nos testes às mensagens LDP. A Tabela 4.7 mostra os resultados para estes testes. Como se pode ver, alguns modos de funcionamento não conseguem estabelecer LSPs e outros terminam mesmo o *daemon ldpd*.

Modos de funcionamento	Teste ao LSR	Teste ao LER
Unsolicited Ordenado	Sem problemas.	Não está a enviar mapeamentos para redes não directamente ligadas.
Unsolicited Independente	Sem problemas.	Sem problemas.
On demand Independente	Problemas nos pedidos de etiquetas.	*Daemon ldpd* morre.
On demand Ordenado	Problemas nos pedidos de etiquetas.	*Daemon ldpd* morre.

Tabela 4.7 - Resultados dos testes aos modos de funcionamento.

Para o modo *unsolicited* ordenado, o DUT comporta-se melhor quando desempenha funções de LSR. As tabelas de mapeamentos seguintes mostram as diferenças quando o DUT desempenha funções de LSR (à esquerda) ou funções de LER (à direita). Estas tabelas mostram as associações

entre FECs e etiquetas realizadas pelo DUT (*local binding*), assinaladas a azul, e as associações realizadas pelos seus LSRs vizinhos (*remote binding*), assinaladas a amarelo.

```
dut# show ldp database
  5.5.5.1/32       local binding: label: gen 10000
  172.29.0.0/24  local binding: label: gen 10002
  172.29.1.0/24  local binding: label: gen 10003
  5.5.5.1/32       local binding: label: gen 10004
  172.29.0.0/24  local binding: label: gen 10006
  172.29.1.0/24  local binding: label: gen 10007
  172.1.1.0/24  remote binding: label: gen 25 lsr:
172.29.0.1:0 ingress
  172.1.1.0/24    local binding:   label: gen 10008
  172.2.2.0/24 remote binding:  label: gen 25 lsr:
172.29.1.2:0 ingress
  172.2.2.0/24    local binding:   label: gen 10009
```

```
dut# show ldp database
  5.5.5.1/32       local binding: label: gen 10003
  172.29.0.0/24  local binding: label: gen 10004
  172.29.1.0/24  local binding: label: gen 10005
  172.1.1.0/24   remote binding: label: gen 34 lsr:
172.29.0.1:0 ingress
```

De acordo com as Figura 4.9 e Figura 4.10, os FECs 172.1.1.0/24 e 172.2.2.0/24 são redes que não se situam dentro da rede MPLS e não fazem fronteira com esta. Para o modo *unsolicited ordenado*, a principal diferença entre estes dois cenários está nas etiquetas distribuídas. Na função de LSR, o DUT atribui uma etiqueta por cada FEC existente na sua tabela de encaminhamento e distribuí aos seus dois vizinhos (as duas interfaces do DUT). Para as redes 172.1.1.0/24 e 172.2.2.0/24, o DUT espera pelas etiquetas dos seus vizinhos de *downstream* e depois atribui uma etiqueta a cada uma destas redes. Desta forma, o DUT tem uma etiqueta de entrada (*local binding*) e uma de saída (*remote binding*) para cada rede. A Figura 4.15 mostra como é realizado o encaminhamento após a distribuição de etiquetas das tabelas anteriores.

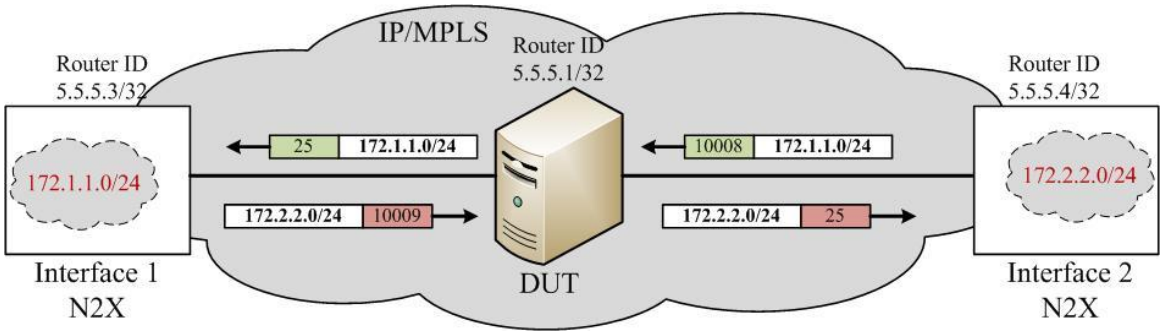


Figura 4.15 - Encaminhamento MPLS após distribuição de etiquetas com o DUT na função de LSR.

A etiqueta 25 é atribuída por cada interface do N2X. As interfaces do N2X funcionam como se fossem routers independentes, por isso atribuem a mesma etiqueta a FECs diferentes. O DUT, que funciona como LSR neste cenário, atribui as etiquetas 10008 à rede 172.1.1.0/24 e 10009 à rede 172.2.2.0/24, e distribui-as às interfaces 2 e 1, respectivamente. Deste modo, são estabelecidos dois LSPs, um para cada rede.

Na função de LER, o DUT deveria atribuir uma etiqueta às redes para as quais é o LER de saída. No cenário da Figura 4.9, o DUT deveria atribuir uma etiqueta à rede 172.2.2.0/24. Como se pode observar nas tabelas de mapeamento anteriores, na tabela do lado direito o DUT recebeu as etiquetas da interface 1 do N2X para a rede 172.1.1.0/24 e apenas atribuiu etiquetas às redes que lhe estão directamente ligadas (estas redes são todas as que estão do lado direito excepto a rede 172.1.1.0/24 que está a amarelo).

No modo *on demand* independente ou ordenado, o DUT não consegue estabelecer LSPs. Como LSR, o DUT tem o problema dos pedidos de etiqueta referido anteriormente. Os mapeamentos, que o DUT envia como resposta aos pedidos das interfaces do N2X, não contêm os TLVs que identificam o pedido e por isso o N2X descarta-os. Como LER, o DUT assim que recebe um pedido da interface 1 do N2X, provoca a morte do *daemon ldpd*. Foi notado que, quando o DUT recebe um pedido ou um mapeamento para um FEC que não consta na sua tabela de encaminhamento, este provoca igualmente a morte do *daemon ldpd*.

O modo *unsolicited* independente funciona sem qualquer problema, quer o DUT desempenhe as funções de LSR ou LER. As etiquetas distribuídas são semelhantes às distribuídas no modo *unsolicited* ordenado.

Os resultados para os testes com os protocolos de encaminhamento BGP e RIP foram em tudo semelhantes aos testes com o protocolo OSPF, revelando assim que o LDP apenas se interessa pelas rotas descobertas e não por "quem" as descobriu.

Detecção de *loops*:

Os testes à detecção de *loops* revelaram que a detecção por *Path Vector* não está a funcionar. O DUT quando recebe mensagens LDP com o TLV *Path Vector* preenchido com o seu próprio endereço IP, não detecta a presença de *loop*. A Tabela 4.8 mostra os resultados do teste à detecção de *loops* com todas as combinações.

Mensagens LDP	***Hop Count***	***Path Vector***	**Ambos**
Pedido de etiqueta	Sem problemas.	Com problemas.	*Hop Count* sem problemas, *Path Vector* com problemas.
Mapeamento de etiqueta	Sem problemas.	Com problemas.	*Hop Count* sem problemas, *Path Vector* com problemas.

Tabela 4.8 - Resultados dos testes à detecção de *loops*.

A detecção por *Hop Count* funciona sem problemas. Se o DUT tiver a detecção de *loops* activada por *Hop Count* realiza uma das duas opções: incrementa o valor do TLV *Hop Count* se este ainda não tiver excedido o máximo, ou se este for igual ao máximo descarta a mensagem. Quando o DUT detecta um *loop* através das mensagens de pedido de etiqueta envia uma notificação para assinalar o *loop*. Quando o DUT detecta um *loop* através das mensagens de mapeamento de etiqueta envia uma mensagem de libertar etiqueta.

Com ambos os modos configurados, o DUT comporta-se de forma semelhante, ou seja, por *Hop Count* consegue detectar *loops* e por *Path Vector* não.

Label Space:

Os testes ao *Label Space* revelaram que o DUT só consegue estabelecer sessões LDP quando todas as suas interfaces se encontram dentro do mesmo *Label Space*. Desta forma, o DUT só suporta *Label Space* por plataforma. Com *Label Space* por interface, o DUT recusa o estabelecimento da sessão LDP enviando uma notificação de "Sessão Rejeitada".

Capítulo 5: Testes de desempenho e escalabilidade

Esta secção descreve os testes que permitem caracterizar o desempenho e escalabilidade da solução LDP. O desempenho e a escalabilidade são aspectos que frequentemente se complementam. O desempenho é caracterizado pela resposta do sistema quando submetido a determinadas necessidades. A escalabilidade é a capacidade do sistema em manter a disponibilidade e o desempenho do serviço à medida que a carga transaccional aumenta.

Os testes de escalabilidade permitem caracterizar a solução em termos de [25]:

- Capacidade: a capacidade dos routers em lidar com elevado número de LSPs e rotas; importante para quem dimensiona uma rede de telecomunicações.
- Débito de estabelecimento de LSPs: o débito máximo de estabelecimento de LSPs é um factor importante na resposta global duma rede.
- Estabilidade do protocolo de sinalização: os limites de um router MPLS que suporte protocolos de sinalização, tais como o LDP, são caracterizados pelo número de sessões que podem ser sustentadas simultaneamente.

Os testes de desempenho permitem averiguar o comportamento da solução quando inserida num ambiente aproximado ao duma rede de telecomunicações. O objectivo é testar a solução quando sujeita a carga, estabelecendo grandes quantidades de LSPs com elevada quantidade de informação de encaminhamento e sinalização. Pretende-se também testar a solução quando sujeita a falha nas ligações.

O desempenho e a escalabilidade da solução LDP foram comparados com os equipamentos de outro fabricante, nomeadamente o já utilizado anteriormente, o Cisco. Dado que este fabricante actualmente possui equipamentos a operar em redes IP/MPLS, estas comparações são muito importantes porque avaliam as capacidades da solução para operar numa rede real. Os resultados dos testes ao DUT serão considerados positivos, se se aproximarem dos resultados do equipamento Cisco.

A plataforma de testes *Agilent* N2X foi o meio usado para simular uma rede real. Através dele consegue-se simular uma rede de routers interligados, em que todas as rotas são anunciadas ao DUT e posteriormente os LSPs estabelecidos.

5.1. Cenário de testes

O objectivo do cenário de testes é reproduzir em laboratório as condições existentes numa rede real de telecomunicações. O cenário usado é o da Figura 5.1, onde é utilizada a plataforma de testes N2X. Esta plataforma, além dos testes mais simples descritos no capítulo anterior, permite simular uma rede com vários routers como referido anteriormente.

Na figura cada interface do N2X liga a uma interface do DUT, envolvendo assim apenas estes dois equipamentos. Por cada interface do N2X é criada uma emulação OSPF, continuando este a ser o protocolo IGP escolhido. Dentro da emulação OSPF é criada uma sessão (*Session Router*), à qual, dentro de diversas opções, se pode associar uma rede de routers interligados por interfaces IP virtuais. Cada uma destas interfaces virtuais, que interligam os routers, vai estar numa rede IP distinta. Como todos estes routers estarão a correr o protocolo OSPF, todas estas redes vão ser anunciadas ao DUT através da sessão criada.

Posteriormente, o DUT pode ser configurado para funcionar como LSR activando o LDP em ambas as interfaces dos dois equipamentos (situação 2. da Figura 5.1) ou como LER activando o LDP em apenas duas interfaces, uma de cada equipamento que estejam ligadas entre si (situação 1. da Figura 5.1).

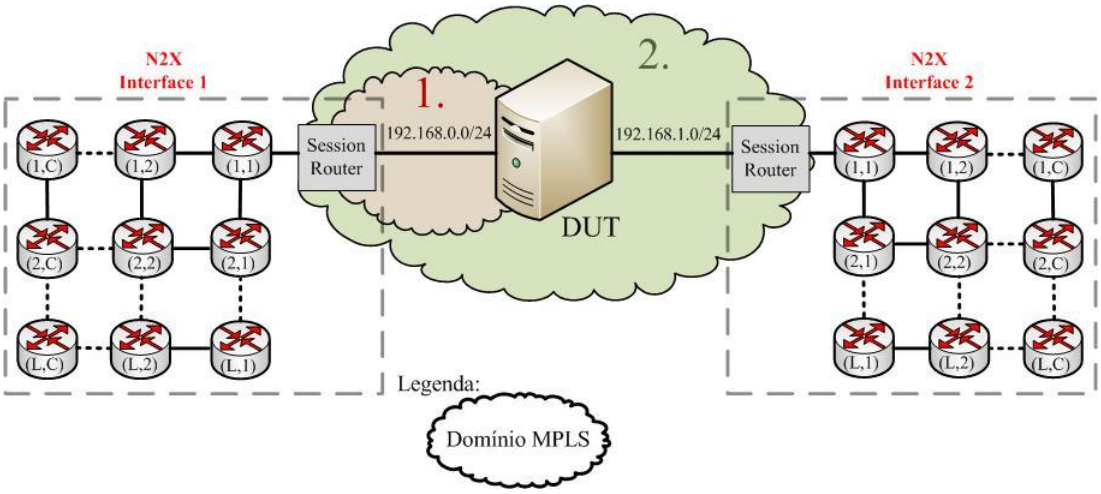


Figura 5.1 - Cenário para os testes de desempenho.

A rede de routers exemplificada pela Figura 5.1 é criada em cada interface da plataforma N2X através do menu da Figura 5.2. Este menu define o número de linhas (M) e colunas (N) que a rede de routers terá, bem como o identificador IP usado para identificar cada router na emulação OSPF. A quantidade de routers presentes na rede será então de $MN+1$. Por sua vez, a quantidade de rotas anunciadas ao DUT será igual à quantidade de interligações existentes na matriz, que pode ser obtida através da equação $2MN-M-N+1$. Nas equações, o router e a rota adicional é relativo ao *Session Router*, que faz interface com o DUT. Como exemplo, se considerarmos uma matriz com 16 linhas e 13 colunas, temos $16\times13+1=209$ routers e $2\times16\times13-16-13+1=388$ redes em que cada rede corresponde a uma rota anunciada.

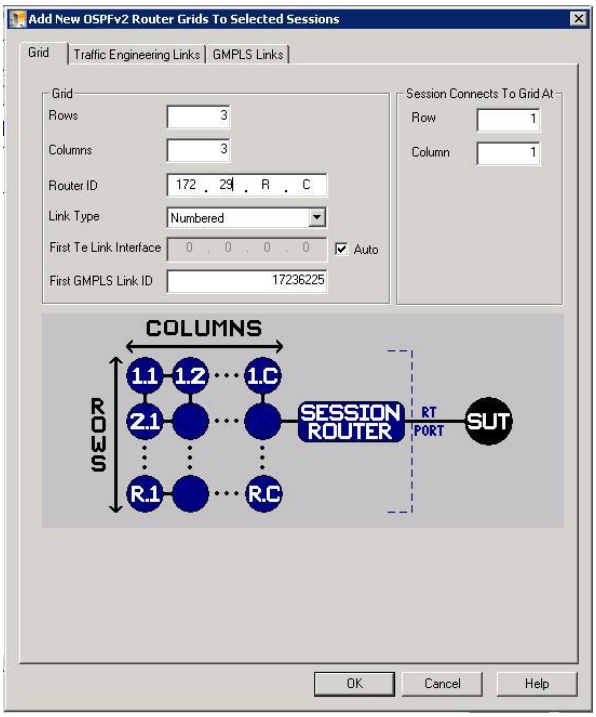


Figura 5.2 - Adicionar uma rede OSPF com vários routers no N2X.

Depois de indicar a quantidade de routers e rotas que a matriz irá conter, o menu da Figura 5.3 permite definir o endereço IP de cada interface dos diversos routers da matriz (lado direito da Figura 5.3) e o endereço de *host* de cada router (lado esquerdo da Figura 5.3).

Por fim, depois de todos os endereços definidos, o *Session Router* irá estabelecer uma sessão OSPF com o DUT e anunciar todas as rotas que permitem a este chegar a qualquer rede da matriz.

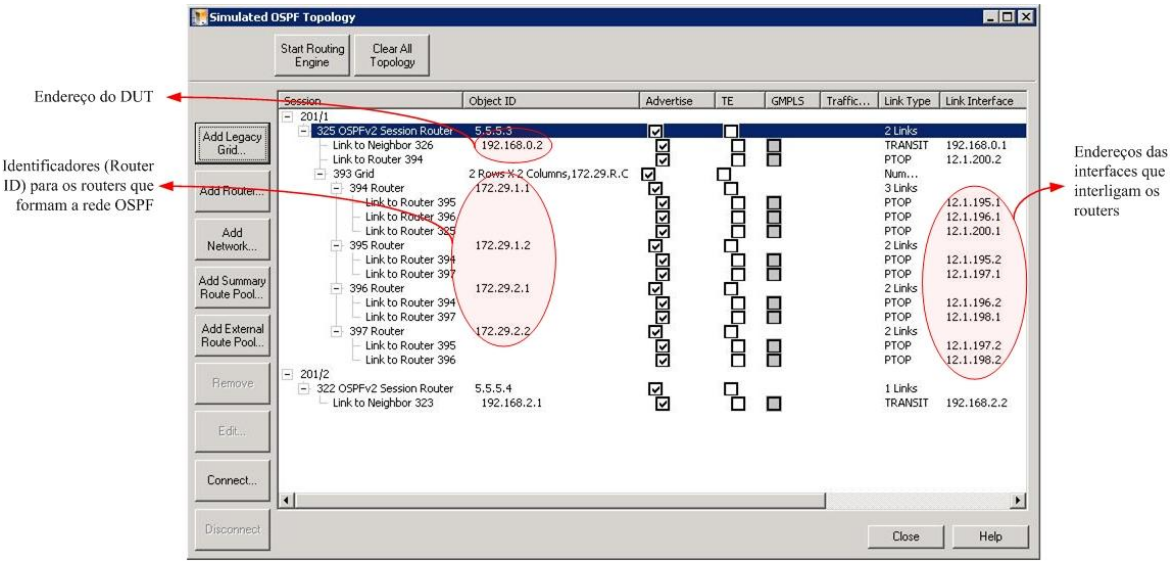


Figura 5.3 - Definir endereços na rede OSPF criada no N2X.

Após a comunicação IP ser estabelecida, é activado o LDP nas interfaces do DUT e do N2X. Nas sessões LDP do N2X, os LSPs não são estabelecidos automaticamente, quando este realiza funções de LER de saída. Por exemplo, para um LSP que se estabeleça entre as duas interfaces do N2X no sentido Interface 1 -> Interface 2, o DUT irá anunciar os seus mapeamentos para a Interface 1, sendo esta interface o LER de entrada. No entanto, a Interface 2, que realiza funções de LER de saída, não anuncia os seus mapeamentos, não sendo assim possível estabelecer o LSP entre as interfaces 1 e 2. Para resolver esta limitação é necessário adicionar o FEC respectivo à lista de *Egress Pool* no N2X.

Para adicionar FECs à lista de *Egress Pool* é preciso entrar na sessão LDP pretendida, seleccionar o menu "LSPs" e posteriormente a opção "*Add Egress LSP Pool*", como ilustra a Figura 5.4. Dentro deste menu são seleccionados os FECs pretendidos. Desta forma, o N2X passa a anunciar, na direcção do DUT, os mapeamentos respectivos, estabelecendo assim os LSPs.

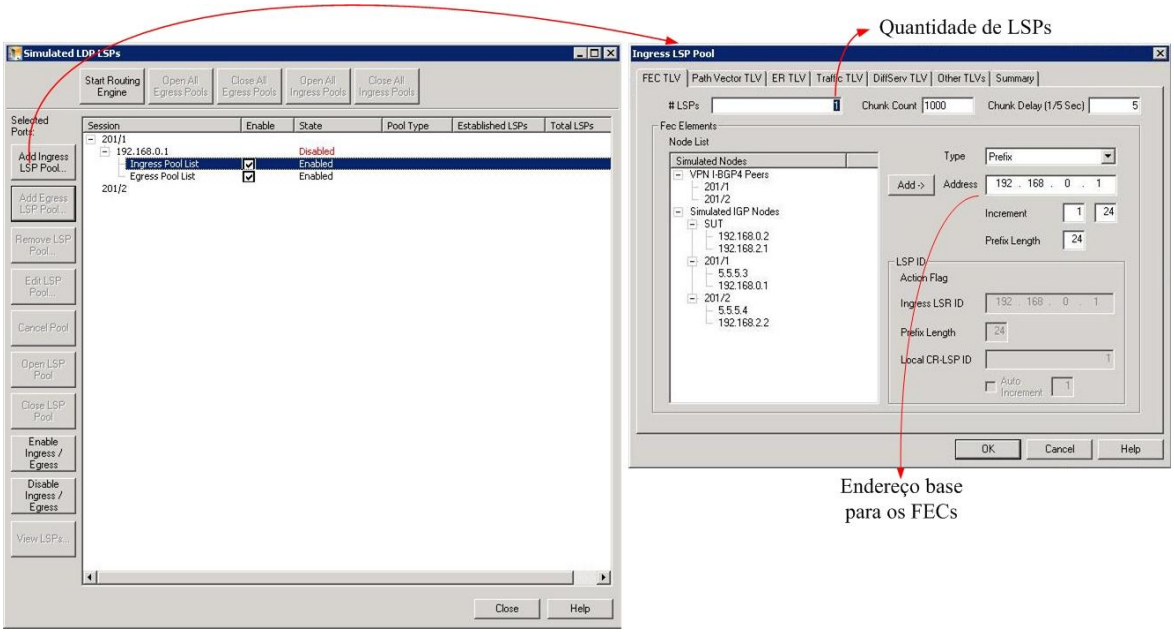


Figura 5.4 - Criar vários LSP no N2X.

Durante o estabelecimento das sessões LDP e dos LSPs, o N2X pode gerar carga de uma interface para a outra encaminhada pelo DUT. Esta carga será designada neste relatório como "carga transaccional" e é constituída por vários fluxos de pacotes IP de tamanho variável. Como estes testes incidem apenas sobre o LDP e este, por sua vez, é um protocolo que corre sobre o IP, faz todo o sentido que durante as comunicações LDP haja tráfego IP a circular. Deste modo, consegue-se uma aproximação mais fidedigna às condições existentes numa rede real.

5.2. Capacidade

5.2.1. Sessões LDP

Neste teste pretende-se determinar a quantidade máxima de sessões LDP que se consegue estabelecer no DUT, bem como o tempo de estabelecimento médio para cada uma dessas sessões. Como o DUT só tem duas interfaces físicas disponíveis, foi usado o N2X para criar múltiplas sessões LDP sobre uma única interface. Esta situação é semelhante a ter o DUT e múltiplos routers ligados a um *switch* Ethernet. Neste caso, o N2X emula todos estes routers, simulando várias interfaces IP e em que cada interface estabelece uma sessão LDP com o DUT. A Figura 5.5 ilustra este cenário de teste. Todas as interfaces estão dentro da mesma rede IP, para assim comunicarem com o DUT e estabelecerem as sessões LDP. Como as sessões LDP se estabelecem apenas entre pares LDP, neste teste apenas importa usar uma interface do DUT para estabelecer sessões LDP. Assim, o cenário usado foi o da Figura 5.1 na situação 1.

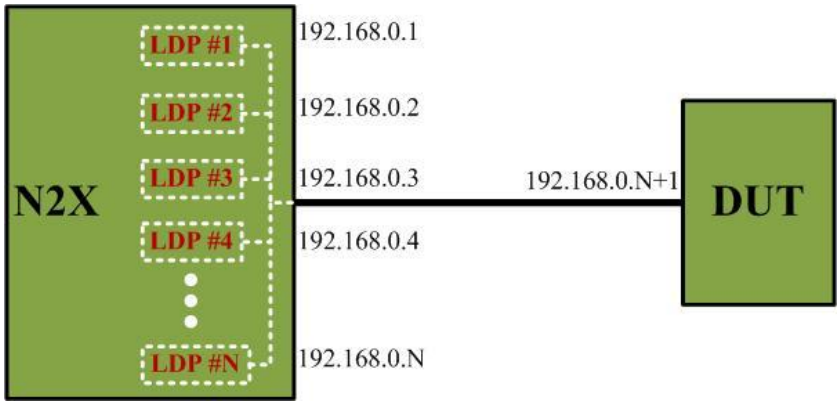


Figura 5.5 - Várias sessões LDP numa só ligação física.

O tempo de estabelecimento de uma sessão LDP é medido desde que se recebe a primeira mensagem de *hello* até a sessão ficar no estado "Operacional", ou seja, até receber a primeira mensagem de *keepalive*. Esta medição foi realizada com recurso a capturas de pacotes e medida a diferença de tempo entre a primeira mensagem de *hello* e a última mensagem (a primeira de *keepalive*). O endereço IP do DUT foi sempre um endereço superior aos endereços do N2X, para que o DUT desempenhasse o papel activo e fosse ele a desencadear o processo de inicialização. O valor medido pode também depender da carga do processador no DUT no período de teste e também do próprio N2X, que pode influenciar a medição. Assim, foram realizadas diversas medições de acordo com a seguinte tabela.

Carga Transaccional (% da carga máxima)	**Tempo médio de estabelecimento por sessão**				
	1	**5**	**10**	**15**	**20**
0%					
22.5%					
45%					
67.5%					
90%					

Tabela 5.1 - Tabela de medições para o tempo de estabelecimento médio de uma sessão LDP.

A tabela anterior relaciona a quantidade de sessões LDP estabelecidas com o tempo médio de estabelecimento para cada sessão e para várias percentagens de carga transaccional. A percentagem de carga transaccional é relativa à carga máxima que a ligação Ethernet pode suportar. Neste caso o DUT suporta no máximo 100Mbps *Full-Duplex*, sendo esta a carga máxima de referência. Não foi usada a carga máxima, pois esta é a situação limite em que pode existir descarte de tráfego.

Foi assumido que a quantidade máxima de sessões LDP que um router MPLS estabelece numa rede real é de 20 sessões. Numa rede Ethernet, os routers IP podem estar interligados por *switches* Ethernet, o que implica que uma interface de um router poderá comunicar com várias interfaces de outros routers. No entanto, para o *backhaul* de redes móveis a densidade de interligações não deverá ser tão elevada.

A Tabela 5.1 apresenta medições para 1, 5, 10, 15 e 20 sessões LDP para percentagens de carga transaccional de 0%, 22.5%, 45%, 67.5%, e 90%. Para obter resultados mais fiáveis, o tempo de estabelecimento foi calculado com base na média de 20 repetições para cada teste. Os modos de funcionamento configurados afectam apenas a distribuição de etiquetas, não tendo influência no estabelecimento das sessões LDP.

Por fim, o DUT foi substituído pelo Cisco, realizando os mesmos testes e comparando assim os resultados.

5.2.2. LSPs

Neste teste pretende-se determinar a quantidade máxima de LSPs que o DUT consegue estabelecer. Foi usado o cenário da situação 2 da Figura 5.1. Neste cenário o DUT funciona como LSR e tem de estabelecer LSPs trocando mensagens LDP com a interface 1 e 2 do N2X. Desta forma, o DUT estabelece apenas uma sessão LDP para cada interface do N2X.

Após serem criadas as sessões LDP entre as duas interfaces do N2X e o DUT, a quantidade de LSPs estabelecidos foi sendo incrementada, aumentando a quantidade de redes de routers presentes nas matrizes do cenário da Figura 5.1. Este teste foi também realizado recorrendo às percentagens de carga transaccional adoptadas para os testes anteriores (apresentadas na Tabela 5.1).

Os modos de funcionamento usados foram os modos 4 e 5, de acordo com a Tabela 4.2. Estes modos, que correspondem à distribuição *Unsolicited* e controlo independente, são os únicos modos completamente funcionais no DUT, tal como observado nos testes funcionais. Estes modos são também os únicos modos suportados pelo Cisco [29].

Mesmo depois dos LSPs estabelecidos, a carga transaccional continua a existir. O N2X não suporta a comutação automática dos fluxos IP para fluxos MPLS após os LSPs serem estabelecidos. Como não é objectivo deste projecto testar o plano de dados do MPLS, o fluxo IP gerado pelo N2X não foi interrompido para dar lugar ao fluxo MPLS.

5.3. Tempo de estabelecimento do LSP

Este teste tem como objectivo medir o tempo de estabelecimento de um LSP. Este tempo depende obviamente do modo de funcionamento configurado. Se o cenário funcionar no modo *Downstream On Demand*, o tempo que demora a estabelecer um LSP será maior em relação ao modo *Unsolicited*. No modo *Downstream on Demand* um LSR só pode enviar o mapeamento se tiver recebido a mensagem de pedido de etiqueta. Este processo atrasa claramente o estabelecimento do LSP contrariamente ao modo *Unsolicited*, em que os mapeamentos podem ser enviados sem receber pedidos de etiqueta. Analogamente, o tempo de estabelecimento de um LSP não vai ser igual para um cenário que use controlo Independente ou Ordenado. Assim, um cenário que utilize os modos 1 e 8 (*Downstream on Demand* e controlo Ordenado) é o cenário em que o tempo de estabelecimento é mais lento. Contrariamente, os modos 3 e 4 (*Unsolicited* e controlo Independente) são os modos em que os LSPs são estabelecidos mais rapidamente.

Como observado nos testes funcionais, o DUT apenas tem o modo *Unsolicited* independente completamente funcional. Como o Cisco também só suporta este modo, foi este o escolhido para realizar este teste.

A medição do tempo de estabelecimento dos LSPs foi realizada recorrendo à captura do tráfego transmitido e recebido nas duas interfaces do DUT, como ilustrado na Figura 5.6. Como o tempo de estabelecimento de um LSP depende de vários factores, como o modo de funcionamento e quantidade de routers presentes ao longo do LSP, este teste pretende apenas medir o tempo entre a recepção da mensagem OSPF, que anuncia a nova rede (designada por *link-state update*), e a transmissão do mapeamento de etiqueta por parte do DUT. Este mapeamento contém a etiqueta anunciada pelo DUT associada à nova rota. No modo *Unsolicited* independente, assim que uma nova rota é estabelecida, o router atribui e transmite uma etiqueta de imediato, sem esperar pelos mapeamentos etiquetas dos routers de *downstream*, como no controlo ordenado; ou sem esperar pelos pedidos de etiquetas dos routers de *upstream*, como na distribuição *On Demand*. Por este motivo é possível realizar o teste da Figura 5.6.

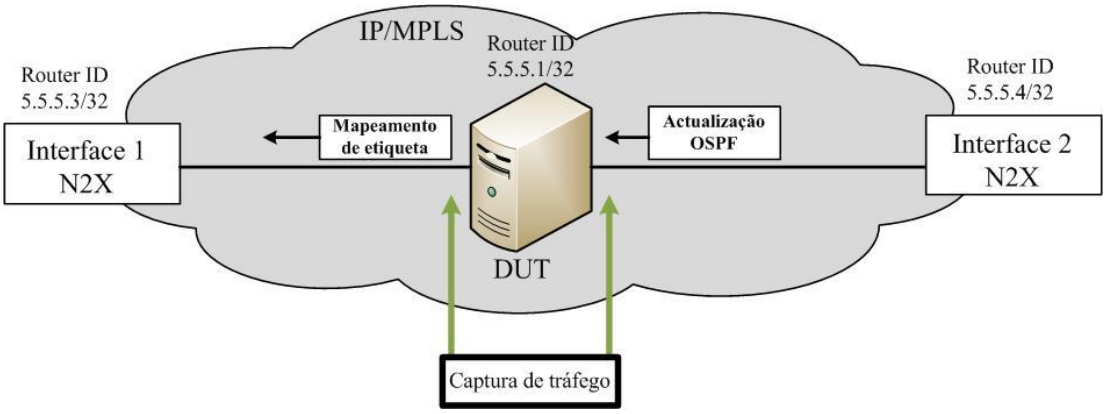


Figura 5.6 - Procedimento para obter o tempo de estabelecimento do LSP.

Para vários LSPs e consequentemente para várias redes anunciadas, o tempo de estabelecimento considerado está entre a primeira mensagem de *link-state update* recebida numa interface e a última mensagem de mapeamento de etiqueta transmitida pelo DUT na outra interface.

Para testar o desempenho da solução para diferentes quantidades de LSPs estabelecidos, foram definidos 6 níveis. Cada nível representa uma percentagem da quantidade máxima de LSPs que se conseguem estabelecer, de acordo com o teste da secção anterior. O tempo de estabelecimento para um só LSP foi igualmente considerado. A Tabela 5.1 é a tabela usada como referência para este teste. Nela são comparados os tempos de estabelecimento para diferentes quantidades de LSPs estabelecidos. Por último, são medidos igualmente os tempos para o Cisco, apenas para o modo de funcionamento suportado. Como os LSPs são unidireccionais, a Tabela 5.2 considera os dois sentidos existentes neste teste: o sentido interface 1 para interface 2 do N2X e vice-versa.

Quantidade de LSPs estabelecidos	**Tempo médio de estabelecimento do LSP**			
	DUT		**Cisco**	
	Sentido 1 -> 2	Sentido 2 -> 1	Sentido 1 -> 2	Sentido 2 -> 1
1 LSP				
20%				
40%				
60%				
80%				
100%				

Tabela 5.2 - Tabela de medições para o tempo de estabelecimento médio de um LSP.

Para cada medição é também verificada a ocupação do processador por parte do DUT. Esta pode ser uma importante medida que pode justificar alguns dos valores do tempo medido. Seguindo o

mesmo raciocínio dos testes anteriores, cada medição foi sujeita a 20 tentativas, permitindo desta forma resultados mais fiáveis.

Por fim, todos os testes anteriores foram igualmente realizados para cada uma das percentagens de carga transaccional usadas anteriormente. Desta forma consegue-se obter uma relação entre o tempo de estabelecimento do LSP e a quantidade de tráfego encaminhado.

5.4. Recuperação de falhas

Nesta secção pretende-se testar o sistema quando sujeito a falhas nas ligações. O objectivo é avaliar a capacidade do DUT em estabelecer um novo LSP, com o mesmo destino, quando o LSP inicial apresenta problemas. Existindo apenas um equipamento disponível (o Cisco), este teste torna-se impossível de realizar pois não se consegue estabelecer dois caminhos para o mesmo destino, como ilustrado na Figura 5.7. Esta figura ilustra um teste hipotético que se poderia realizar para as recuperações em caso de falhas.

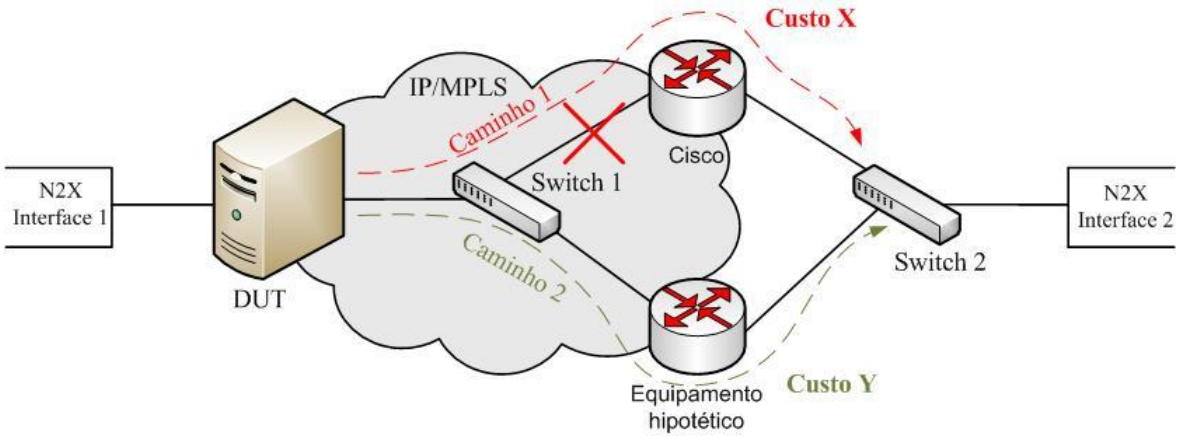


Figura 5.7 – Cenário hipotético para o teste de recuperação de falhas.

Usando OSPF no cenário da figura anterior, é possível definir diferentes custos (em inglês, *costs* ou também conhecido por métricas) diferentes nas interfaces do Cisco e do equipamento hipotético. De acordo com a metodologia do OSPF, a rota que apresentar menos custo será a rota escolhida.

Os custos são tidos em conta nas interfaces de saída dos routers. Atendendo ao sentido dos caminhos da figura, os pacotes viajam da interface 1 para a interface 2 e as redes são anunciadas no sentido inverso. Desta forma, os custos podem ser definidos nas interfaces, do Cisco e do equipamento hipotético, que comunicam directamente com o N2X. Assim, podemos definir, por configuração de custos, qual a rota que os LSPs irão usar entre a interface 1 e 2 do N2X. Por exemplo, se o custo X, na interface do Cisco, for maior que o custo Y do equipamento hipotético, a

rota escolhida será a que passa pelo equipamento hipotético. Desta forma o LSP será estabelecido pelo equipamento hipotético. Se o custo X for menor que o Y, então o LSP será estabelecido pelo Cisco.

Como este teste se torna impossível de realizar, foi pensado um cenário idêntico baseado nos custos atribuídos às interfaces, possível graças à versatilidade do N2X. O cenário usado foi o da Figura 5.8 em que a interface 2 contém duas sessões LDP e OSPF para dois endereços IP diferentes. Desta forma, o DUT vê estas duas interfaces como sendo routers distintos, apesar da interface de saída ser a mesma.

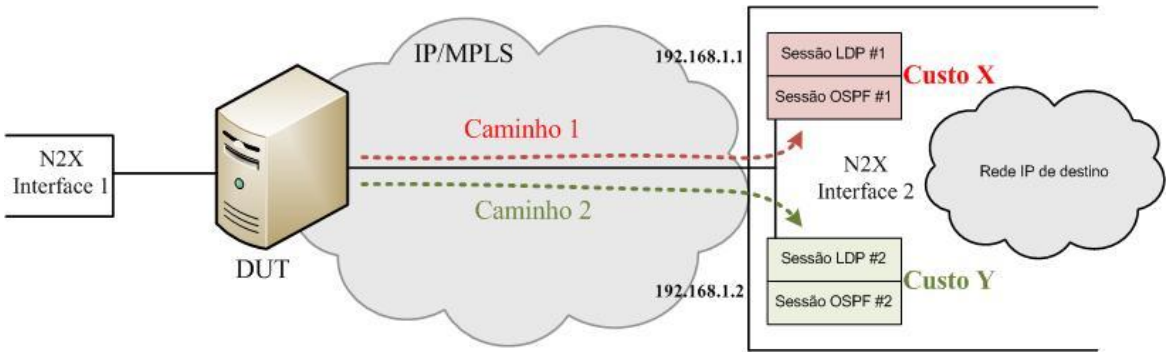


Figura 5.8 - Cenário para teste à convergência de LSPs.

Na figura, as duas sessões OSPF anunciam a mesma rede de destino mas com custos diferentes, permitindo ao DUT escolher apenas um dos caminhos para encaminhamento de tráfego. Modificando os custos das rotas anunciadas, é possível comutar entre os dois caminhos, uma vez que o N2X envia os *link-state update* permitindo ao DUT escolher o novo próximo salto. O objectivo deste teste é avaliar a recuperação do LSP, ou seja, medir o tempo que vai entre a recepção da mensagem de *link-state update* pelo DUT e a transmissão no pedido de etiqueta.

Com este teste é possível observar as diferenças entre os modos de retenção liberal e conservativo. As diferenças na distribuição de etiquetas, pelos dois modos, são notórias após o estabelecimento do caminho pelo OSPF. Usando também o modo *Unsolicited* independente, os procedimentos para os dois modos de retenção devem ser os seguintes:

- Modo liberal:
    - A interface 2 anuncia duas etiquetas, uma por cada sessão LDP;
    - Ao mesmo tempo o DUT anuncia uma etiqueta para cada vizinho que não seja o próximo salto para o FEC em questão. Neste caso, o DUT anuncia uma etiqueta para a interface 1 do N2X e para uma das sessões da interface 2 (que não pertença ao endereço IP do próximo salto);

  - As duas etiquetas anunciadas pela interface 2 do N2X são mantidas pelo DUT, mas apenas utiliza uma, a que estiver associada ao próximo salto usado para encaminhamento.
  - Após a comutação de caminhos, o DUT passa a utilizar a segunda etiqueta anunciada pela interface 2 do N2X, não havendo neste caso troca de mensagens LDP.

- Modo conservativo:
  - A interface 2 anuncia duas etiquetas, uma por cada sessão LDP;
  - Ao mesmo tempo o DUT anuncia uma etiqueta para cada vizinho que não seja o próximo salto para o FEC em questão. Neste caso, o DUT anuncia uma etiqueta para a interface 1 do N2X e para uma das sessões da interface 2 (que não pertença ao endereço IP do próximo salto);
  - No DUT apenas é mantida a etiqueta anunciada pela interface 2 do N2X. O DUT envia uma mensagem de libertar etiqueta para o N2X para libertar a segunda etiqueta anunciada e não usada para encaminhamento.
  - Após a comutação de caminhos, o DUT deve enviar uma mensagem LDP de pedido de etiqueta para que a interface 2 do N2X envie a nova etiqueta a usar.

Desta forma, este teste é importante apenas para o modo de retenção conservativo, pois é ele que permite a troca de mensagens LDP para a convergência de LSPs. Estes procedimentos seriam exactamente os mesmos para o cenário da Figura 5.7, sendo por isso uma forma fiável de testar o DUT sem a necessidade de um terceiro equipamento. No modo liberal, todos os caminhos possíveis são estabelecidos ao inicio, não havendo necessidade de trocas de mensagens LDP aquando da convergência de LSPs.

Para obter resultados mais fiáveis e, à semelhança do que foi feito para o tempo de estabelecimento do LSP, foram realizadas diversas medições para diferentes quantidades de LSPs. Este teste assemelha-se em tudo ao teste anterior, no entanto, este avalia o comportamento do LDP às mudanças que podem ocorrer nas tabelas de encaminhamento.

A tabela de medições usada foi a Tabela 5.2 que, neste teste, contém a diferença de tempo entre a primeira mensagem de *link-state update* recebida pelo DUT (devido ao novo caminho) e a última mensagem de pedido de etiqueta transmitida. Estas medições foram realizadas com base na captura de pacotes na interface do DUT que comunica directamente com a interface 2 do N2X.

5.5. Apresentação e discussão de resultados

A solução LDP testada apresentou, na generalidade, um bom desempenho para os testes efectuados. Comparando os resultados obtidos com os do Cisco, pode-se afirmar que o DUT apresenta um bom desempenho que, no entanto, é influenciado pela maior capacidade de processamento que o DUT tem em relação ao Cisco. Os equipamentos da PT Inovação têm uma capacidade de processamento muito semelhante à do Cisco, por isso pode-se esperar que o LDP do "*MPLS for Linux*", quando em funcionamento num equipamento da PT Inovação, apresente as mesmas limitações reveladas no Cisco.

Apesar do bom desempenho, estes testes revelaram mais algumas lacunas nas funcionalidades da solução LDP. De seguida serão apresentados os resultados por teste efectuado.

Sessões LDP:

Estes testes revelaram, logo à partida, que o DUT suporta até 5 sessões LDP por interface. Na tentativa de se estabelecer uma sexta sessão LDP, o *daemon ldpd* morre encerrando todas as outras sessões. Desta forma, os testes previamente planeados tiveram de ser ligeiramente modificados. Em vez dos testes às sessões LDP numa só interface, baseado na situação 1 da Figura 5.1, utilizou-se a situação 2 testando as duas interfaces do DUT simultaneamente. Neste caso pode-se estabelecer até 10 sessões LDP, 5 por interface. As médias do tempo de estabelecimento das sessões LDP, para as duas interfaces do DUT, estão expressas na Tabela 5.1.

Carga transaccional (Mbps)	**Tempo de estabelecimento das sessões LDP no DUT (minutos:segundos,milisegundos)**					
	1		3		5	
0	00:03,517	00:03,528	00:03,578	00:03,577	00:03,600	00:03,589
22,5	00:03,520	00:03,524	00:03,502	00:03,483	00:03,614	00:03,560
45	00:03,543	00:03,528	00:03,531	00:03,535	00:03,625	00:03,616
67,5	00:03,529	00:03,547	00:03,563	00:03,547	00:03,563	00:03,547
90	00:03,542	00:04,954	00:03,578	00:05,090	00:03,747	00:05,675

Tabela 5.3 - Resultados para o tempo de estabelecimento das sessões LDP no DUT.

A tabela mostra que os tempos de estabelecimento de uma, três ou cinco sessões LDP são muito semelhantes. Isto mostra que a implementação LDP processa independentemente cada sessão, não

as estabelecendo sequencialmente. Nas capturas realizadas é possível observar que o DUT envia as mensagens LDP de inicialização quase simultaneamente para as várias sessões, não esperando que uma sessão seja estabelecida para iniciar o estabelecimento da seguinte. Os valores médios para o estabelecimento das sessões LDP no DUT rondam o intervalo de 3 segundos e 400 milissegundos e os 3 segundos e 600 milissegundos. Os valores para o Cisco estão expressos na Tabela 5.2 que mostra resultados semelhantes aos do DUT.

Carga transaccional (Mbps)	**Tempo de estabelecimento das sessões LDP no Cisco (segundos,milisegundos)**					
	1		3		5	
0	3,616	3,615	3,607	3,616	3,654	3,610
22,5	3,616	3,623	3,664	3,640	3,649	3,584
45	3,614	3,615	3,626	3,610	3,704	3,598
67,5	3,623	3,618	3,649	3,611	3,705	3,601
90	3,590	3,618	3,583	3,597	3,674	3,593

Tabela 5.4 - Resultados para o tempo de estabelecimento das sessões LDP no Cisco.

A única diferença assinalável entre as duas tabelas anteriores está no teste efectuado com uma carga transaccional de 90Mbps. Com esta carga o DUT apresenta perdas no encaminhamento, revelando limitações ao nível de *hardware*. Estas perdas apenas aconteciam para uma carga transaccional superior a 70Mbps nos dois sentidos e apenas na interface incorporada na placa de rede do DUT. Este facto revela uma limitação na comunicação entre o processador e a placa de rede. Devido às perdas observadas, com uma carga de 90Mbps, o tempo de estabelecimento das sessões, numa das interfaces do DUT, aumenta. Isto pode ser observado mais facilmente no gráfico da Figura 5.9, onde é também apresentado o desvio padrão para as 20 medições realizadas em cada teste.

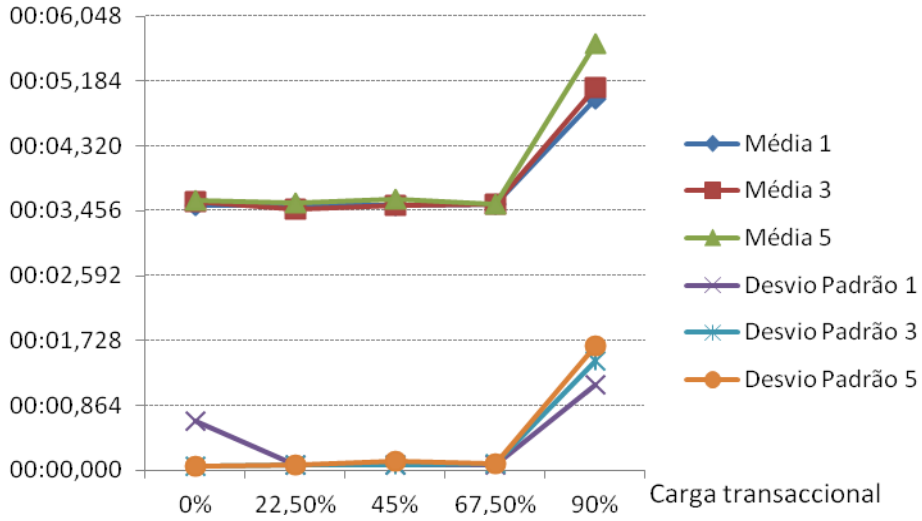


Figura 5.9 - Média e desvio padrão para o tempo de estabelecimento da sessão LDP no DUT.

O gráfico anterior mostra a média para os tempos de estabelecimento de uma, duas e três sessões LDP e respectivos desvios padrões para uma interface. No gráfico pode-se observar o pico que acontece para 90Mbps devido à situação referida anteriormente. Este pico acontece igualmente para os valores de desvio padrão. Nesta situação existe também uma maior dispersão nos valores do tempo de estabelecimento. No gráfico é também observável que a média e o desvio padrão são mais afectados no estabelecimento de 5 sessões LDP para 90Mbps. Como cinco sessões requerem obviamente mais mensagens trocadas, mais mensagens são descartadas e mais retransmissões irão existir, aumentando o tempo de estabelecimento das sessões. Estas retransmissões apenas existem nas mensagens de inicialização e *keepalive*, uma vez que são enviadas sobre ligações TCP. Os *hellos*, enviados sobre UDP, se se perderem não são retransmitidos.

O gráfico da Figura 5.10 mostra a mesma informação que o gráfico da Figura 5.9, mas para o Cisco. Para este equipamento os valores são idênticos aos do DUT, tanto a média como o desvio padrão, não existindo o pico observado no DUT para 90Mbps. No Cisco, os tempos de estabelecimentos das sessões LDP também não são muito afectados pela quantidade de sessões estabelecidas, tal como acontece no DUT. De referir que no Cisco consegue-se estabelecer mais de 5 sessões LDP por interface.

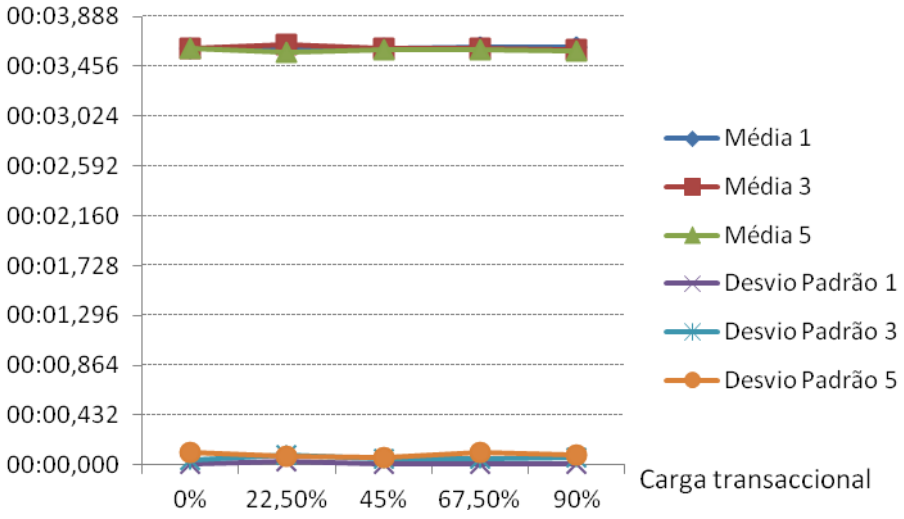


Figura 5.10 - Média e desvio padrão para o tempo de estabelecimento da sessão LDP no Cisco.

LSPs:

Estes testes permitiram determinar a quantidade máxima de LSPs que o DUT consegue estabelecer. Incrementando sucessivamente o número de LSPs estabelecidos, chegou-se à conclusão que aproximadamente 500 LSPs, por cada sentido, poderão ser estabelecidos sem problemas. Acima deste valor, o DUT apresenta diversos problemas como, o encerramento das sessões LDP ou mesmo a morte do *daemon ldpd*. Foi observado que após estabelecer e encerrar grandes quantidades de LSPs, o *daemon ldpd* morre apresentado a mensagem de "*double-linked list corrupted*", revelando deficiências na implementação do software.

Ainda assim, definindo um tecto máximo de 500 LSPs por sentido e assumindo que este valor é mais do que aceitável para um router IP/MPLS, o DUT mostrou ser capaz de os estabelecer rapidamente sem problemas. Esta quantidade de LSPs é usada como tecto máximo no teste seguinte.

Tempo de estabelecimento do LSP:

Neste teste foram medidos os tempos de estabelecimento do LSP, ou sejam, o tempo entre a recepção da mensagem de *link-state update* do OSPF e a transmissão do mapeamento de etiqueta. Os resultados obtidos, para o DUT, são apresentados na Tabela 5.5. A tabela contém os tempos de estabelecimento médios (de 20 amostras) da quantidade de LSPs indicadas, para os dois sentidos e para diferentes cargas transaccionais.

Quantidade de LSPs estabelecidos	Tempo médio de estabelecimento do LSP (segundos,milissegundos)									
	0Mbps		22,5Mbps		45Mbps		67,5Mbps		90Mbps	
1 LSP	0,234	0,386	0,226	0,482	0,227	0,563	0,228	0,426	--	--
100 LSPs	0,328	0,675	0,357	0,713	0,426	0,756	0,426	0,778	--	--
200 LSPs	0,403	0,757	0,455	0,935	0,552	0,910	0,629	1,009	--	--
300 LSPs	0,888	0,917	1,204	1,191	1,257	1,291	1,315	1,572	--	--
400 LSPs	1,331	1,258	1,274	1,543	1,322	1,681	1,393	2,058	--	--
500 LSPs	1,204	1,509	1,568	2,145	1,318	2,412	1,124	2,759	--	--

Tabela 5.5 - Resultados para o tempo de estabelecimento do LSP no DUT.

A tabela anterior mostra que quanto maior a quantidade de LSPs estabelecidos maior o tempo necessário para os estabelecer. Apesar de este aumento ser óbvio, devido à maior quantidade de mensagens trocadas, os valores indicam que o tempo de estabelecimento por LSP diminui. Isto deve-se à não sequencialidade de estabelecimento dos LSP, à semelhança do que acontece para as sessões LDP. Também se pode observar que os tempos para o segundo sentido dos LSPs são sempre superiores ao primeiro sentido. A razão para isto acontecer está na forma com foi realizado o teste. Como o N2X não permite o anúncio simultâneo das redes (que posteriormente dão lugar aos LSPs), através das sessões OSPF, tem de se seleccionar sequencialmente estes anúncios. O primeiro anúncio faz com que o processamento no DUT aumente, afectando o processamento do segundo anúncio para o outro sentido. É por esta razão que os tempos para o segundo sentido são sempre superiores aos do primeiro sentido.

Associadas às médias do tempo de estabelecimento dos LSPs estão os desvios padrão das 20 medidas realizadas. Estes resultados estão expressos na Tabela 5.6.

Quantidade de LSPs estabelecidos	Desvio padrão das medidas do estabelecimento do LSP (segundos,milissegundos)									
	0Mbps		22,5Mbps		45Mbps		67,5Mbps		90Mbps	
1 LSP	0,029	0,122	0,003	0,095	0,003	0,108	0,003	0,096	--	--

100 LSPs	0,031	0,099	0,035	0,105	0,192	0,109	0,027	0,108	--	--
200 LSPs	0,023	0,094	0,031	0,120	0,032	0,089	0,032	0,190	--	--
300 LSPs	0,414	0,359	0,208	0,083	0,176	0,184	0,134	0,186	--	--
400 LSPs	0,292	0,258	0,411	0,213	0,416	0,501	0,551	0,596	--	--
500 LSPs	0,574	0,350	0,423	0,451	0,644	0,942	0,757	1,144	--	--

Tabela 5.6 - Desvio padrão das medidas do estabelecimento do LSP no DUT.

Nas duas tabelas anteriores, as medidas para a carga transaccional de 90Mbps não foram realizadas, devido à limitação da placa de rede usada no DUT, já referida anteriormente. Neste caso, os tempos de estabelecimento dos LSPs aumentavam muito, não permitindo obter valores que reflectissem a capacidade da solução LDP (devido ao descarte de mensagens).

A Tabela 5.6 mostra que para grandes quantidades de LSPs o desvio padrão aumenta ligeiramente, mostrando uma maior dispersão das medidas realizadas.

O gráfico da Figura 5.11 ilustra aquilo que já foi referido anteriormente, ou seja, o aumento do tempo de estabelecimento dos LSPs e dos desvios padrão com o aumento da quantidade de LSPs. Para as maiores quantidades de LSPs, também se pode observar que o tempo de estabelecimento é maior quanto maior for a carga transaccional. Ou seja, para elevadas quantidades de LSPs, a carga transaccional afecta de forma mais significativo o tempo de estabelecimentos dos LSPs.

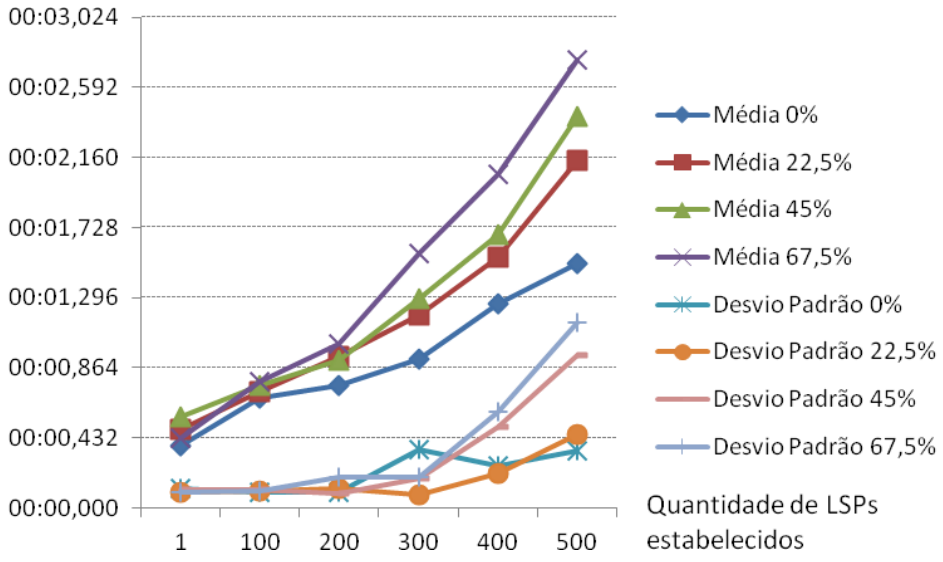


Figura 5.11 - Média e desvio padrão para o tempo de estabelecimento dos LSPs no DUT.

Durante os testes foram também anotadas as percentagens de ocupação do processador por parte dos três *daemons* do *quagga*: zebra, ospfd e ldpd. A Tabela 5.7 contém as percentagens de ocupação dos três *daemons* para todos os testes realizados.

LSPs	0Mbps			22,5Mbps			45Mbps			67,5Mbps		
	zebra	ospfd	ldpd	zebra	ospfd	ldpd	zebra	ospfd	ldpd	zebra	ospfd	ldpd
1 LSP	0,7%	0,2%	0,3%	0,7%	0,2%	0,3%	0,7%	0,2%	0,3%	0,7%	0,2%	0,3%
100 LSPs	7%	2%	4%	5%	2%	4%	6%	2%	7%	9%	7%	10%
200 LSPs	7%	1%	6%	14%	1%	18%	6%	2%	7%	23%	8%	26%
300 LSPs	7%	3%	12%	22%	10%	37%	25%	10%	45%	35%	17%	48%
400 LSPs	11%	4%	21%	27%	10%	43%	33%	23%	64%	42%	17%	54%
500 LSPs	26%	9%	73%	29%	10%	72%	33%	17%	65%	33%	17%	65%

Tabela 5.7 - Ocupação do processador durante o estabelecimento de LSPs.

Os valores de ocupação do processador são maiores quanto maior é a quantidade de LSPs a estabelecer. Os *daemons* com mais carga são o zebra e o ldpd. Como se pode observar pela tabela anterior, a ocupação do processador não é afectada pela carga transaccional, mostrando aqui uma clara dissociação do processo encaminhamento de pacotes com os *daemons* do *quagga*. As elevadas percentagens de ocupação, para grandes quantidades de LSPs, mostram que um router MPLS com menos capacidade de processamento terá alguma dificuldade para processar estas elevadas quantidades de LSPs.

O que foi referido no parágrafo anterior foi observado nos testes com o Cisco. Ou seja, para grandes quantidades de LSPs, o Cisco apresenta grandes dificuldades para os estabelecer, encerrando as sessões LDP na maior parte das vezes. O DUT apresenta claramente uma capacidade de processamento superior à do Cisco, levando a concluir que a solução LDP poderá ter o mesmo desempenho do Cisco quando colocada sobre capacidades de processamento semelhantes.

Devido à grande dificuldade em fazer os testes com o Cisco, por causa da inexistência de ferramentas capazes de capturar pacotes nas suas interfaces (nos testes ao DUT foi usada aplicação *tshark* do Linux), os tempos de estabelecimento foram medidos recorrendo aos *logs* do Cisco. Sem carga transaccional e para 100 LSPs foi medido um tempo de 220ms, aproximadamente; para 200 LSPs foi medido um tempo de 1s e 12ms; e para 100 LSPs com uma carga transaccional de 67,5Mbps, o tempo medido foi de 5s e 336ms. Valores estes claramente superiores aos do DUT.

Recuperação de falhas:

Os testes às recuperações de falhas revelaram um mau desempenho do DUT. Funcionalmente, o DUT é incapaz de recuperar falhas, não restabelecendo os LSPs no modo de retenção conservativo. O modo liberal funciona sem problemas uma vez que todos os LSPs alternativos são estabelecidos inicialmente. Após a falha o DUT apenas tem de usar a etiqueta previamente anunciada. No modo conservativo, como apenas são guardadas as etiquetas usadas para encaminhamento, após a falha o DUT deveria transmitir um pedido de etiqueta no sentido *upstream* requisitando uma etiqueta ao seu novo LSR de próximo salto. As tabelas seguintes ilustram esta não conformidade do DUT.

```
dut# show ip route
Codes: K - kernel route, C - connected, S - static, R -
RIP, O - OSPF, I - ISIS, B - BGP, > - selected route, *
- FIB route

O>* 1.1.1.0/24 [110/11] via 172.29.0.1, eth1 (label 22)
C>* 5.5.5.1/32 is directly connected, dummy0
O>* 5.5.5.3/32 [110/10] via 192.168.0.1, eth0
O>* 5.5.5.4/32 [110/10] via 192.168.1.1, eth1
O>* 5.5.5.5/32 [110/10] via 192.168.1.2, eth1
C>* 127.0.0.0/8 is directly connected, lo
O   192.168.1.0/24 [110/10] is directly connected, eth1
C>* 192.168.1.0/24 is directly connected, eth1
O   192.168.0.0/24 [110/10] is directly connected, eth0
C>* 192.168.0.0/24 is directly connected, eth0

dut# show ldp database
  5.5.5.1/32        local binding:  label: gen 10241
  192.168.1.0/24    local binding:  label: gen 10242
  192.168.0.0/24    local binding:  label: gen 10243
  5.5.5.4/32        local binding:  label: gen 10244
  5.5.5.5/32        local binding:  label: gen 10245
  5.5.5.1/32        local binding:  label: gen 10246
  192.168.1.0/24    local binding:  label: gen 10247
  192.168.0.0/24    local binding:  label: gen 10248
  5.5.5.5/32        local binding:  label: gen 10249
  5.5.5.3/32        local binding:  label: gen 10250
  5.5.5.1/32        local binding:  label: gen 10251
  192.168.1.0/24    local binding:  label: gen 10252
  192.168.0.0/24    local binding:  label: gen 10253
  5.5.5.4/32        local binding:  label: gen 10254
  5.5.5.3/32        local binding:  label: gen 10255
  1.1.1.0/24        local binding:  label: gen 10256
  1.1.1.0/24        local binding:  label: gen 10257
  1.1.1.0/24  remote  binding:  label:  gen  22  lsr:
192.168.1.1:0 ingress
```

```
dut# show ip route
Codes: K - kernel route, C - connected, S - static, R -
RIP, O - OSPF,
      I - ISIS, B - BGP, > - selected route, * - FIB route

O>* 1.1.1.0/24 [110/12] via 172.29.0.2, eth1
C>* 5.5.5.1/32 is directly connected, dummy0
O>* 5.5.5.3/32 [110/10] via 192.168.0.1, eth0
O>* 5.5.5.4/32 [110/10] via 172.29.0.1, eth1
O>* 5.5.5.5/32 [110/10] via 172.29.0.2, eth1
C>* 127.0.0.0/8 is directly connected, lo
O   172.29.0.0/24 [110/10] is directly connected, eth1
C>* 172.29.0.0/24 is directly connected, eth1
O   192.168.0.0/24 [110/10] is directly connected, eth0
C>* 192.168.0.0/24 is directly connected, eth0
dut# show ldp database
  5.5.5.1/32        local binding:  label: gen 10241
  192.168.1.0/24    local binding:  label: gen 10242
  192.168.0.0/24    local binding:  label: gen 10243
  5.5.5.4/32        local binding:  label: gen 10244
  5.5.5.5/32        local binding:  label: gen 10245
  5.5.5.1/32        local binding:  label: gen 10246
  192.168.1.0/24    local binding:  label: gen 10247
  192.168.0.0/24    local binding:  label: gen 10248
  5.5.5.5/32        local binding:  label: gen 10249
  5.5.5.3/32        local binding:  label: gen 10250
  5.5.5.1/32        local binding:  label: gen 10251
  192.168.1.0/24    local binding:  label: gen 10252
  192.168.0.0/24    local binding:  label: gen 10253
  5.5.5.4/32        local binding:  label: gen 10254
  5.5.5.3/32        local binding:  label: gen 10255
  1.1.1.0/24        local binding:  label: gen 10259
```

A tabela da esquerda foi retirada do DUT antes da convergência de LSPs. Nela está a tabela de rotas do DUT que contém, além das redes directamente ligadas, a rede 1.1.1.0/24 exterior à rede MPLS. Para esta rede é estabelecido um LSP que tem como etiqueta de saída (no DUT) a etiqueta 22. Esta etiqueta será a etiqueta de entrada da interface 2 do N2X (de acordo com a Figura 5.8). À direita são apresentadas as tabelas de encaminhamento IP e de etiquetas do DUT após a

convergência dos LSPs. Como se pode observar, para a rede 1.1.1.0/24 não existe qualquer etiqueta de saída.

Capítulo 6: Conclusões

O trabalho apresentado nesta dissertação teve como objectivo a avaliação de uma solução IP/MPLS para utilização nos equipamentos da PT Inovação. Esta solução foi avaliada com base nos requisitos definidos pelo *Broadband-Forum* (especificação "*IP/MPLS Forum 20.0.0*") para aplicação no *backhaul*, baseado em IP/MPLS, de redes móveis.

Depois de uma primeira abordagem teórica sobre alguns conceitos MPLS, partiu-se para o levantamento dos requisitos definidos pelo *Broadband-Forum*. Estes requisitos têm em conta os serviços usados actualmente nas redes móveis e os serviços futuros. A estes requisitos foram adicionados outros pretendidos pela PT Inovação. Com base neste conjunto de requisitos foi então realizada uma pesquisa sobre as soluções IP/MPLS existentes no mercado. Dentro das várias soluções disponíveis, foi escolhida aquela que, além de cumprir com alguns dos requisitos, era mais fácil de usar, de rápida disponibilização do software e sobretudo grátis.

Para caracterizar a solução escolhida em termos de funcionalidades, capacidades e desempenho, foram concebidos e realizados diferentes testes em laboratório. Estes testes foram inicialmente planeados e justificados, e divididos em dois grandes grupos: testes funcionais e testes de desempenho. Os testes funcionais tiveram como objectivo garantir a conformidade da solução com as normas. Os testes de desempenho permitiram caracterizar a solução em termos de capacidade e robustez.

O *backhaul* das redes móveis enfrenta actualmente diversas mudanças que lhe permite suportar os serviços emergentes, como o HSPA e LTE, mas também os serviços actualmente já existentes, como o GSM e UMTS. Desta forma, as novas soluções terão de suportar diversas tecnologias simultaneamente, bem como serem dotadas de elevada escalabilidade. O IP/MPLS surge como aposta para muitos dos operadores destas redes, uma vez que permite o transporte do IP, usado nas tecnologias emergentes, e também o transporte das tecnologias TDM /ATM, usadas no GSM e UMTS, através do estabelecimento de PWs.

Os requisitos definidos pela especificação "*IP/MPLS Forum 20.0.0"* assentam principalmente sobre o estabelecimento de PWs e VPNs MPLS. Estes requisitos são indispensáveis à implementação do IP/MPLS nas redes móveis. Além destes requisitos, é necessário que as falhas

sejam detectadas através de mecanismos OAM e recuperadas através do FRR. A sinalização LDP ou RSVP-TE surge como opcional, mas muito importante uma vez que facilita a gestão e operação destas redes.

A pesquisa efectuada permitiu encontrar três soluções: duas soluções que cumprem todos os requisitos definidos mas com custos e uma de software grátis, disponível na Internet, mas que não suporta alguns dos requisitos. As duas soluções pagas estão disponíveis através das empresas Metaswitch [21] e IP Infusion [22]. A solução grátis, designada por "MPLS for Linux", suporta encaminhamento MPLS, embutido no *kernel* do Linux, encaminhamento IP dinâmico e sinalização LDP, através da aplicação *quagga*. Esta última foi a solução escolhida.

A PT Inovação tem actualmente mecanismos que lhe permite implementar e suportar o plano de dados do encaminhamento MPLS. Por este motivo os testes efectuados incidiram apenas sobre o protocolo LDP suportado no "MPLS for Linux".

Os testes funcionais efectuados revelaram algumas lacunas na solução LDP. As mensagens LDP com mais problemas são as mensagens de pedido de etiqueta e de notificação. As mensagens de pedido de etiqueta não cumprem com algumas das especificações, inviabilizando o uso do modo de distribuição *Downstream On Demand*. As mensagens de notificação apresentam uma implementação deficiente. De acordo com as especificações, em algumas das situações negativas testadas nesta dissertação, a solução LDP não respondeu com a mensagem de notificação correcta.

A solução LDP escolhida apenas consegue funcionar no modo em que realiza *merge* de etiquetas. Esta limitação foi considerada pouco grave, uma vez que para o transporte IP, o único usado nos testes, é aconselhado sempre o uso de *merge* de etiquetas. A descoberta básica de pares LDP encontra-se a funcionar sem problemas.

Os testes aos modos de funcionamento revelaram problemas no modo *Downstream on Demand* (devido aos problemas nas mensagens de pedido de etiqueta) e no modo *Unsolicited* ordenado quando o LDP realiza funções de LER de saída. Neste caso, o LER de saída deveria ser o primeiro a enviar as suas associações FEC-etiqueta no sentido *downstream*, o que não está acontecer. O modo *Unsolicited* independente encontra-se completamente funcional. Os testes revelaram também que a solução LDP funciona com os diversos tipos de protocolos de encaminhamento IP. Foram testados o RIP, OSPF e BGP sem qualquer problema. A detecção de *loops* funciona apenas através de *Hop Count* e o *Label Space* por plataforma é o único suportado.

Os testes de desempenho revelaram um bom desempenho da solução. O estabelecimento de sessões LDP e LSPs são realizados em quantidades suficientes para uma rede móvel. A solução LDP suporta até 5 sessões LDP por interface e 500 LSPs por sentido. Os tempos de estabelecimento

revelaram dependência com a capacidade de processamento utilizada. Em comparação com o Cisco estes tempos são muito semelhantes, sendo por vezes até melhores. A recuperação de falhas não se encontra a funcionar para o modo de retenção conservativo.

Através dos resultados obtidos pode-se concluir que a solução LDP testada apresenta problemas que põem em causa a sua utilização em cenários reais. O único modo de configuração completamente funcional é o modo distribuição *Unsolicited* com controlo independente e retenção liberal. A solução funciona também realizando *merge* de etiquetas e com *Label Space* por plataforma configurado. A detecção de *loops* que deve ser usada é a detecção por *Hop Count*. Apesar de estes modos funcionarem, esta solução LDP apresenta alguma instabilidade. Por diversas vezes, a aplicação morre, encerrando os LSPs e sessões LDP, originando assim possíveis perdas de tráfego. Além disto, a solução escolhida não suporta estabelecimento de PWs e VPNs, imprescindíveis para as redes móveis.

A versão de software utilizada data de Junho de 2009. Entretanto, algumas correcções já foram disponibilizadas e alguns dos problemas referidos anteriormente poderão já estar corrigidos.

Como trabalho futuro é sugerida uma comparação dos problemas encontrados nesta dissertação com as correcções efectuadas no software desde Junho 2009. Planos de teste para o MPLS-TE, nomeadamente para o protocolo RSVP-TE, e para as VPNs sobre MPLS são também trabalhos que se poderão realizar para completar a validação de soluções que cumpram com mais requisitos.

Os objectivos desta dissertação foram alcançados, na medida em que se conseguiu caracterizar completamente uma solução IP/MPLS. Através destes resultados a PT Inovação pode decidir sobre a sua incorporação nos seus equipamentos, ou então decidir sobre uma das opções pagas. O trabalho de levantamento de requisitos foi também importante, permitindo à PT Inovação ter um conhecimento mais aprofundado sobre a implementação do IP/MPLS em redes móveis.

Como conclusão final, pode-se afirmar que durante a realização desta dissertação foram adquiridos sólidos conhecimentos ao nível do IP/MPLS e do seu funcionamento. Paralelamente, desenvolveu-se a capacidade de espírito crítico, muito importante nos testes efectuados, e foram adquiridos conhecimentos sobre a estrutura interna do *kernel* Linux usado para instalar o software testado.

Anexo I: Formato das mensagens LDP

Este anexo descreve sucintamente o formato das mensagens LDP. Esta descrição resulta da necessidade adquirida durante os testes funcionais, em que constantemente era necessário aceder às especificações para análise das mensagens.

Todas as mensagens LDP são enviadas em unidades de dados - em inglês, *Protocol Data Units* (PDUs), sobre ligações TCP; apenas as mensagens de *Hello* são enviados sobre ligações UDP. Cada PDU pode conter mais que uma mensagem LDP. Desta forma, cada PDU contém um cabeçalho seguido de uma ou várias mensagens LDP. A figura seguinte apresenta o formato de um cabeçalho LDP.

Figura I.1 - Cabeçalho de uma mensagem LDP [5].

Na Figura I.1 estão identificados os seguintes campos:

- Versão: identificação da versão do protocolo. Actualmente, a versão é a 1.
- Tamanho PDU: tamanho total do PDU em bytes, excluindo o tamanho do campo Versão e Tamanho PDU. O tamanho máximo do PDU é negociado na inicialização da sessão LDP. Até esta ser completada o tamanho máximo é de 4096 bytes.
- Identificador LDP: identifica o *Label Space* usado pelo transmissor da mensagem. Os primeiros 4 bytes identificam o LSR e deverá ser um valor único global. Os últimos 2 bytes identificam o *Label Space* dentro do LSR. Para um *Label Space* por plataforma, estes últimos 2 bytes devem ser preenchidos com zero.

A seguir ao cabeçalho LDP vêm uma ou várias mensagens, todas elas com o formato descrito na Figura I.2.

Figura I.2 - Formato de uma mensagem LDP [5].

Os campos constituintes de uma mensagem LDP são os seguintes:

- U-bit: bit que determina se na recepção de uma mensagem desconhecida, uma notificação deve ser enviada de volta ao criador da mensagem (valor 0); ou, por outro lado, se a mensagem desconhecida deve ser silenciosamente ignorada (valor 1).
- Tipo: tipo da mensagem. Existem os seguintes tipos de mensagens:

Tipo	Mensagem
0x001	Notificação
0x100	*Hello*
0x200	Inicialização
0x201	*Keepalive*
0x300	Endereço
0x301	Remoção de endereço
0x400	Mapeamento de etiqueta
0x401	Pedido de etiqueta
0x404	Abortar pedido de etiqueta
0x402	Remover etiqueta
0x403	Libertar etiqueta

Tabela I.1 - Mensagens LDP e respectivos identificadores [5].

- Tamanho: comprimento da mensagem em bytes, incluindo o ID da mensagem e parâmetros.
- ID Mensagem: campo usado para identificar esta mensagem. Usado pelo LSR transmissor para facilitar a identificação das mensagens de notificação que possam estar relacionadas com esta mensagem.
- Parâmetros: campo que contem parâmetros obrigatórios ou opcionais das mensagens.

O campo "parâmetros" de uma mensagem LDP tem um formato comum que usa um esquema de codificação *Type-Length-Value* (TLV). A figura seguinte mostra a codificação TLV de uma mensagem LDP.

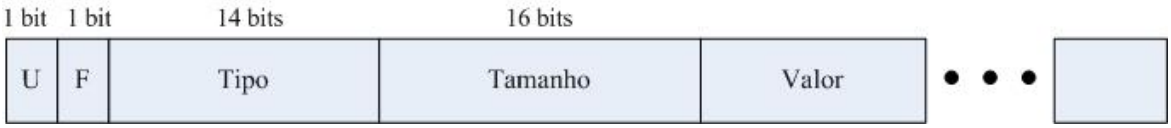


Figura I.3 - Codificação TLV de uma mensagem LDP [5].

O formato TLV é constituído pelos seguintes campos:

- U-bit (*Unknown TLV bit*): bit que determina se uma mensagem desconhecida deve ser processada ou enviada de volta para o seu criador.
- F-bit (*Forward Unknown TLV bit*): este bit apenas tem importância quando o bit U é igual a 1 (mensagem vai ser processada), e determina se uma mensagem desconhecida deverá ser encaminhada.
- Tipo: codifica a forma como o campo Valor vai ser interpretado.
- Tamanho: tamanho do campo Valor em bytes.
- Valor: *array* de bytes que codifica a informação a ser interpretada.

Os TLVs definidos em [5] estão listados na Tabela I.2.

Tipo	**TLV**	**Tipo**	**TLV**
0x0100	FEC	0x0400	*Common Hello Parameters*
0x0101	*Address List*	0x0401	*IPv4 Transport Address*
0x0103	*Hop Count*	0x0402	*Configuration Sequence Number*
0x0104	*Path Vector*	0x0403	*IPv6 Transport Address*
0x0200	*Generic Label*	0x0500	*Common Session Parameters*
0x0201	*ATM Label*	0x0501	*ATM Session Parameters*
0x0202	*Frame Relay Label*	0x0502	*Frame Relay Session Parameters*
0x0300	*Status*	0x0600	*Label Request Message ID*
0x0301	*Extended Status*	0x3E00-0x3EFF	*Vendor-Private*
0x0302	*Returned PDU*	0x3F00-0x3FFF	*Experimental*
0x0303	*Returned Message*		

Tabela I.2 - Tipos de TLV suportados por [5].

Anexo II: Instalação do "MPLS for Linux"

Neste documento pretende-se explicar de forma sucinta, mas clara, todos os passos necessários à instalação do software MPLS bem como do protocolo LDP de código aberto para Linux.

Software específico para MPLS e LDP em Linux:

- **kernel-2.6.27.21-170.2.56.fc10mpls.1.193.src.rpm**
- **mpls.h**
- **iproute-2.6.27-2.fc10mpls.1.963.src.rpm**
- **iptables-1.4.1.1-2.fc10mpls.1.963.src.rpm**
- **ebtables-2.0.8-5.fc10.mpls.1.963.src.rpm**
- **quagga-0.99.6.tar**
- **ldp-portable_noipv6.tar**

Kernel com suporte para MPLS:

A instalação do *kernel* requer que os seguintes pacotes de software sejam previamente instalados:

- alien → **`#apt-get install alien`**
- ncurses-dev → **`#apt-get install ncurses-dev`**

O pacote para instalação está no formato **.rmp** (*Red Hat*) no caso do Ubuntu será necessário converter para o formato **.deb** (*Debian*), para tal usa-se o conversor de formatos **`alien`**.

```
#alien -d kernel-2.6.27.21-170.2.56.fc10.mpls.1.963.src.rpm
```

O ficheiro **`kernel_2.6.27.24-171.2_i386.deb`** será então gerado.

Após a conversão de formato, descompacta-se o ficheiro com a extensão **.deb** para uma pasta especificada (opção –x do comando **`dpkg`**),

```
#dpkg –x kernel_2.6.27.24-171.2_i386.deb <pasta_destino>
```

Compilar e instalar *kernel*:

1. Na pasta onde se descompacta os ficheiros do *kernel* existe um ficheiro denominado **`linux-2.6.27.tar.bz`** que necessita também ser descompactado:

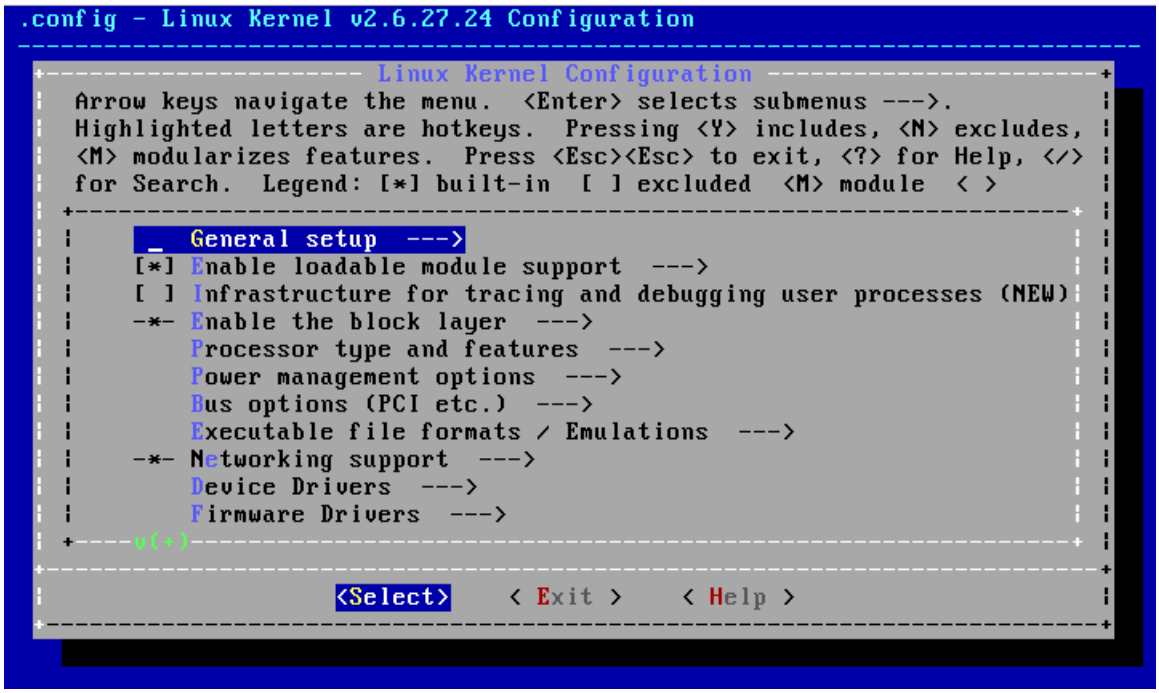
#tar -jxvf linux-2.6.27.tar.bz

Depois de descompactado o ficheiro, será criada uma pasta com o mesmo nome dentro da qual estarão todos os ficheiros do *kernel*. Mudar de directório para a referida pasta, **linux-2.6.27**.

2. Antes da compilação será necessário configurar o *kernel*, através do menu de configuração:

#make menuconfig

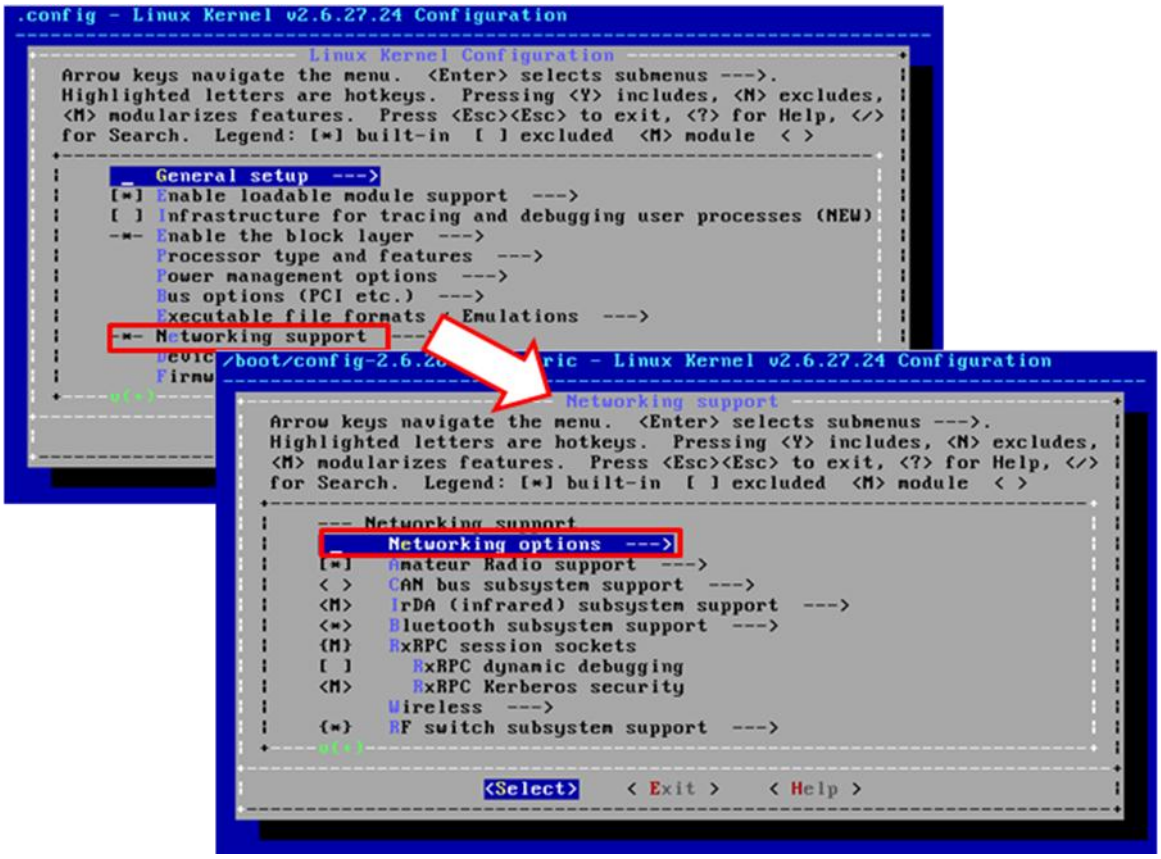
Aparecerá então o menu **Linux Kernel Configuration**, abaixo apresentado.



3. Em primeiro é necessário carregar as configurações do *kernel* actual, para servirem como ponto de partida para as configurações necessárias. Para tal navegue até ao final do menu onde aparecerá a opção **Load na Alternate Configuration File** (salientado a vermelho) que deverá ser seleccionada.

   No menu que aparecerá de seguida indique o caminho para o ficheiro de configuração do *kernel* actual. Neste caso, o caminho é **/boot/config-2.6.28-11-generic**.

4. De volta ao menu principal deverá ser seleccionado o menu **Networking Support** e neste **Networking Options**.



5. Dentro do menu **Networking Options** várias opções deverão ser seleccionadas (tecla y):
    - **Multiprotocol Label Switching**
        - **MPLS: Virtual tunnel interface**
    - **IP: MPLS Support**
    - **802.1d Ethernet Bridging**
        - **Bridge: MPLS support**

6. Posteriormente e ainda dentro do menu **Networking Options** será necessário ainda aceder a dois submenus,
    - **The IPv6 protocol**
        - **IPv6 : MPLS Support**
    - **Network packet filtering framework (Netfilter)**
        - **Core Netfilter Configuration**
            - **Netfilter Xtables support (req.ip_tables)**
                - **"mpls" target support**
        - **Bridge: Netfilter Configuration**
            - **Ethernet Bridge tables (ebtables) support**
                - **ebt: MPLS target support**

7. Foi observado que existe um erro na compilação devido à opção **Frame buffer Console Support** que deverá ser desseleccionada, para tal aceda à sequência de menus abaixo indicada.
   No menu inicial **Linux Kernel Configuration**:
   - **Device Drivers**
     - **Graphics Support**
       - **Console display driver support**
         - **Frame buffer Console support**

   Para desseleccionar a opção prima a tecla n (deixará de ver o carácter * que indica a selecção da opção).

8. Agora é necessário salvar as configurações efectuadas. De forma a não escrever por cima das configurações do *kernel* actualmente instalado deverá ser salva a configuração num ficheiro diferente do inicialmente indicado.
   De novo no menu principal seleccione a opção **Save an Alternate Configuration** file indicando o nome **.config** para o ficheiro de configurações.

9. Estando todas das configurações necessárias efectuadas podemos agora sair do menu de configuração passando para a compilação e instalação do *kernel*, com a seguinte sequência de comandos:

   ```
   #make
   #make modules            (vai devolver um warning que pode ser ignorado)
   #make modules_install
   #make install
   ```

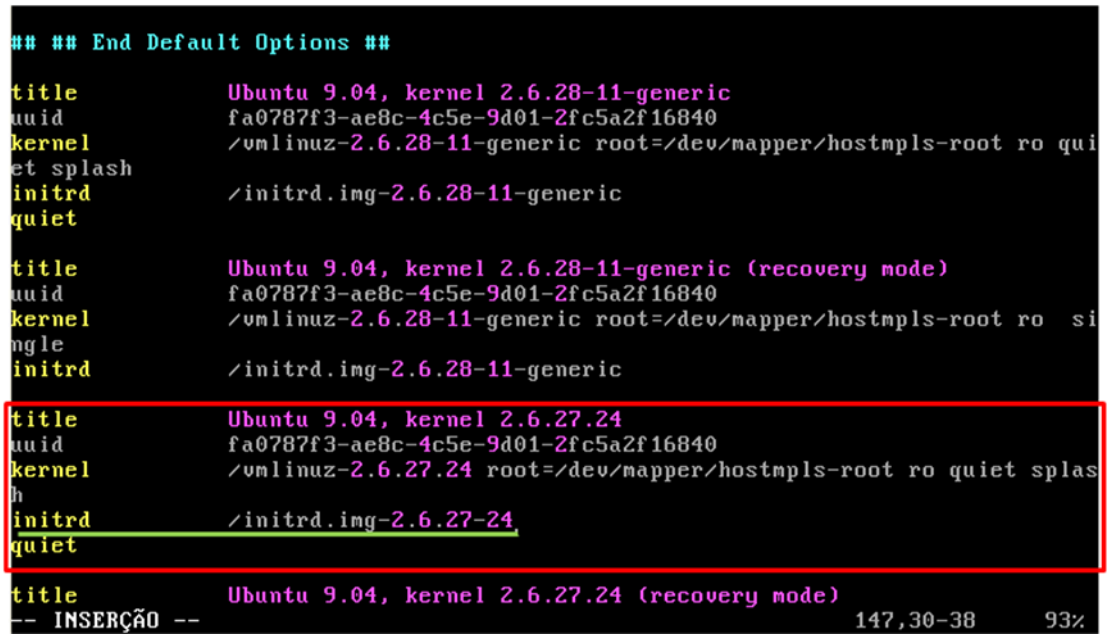
10. Agora que o *kernel* foi instalado com sucesso são necessários mais uns pequenos passos
    - Criar o ficheiro **initrd** para o *kernel* que acabamos de instalar, bastando para tal correr o seguinte comando na directoria **/boot**

      ```
      #mkinitramfs /lib/modules/2.6.27.24/ -o initrd.img-2.6.27-24
      ```

    - É então necessário actualizar o **grub** :

      ```
      #update grub
      ```

- Para finalizar é necessário modificar o menu de arranque editando o ficheiro **/boot/grub/menu.lst.** Na figura abaixo a edição do ficheiro **menu.lst**, a vermelho a nova entrada adicionada relativa ao *kernel* que acabamos de instalar. A única alteração que é necessário fazer é a adição da linha relativa ao **initrd**, ficheiro que criamos anteriormente, a verde a linha adicionada.



Uma outra opção que poderá também ser muito útil prende-se com o tempo em que o menu aparece no arranque antes de iniciar o *kernel* seleccionado por defeito. A opção respectiva é o **timeout**, que foi alterado para 5 segundos (eram 3 por defeito), esta opção aparece nas linhas iniciais do ficheiro **menu.lst**.

Será necessário colocar uma cópia de determinados ficheiros do novo *kernel* em certas pastas, de forma a evitar erros de compilação, assim torna-se necessário:

- Copiar **shim.h** do novo *kernel* para **/usr/include/Linux**. Da pasta onde estão os ficheiros do *kernel* o comando será o seguinte:

  #cp linux-2.6.27/include/linux/shim.h /usr/include/linux/.

- Copiar **rtnetlink.h** do novo *kernel* para **/usr/include/linux**, comando análogo ao primeiro uma vez que a localização dos ficheiros é na mesma directoria.

  #cp linux-2.6.27/include/linux/rtnetlink.h usr/include/linux/.

- Copiar **shim.h** do novo *kernel* mas da pasta **net** para **/usr/include/net**:

  #cp linux-2.6.27/include/net/shim.h /usr/include/net/.

- Copiar **mpls.h** do novo *kernel* da pasta **net** para **/usr/include/net**:

  #cp linux-2.6.27/include/net/mpls.h /usr/include/net/.

É também necessário colocar o ficheiro **mpls.h** fornecido, que é diferente do ficheiro com o mesmo nome da pasta **/include/linux/**do *kernel*, na pasta**/usr/include/linux.**

Reiniciar o Linux e escolher o novo *kernel* instalado.

Instalar IPRoute, IPTables e EBTables:

Para a instalação do software **IPRoute**, **IPTables** e **EBtables** é necessário que os seguintes pacotes de software sejam previamente instalados:

- libdb-dev → **#apt-get install libdb-dev**
- bison → **#apt-get install bison**
- flex **#apt-gcdet install flex**

Compilação e instalação:

- **IPRoute**
  Assim como para o *kernel* o pacote para instalação deverá ser convertido para o formato **.deb.**

  #alien -d iproute-2.6.27-2.fc10mpls.1.963.src.rpm

  O ficheiro **iproute_2.6.27-3_i386.deb** será gerado.

  Descompactar para uma pasta especificada

  #dpkg –x iproute_2.6.27-3_i386.deb iproute

  Dentro da pasta para onde foram extraídos os ficheiros (**iproute**) existe um ficheiro **iproute2-2.6.27.tar.gz** que deve ser descompactado:

  #tar –zxvf iproute2-2.6.27.tar.gz

  Mudar para a directoria **iproute2-2.6.27**, criada na descompactação do ficheiro anterior, compilar e instalar através dos seguintes comandos:

  #./configure

  #make

  #make install

- **IPTables**
  Procedimento em tudo semelhante ao anteriormente descrito, o ficheiro a usar é neste caso o **iptables-1.4.1.1-2.fc10mpls.1.963.src.rpm**

Para descompactar deveremos usar a opção **-jxvf** devido à diferente extensão do ficheiro **.tar.bz2**

- **EBTables**

  Procedimento semelhante ao anteriormente descrito, o ficheiro é neste caso o **ebtables-2.0.8-5.fc10.mpls.1.963.src.rpm,** no entanto, para este software será necessário alterar o **Makefile** antes da instalação, a vermelho na imagem a baixo as linhas alteradas.

```
# ebtables Makefile

PROGNAME:=ebtables
PROGRELEASE:=2
PROGVERSION_:=2.0.8
PROGVERSION:=$(PROGVERSION_)-$(PROGRELEASE)
PROGDATE:=May\ 2007

# default paths
LIBDIR:=/usr/lib
MANDIR:=/usr/local/man
BINDIR:=/usr/local/sbin
ETCDIR:=/etc
INITDIR:=/etc/init.d
SYSCONFIGDIR:=/etc/susctl.d
DESTDIR:=

CFLAGS:=-Wall -Wunused -shared
CFLAGS_SH_LIB:=-fPIC
CC:=gcc
LD:=ld

ifeq ($(shell uname -m),sparc64)
CFLAGS+=-DEBT_MIN_ALIGN=8 -DKERNEL_64_USERSPACE_32
```

Neste caso a sequência para instalação será apenas:

```
#make
#make install
```

<u>Instalar Quagga 0.99.6 com suporte LDP</u>:

Para a instalação do software *Quagga* é necessário que os seguintes pacotes de software sejam previamente instalados:

- automake → **#apt-get install automake**
- libtool → **#apt-get install libtool**
- libreadline → **#apt-get install libreadline-dev**

Descompactar os ficheiros relativos ao *Quagga* e ao LDP:

```
#tar -zxvf quagga-0.99.6.tar
```

```
#tar -zxvf ldp-portable_noipv6.tar
```

O software **patch**, que serve para aplicar diferenças textuais entre dois programas, deve ser efectuado de forma a efectuar as alterações necessárias no *Quagga* para o suporte LDP.

O ficheiro com as modificações a serem aplicadas está encontra-se em **mpls-ldp-portable/quagga-mpls.diff**. Este ficheiro já não é igual ao original, extraído de **ldp-portable.tar**, foi modificado manualmente de forma a serem comentadas as partes relativas ao IPv6 e a compilar correctamente o *Quagga*.

Assim dentro da directoria **quagga-0.99.6** é necessário fazer:

```
#patch -p1 < ../mpls-ldp-portable/quagga-mpls.diff
```

No decorrer do processo de **patch** aparece a seguinte situação:

```
patching file lib/zmpls.h
can't find file to patch at input line 34384
Perhaps you used the wrong -p or --strip option?
The text leading up to this was:
--------------------------
|diff --exclude=.p4config -uNr quagga/make-rpm-jleu quagga-mpls/make-rpm-jleu
|--- quagga/make-rpm-jleu       2007-01-16 00:24:46.000000000 -0500
|+++ quagga-mpls/make-rpm-jleu  2007-01-29 21:34:37.000000000 -0500
--------------------------
File to patch:
Skip this patch? [y] y
Skipping patch.
```

Foi detectado a falta de um ficheiro para fazer patch, no entanto este não é importante por isso pode ser ignorado, para tal na situação **File to patch:** prima a tecla **Enter** e posteriormente **y** em **Skip this patch?**, o patch deverá então continuar sem quaisquer erros.

É necessário agora efectuar a seguinte sequência de comandos:

```
#aclocal
```

```
#autoheader
```

```
#automake
```

```
#libtoolize
```

```
#autoconf
```

Para poder efectuar a compilação com GNU é necessário indicar onde vai ser feita a "lincagem" no ficheiro **/etc/ld.so.conf.d/i486-linux-gnu.conf** adicionando no final do ficheiro uma linha com **/usr/local/lib**. Posteriormente correr o comando **ldconfig**.

Vamos agora proceder à configuração:

```
#./configure --enable-mpls --disable-ipv6 --enable-vtysh --enable-user=root --enable-group=root --with-gnu-ld CFLAGS=-Wall
```

As seguintes linhas devem aparecer durante o processo de configuração:

```
checking for getaddrinfo... yes
checking whether does this OS have IPv6 stack... disabled
checking whether does this OS have MPLS stack... MPLS Linux
checking for inet_ntop in -lc... yes
checking for inet_pton in -lc... yes
```

```
config.status: creating watchquagga/Makefile
config.status: creating ldpd/Makefile
config.status: creating ospf6d/Makefile
```

Para terminar a instalação:

```
#make

#make install
```

Ao tentar iniciar o **ospfd** pela primeira vez, bem como os outros aplicativos que irá precisar como **zebra** ou **ldpd**, obterá um erro indicando que falhou a abertura do ficheiro de configuração. Para evitar que tal aconteça e que o software seja inicializado correctamente torna-se necessário criar, através do comando **touch**, os ficheiros de inicialização dos referidos programas. Abaixo o referido procedimento bem como a inicialização dos vários programas pela ordem necessária.

```
# touch /usr/local/etc/zebra.conf

# zebra &

# touch /usr/local/etc/ospfd.conf

# ospfd &

# touch /usr/local/etc/ldpd.conf

# ldpd &
```

Para efectuar configurações em qualquer destas aplicações usar a interface por linha de comandos **vtysh**.

Anexo II: Instalação do "MPLS for Linux"

Anexo III: Resultados dos testes às mensagens LDP

A tabela seguinte contém os resultados dos testes realizados às mensagens LDP. Estes resultados são apresentados para cada um dos modos de funcionamento. Os testes positivos são identificáveis pela letra "S". A letra "N" identifica os testes que indicaram lacunas na solução. Os quadrados a preto indica que determinado modo de funcionamento não faz sentido para o teste em questão.

Mensagem	**Descrição do teste**		**Resultados**										
			Modos de funcionamento										**Comentários**
			1	2	3	4	5	6	7	8	9	10	
Hello	1	Verificar que o DUT envia mensagens de *hello*.	S	S	S	S	S	S	S	N	N	N	
	2	Verificar que o DUT ajusta o valor do seu temporizador se receber *hellos* com valor de temporizador menor.	S	S	S	S	S	S	S	N	N	N	
	3	Verificar que o DUT mantém o valor do seu temporizador se receber *hellos* com valor de temporizador maior.	S	S	S	S	S	S	S	N	N	N	
	4	Verificar que o DUT ajusta o valor do seu temporizador para 15s se receber *hellos* com valor de temporizador nulo.	S	S	S	S	S	S	S	N	N	N	
	5	Verificar que o DUT ignora, nos *hellos* recebidos, o campo *Reserved* do TLV *Common Hello Parameters*,se este vier com valor diferente de zero.	S	S	S	S	S	S	S	N	N	N	
	6	Verificar que o DUT envia *hellos* com o campo *Reserved* preenchido a zeros.	S	S	S	S	S	S	S	N	N	N	
Inicialização	7	Verificar que o DUT adopta o modo de distribuição *Unsolicited* se receber mensagem de inicialização com modo *Unsolicited* e estiver configurado para *On Demand*.	S					S	S	N	N	N	O N2X, sendo um equipamento de testes, não suporta modos de distribuição diferentes.
	8	Verificar que o DUT envia mensagens de inicialização preenchidas com o modo de distribuição configurado.	S	S	S	S	S	S	S	N	N	N	
	9	Verificar que o DUT envia mensagens de inicialização com o campo *Loop Dectection* preenchido correctamente, dependendo se tem a detecção de *loops* activa ou não.	S	S	S	S	S	S	S	N	N	N	
	10	Verificar que o DUT envia mensagens de inicialização com o campo *Path Vector Limit* preenchido a zeros se a detecção de *loops* estiver desactivada.	S	S	S	S	S	S	S	N	N	N	
	11	Verificar que o DUT envia mensagens de inicialização com o campo *Path Vector Limit* preenchido com o valor configurado se a detecção de *loops*	S	S	S	S	S	S	S	N	N	N	

		estiver activada.												
	12	Verificar que o DUT ignora o campo Reserved e transmite zeros neste campo nas mensagens de inicialização.	S	S	S	S	S	S	S	N	N	N		
	13	Verificar que o DUT adopta o valor do temporizador *keepalive* se receber uma mensagem de inicialização com menor valor (neste temporizador).	S	S	S	S	S	S	S	N	N	N		
Keepalive	14	Verificar que o DUT envia mensagens de *keepalive* de acordo com a periodicidade configurada.	S	S	S	S	S	S	S	N	N	N		
Endereço	15	Verificar que o DUT envia mensagens de endereço antes de enviar qualquer mensagem de pedido ou mapeamento de etiqueta.	S	S	S	S	S	S	S	N	N	N		
	16	Verificar que o DUT envia mensagens de endereço com todos os endereços das interfaces configuradas.	S	S	S	S	S	S	S	N	N	N		
	17	Verificar que o DUT envia mensagens de endereço com a família de endereços IPv4.	S	S	S	S	S	S	S	N	N	N		
	18	Verificar que o DUT envia uma notificação *Unsupported Address Family* se receber uma mensagem de endereço com família de endereços diferente de IPv4.	N	N	N	N	N	N	N	N	N	N		Em vez da notificação que deveria enviar, o DUT envia uma notificação de shutdown.
Remoção de endereço	19	Verificar que o DUT envia mensagem de remoção de endereço quando é desactivada uma interface. Verificar que o endereço enviado na mensagem é o correcto.	S	S	S	S	S	S	S	N	N	N		
Mapeamento de etiqueta	20	Verificar que o DUT inclui o TLV *Request Message ID* na mensagem de mapeamento quando recebe uma mensagem de pedido de etiqueta.	N	N			N	N	N	N	N	N		Ao responder a um pedido não está a incluir o TLV Request Message ID. Por consequência, o N2X responde com a mensagem de libertar etiqueta.
	21	Verificar que o DUT envia mensagem de mapeamento quando recebe um pedido para um FEC para o qual já tinha sido fornecido mapeamento. O pedido não pode ser duplicado.	S					S	S	N	N	N		
	22	Verificar que o DUT envia mensagem de mapeamento quando recebe um pedido para um FEC para o qual é o LER de saída.	S	S			S	S	S	N	N	N		
	23	Verificar que o DUT envia mensagem de mapeamento quando recebe um pedido para um FEC para o qual ainda não tem mapeamento do LSR de *downstream.*						S	S		N	N		
	24	Verificar que o DUT envia mensagem de mapeamento quando recebe um mapeamento de downstream, o qual já tinha sido fornecido, mas este tem atributos – *Hop Count* e *Path Vector* – diferentes	S	S	S		S	S		N	N	N		O *Path Vector* parece não estar a funcionar.
	25	Verificar que o DUT envia mensagem de mapeamento quando reconhece um novo FEC para o qual não é o LER de			S	S								

		saída.											
	26	Verificar que o DUT envia mensagem de mapeamento quando reconhece um novo FEC para o qual é o LER de saída.		N	S	S	N						No modo ordenado, não está a enviar mapeamentos quando é o LER de saída. Só envia para as redes directamente ligadas.
	27	Verificar que o DUT envia mensagem de mapeamento quando recebe um mapeamento do LSR de *downstream* e tem um pedido pendente para esse FEC.	S	S			S			N			
	28	Verificar que o DUT envia mensagem de mapeamento quando reconhece um novo FEC e já tem o mapeamento do LSR de *downstream.*		S			S						
	29	Verificar que o DUT não envia mensagem de mapeamento quando recebe um mapeamento de *downstream*, o qual já tinha sido fornecido com os mesmos atributos – *Hop Count* e *Path Vector*.	N	N	N		N	N		N		N	
	30	Verificar que o DUT não envia mensagem de mapeamento quando ainda não recebeu o mapeamento para um determinado FEC do seu LSR de *downstream.*		S			S						
	31	Verificar que o DUT envia uma notificação ou mensagem de libertar etiqueta quando recebe um mapeamento com um elemento do TLV FEC do tipo *"Wildcard"*.	N	N	N	N	N	N	N	N	N	N	O *deamon ldpd* está a rebentar quando recebe um mapeamento com *wildcard.*
Pedido de etiqueta	32	Verificar que o DUT envia mensagem de pedido se detectar uma alteração no próximo salto para um determinado FEC e ainda não foi fornecido mapeamento desse LSR.	N	N				N	N	N	N	N	Quando o DUT reconhece um novo FEC, não está a enviar pedidos em qualquer situação.
	33	Verificar que o DUT envia mensagem de pedido se detectar um novo FEC para o qual ainda não tem mapeamento do LSR de *downstream*.	N	N				N	N	N	N	N	
	34	Verificar que o DUT envia notificação (*No Route*) se recebeu um pedido para o qual não tem rota na tabela de encaminhamento.	N	N				N	N	N	N	N	
	35	Verificar que o DUT não envia mensagem de pedido se surgir um novo FEC para o qual não tem mapeamento do próximo salto.			S	S	S						Como o DUT não envia pedidos, esta situação corre bem…
	36	Verificar que o DUT não envia mais qualquer mensagem de pedido para o seu par LSR, se receber uma notificação (*No Resources*) como resposta a um dos seus pedidos.	N	N				N	N	N	N	N	*Deamon ldpd* estoira.
	37	Verificar que o DUT poderá enviar múltiplas mensagens de pedido para o mesmo FEC se receber pedidos que não são duplicados.								N	N	N	
	38	Verificar que o DUT não envia múltiplas mensagens de pedido para o	N					N	N				

		mesmo FEC quando recebeu pedidos que não são duplicados.										
	39	Verificar que o DUT envia mensagem de pedido se surgiu um novo FEC para o qual é o LER de entrada.	N				N	N	N	N	N	Já foi visto atrás que o DUT não envia mensagens de pedido a FECs novos… Só retransmite pedidos.
	40	Verificar que o DUT envia mensagem de pedido se receber uma notificação (*Resources Available*) de um LSR que anteriormente tinha enviado a notificação *No Resources*.	N	N			N	N	N	N	N	
	41	Verificar que o DUT envia uma notificação se receber um pedido com um elemento do TLV FEC do tipo *"Wildcard"*.	N	N		N	N	N	N	N	N	
Abortar pedido de etiqueta	42	Verificar que o DUT ignora a recepção de mensagens de abortar pedido para um determinado FEC, se o LSP correspondente já tiver estabelecido.	N	N		N	N	N	N	N	N	O DUT ignora a recepção destas mensagens quando o LSP já está estabelecido. No entanto, na notificação que ele envia como resposta não inclui o TLV *Message ID*.
	43	Verificar que o DUT responde com uma notificação (*Label Request Aborted*) se receber uma mensagem de abortar pedido e tiver um pedido pendente (ainda não enviou o mapeamento).	N	N					N			Envia notificação, mas de fatal error.
	44	Verificar que o DUT não responde com uma notificação (*Label Request Aborted*) se receber uma mensagem de abortar pedido e o TLV "*Messsage ID*" não igualar o mesmo TLV no pedido pendente.	N	N					N			
	45	Verificar que o DUT envia múltiplas mensagens de abortar pedido para um FEC se detectar uma alteração no próximo salto e tiver enviado múltiplos pedidos para esse mesmo FEC.							N			
	46	Verificar que o DUT inclui o TLV *Request Message ID* com o *Message ID* correcto na notificação *Label Request Aborted* quando o pedido é abortado com sucesso.	N	N					N			Não envia mensagens de notificação abortar pedido de etiqueta.
	47	Verificar que o DUT não envia mensagens de abortar pedido se tiver outros pedidos pendentes para o mesmo FEC (recebidos do LSR de *upstream*) e se já tiver enviado um pedido para o LSR de *downstream*.	S	S								
Remover etiqueta	48	Verificar que o DUT remove todas as etiquetas enviadas no sentido *upstream* se receber uma mensagem de remover etiqueta do LSR de *downstream* com wildcard.	N	N		N			N			
	49	Verificar que o DUT remove todas as etiquetas enviadas no sentido *upstream* para um FEC se receber uma mensagem de remover etiqueta com o TLV *Label*	S	S		S			N			

		do LSR de *downstream.*											
	50	Verificar que o DUT envia mensagem de remover etiqueta para o LSR de *upstream* se receber a mesma mensagem do LSR de *downstream*.	S	S			S			N			
	51	Verificar que o DUT envia mensagem de remover etiqueta se detectar alteração no próximo salto e já tiver distribuído mapeamentos para os seus LSRs vizinhos.	S	S	S	S	S	S	S	N	N	N	
Libertar etiqueta	52	Verificar que o DUT envia mensagem de libertar etiqueta se já tiver mapeamento para determinado FEC e receber outro mapeamento com etiqueta diferente.	N	S			S	N		N		N	
	53	Verificar que o DUT liberta todas as etiquetas de um determinado FEC se receber uma mensagem de libertar etiqueta para esse FEC.	S	S	S	S	S	S	S	N	N	N	
	54	Verificar que o DUT envia mensagem de libertar etiqueta se o LSR que enviou o mapeamento já não é o próximo salto.	S	S				S	S	N	N	N	
	55	Verificar que o DUT envia mensagem de libertar etiqueta se receber um mapeamento dum LSR que não é o próximo salto.	S	S				S	S	N	N	N	
	56	Verificar que o DUT envia mensagem de libertar etiqueta no sentido *downstream* se receber uma mensagem de remover etiqueta.	S	S	S	S	S	S	S	N	N	N	
Notificação	57	Verificar que o DUT envia uma notificação de "*Bad LDP Identifier*" se receber qualquer mensagem LDP com um identificador LDP desconhecido (LSR ID).	N	N	N	N	N	N	N	N	N	N	
	58	Verificar que o DUT envia uma notificação de "*Bad Protocol Version*" se receber qualquer mensagem LDP com uma versão de protocolo LDP não suportada.	N	N	N	N	N	N	N	N	N	N	
	59	Verificar que o DUT envia uma notificação de "*Bad PDU Length*" se receber qualquer LDP com um tamanho de PDU superior ao máximo (>4096 bytes) ou inferior ao mínimo (<14 bytes).	N	N	N	N	N	N	N	N	N	N	
	60	Verificar que o DUT envia uma notificação de "*Unknown Message Type*" se receber uma mensagem LDP com tipo desconhecido (<0x8000).	N	N	N	N	N	N	N	N	N	N	
	61	Verificar que o DUT envia uma notificação de "*Bad Message Length*" se receber qualquer mensagem LDP com tamanho inválido.	N	N	N	N	N	N	N	N	N	N	
	62	Verificar que o DUT envia uma notificação de "*Missing Message Parameters*" se receber qualquer mensagem LDP sem alguns parâmetros obrigatórios.	N	N	N	N	N	N	N	N	N	N	
	63	Verificar que o DUT envia uma notificação de "*Bad TLV Length*" se receber qualquer mensagem LDP com o tamanho do TLV demasiado grande.	N	N	N	N	N	N	N	N	N	N	
	64	Verificar que o DUT envia uma notificação de "*Unknown TLV*" se	N	N	N	N	N	N	N	N	N	N	

		receber qualquer mensagem LDP com tipo de TLV desconhecido (<0x8000).											
	65	Verificar que o DUT envia uma notificação de "*Malformed TLV Value*" se receber qualquer mensagem LDP com TLVs preenchidos por valores errados.	N	N	N	N	N	N	N	N	N	N	
	66	Verificar que o DUT envia uma notificação de "*Unknown FEC*" se receber qualquer mensagem com um TLV FEC de tipo errado.	N	N	N	N	N	N	N	N	N	N	
	67	Verificar que o DUT retransmite uma notificação para o LSR de *upstream*, quando recebe do LSR de *downstream* uma notificação com o bit 'F' a 1.	N	N	N	N	N	N	N	N	N	N	
	68	Verificar que o DUT não responde com nenhuma mensagem quando recebe do LSR de *downstream* uma notificação com o bit 'F' a 0.	S	S	S	S	S	S	S	N	N	N	
	69	Verificar que o DUT remove todas as etiquetas descobertas sobre uma sessão LDP se a ligação TCP cair.	S	S	S	S	S	S	S	N	N	N	

Referências

[1] "*Mobile Backhaul Network Migration: Building an Evolution-Ready Backhaul*", *White paper* de *Ceragon Networks*, Ran Avital, Outubro, 2007

[2] "*RFC3031, Multiprotocol Label Switching Architecture*", Janeiro, 2001

[3] "*MPLS Fundamentals*", *Chapter 2: "MPLS Architecture"*, Cisco Systems Inc., *Luc De Ghein*, 2007

[4] "*MPLS Fundamentals*", *Chapter 4: "Label Distribution Protocol"*, Cisco Systems Inc., *Luc De Ghein*, 2007

[5] "*RFC5036, LDP Specification*", Outubro, 2007.

[6] "*MPLS Fundamentals*", *Chapter 8: "MPLS Traffic Engineering"*, Cisco Systems Inc., *Luc De Ghein*, 2007

[7] "*QoS for IP/MPLS Networks*", *Chapter* 2: "*MPLS TE Technology Overview*", Cisco Systems Inc., Santiago Alvarez, 2 de Junho, 2006

[8] "*MPLS Fundamentals*", *Chapter 7: "MPLS VPN"*, Cisco Systems Inc., *Luc De Ghein*, 2007

[9] "*MPLS/BGP Virtual Private Network*", *Spirent White Paper*, 2002

[10] *"RFC 2474, Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers"*, Dezembro, 2008

[11] "*MPLS Fundamentals*", *Chapter 12: "MPLS and Quality of Service"*, Cisco Systems Inc., *Luc De Ghein*, 2007

[12] "*Mobile Backhaul Evolution and Convergence*", *NEC White Paper*, Janeiro, 2010

[13] "*Mobile Backhaul Evolution: for deploying Mobile Next Generations Networks (Mobile NGN)*", *NEC White Paper*, Fevereiro, 2007

[14] "*Use of MPLS technology in Mobile Backhaul Networks*", *IP/MPLS Forum White Paper*, Fevereiro, 2008

[15] "*The Case for IP/MPLS in Backhaul*", Nikhil Shah, *Juniper Networks*, 16 de Maio, 2008

[16] *"MPLS in Mobile Backhaul Networks Framework and Requirements"*, *IP/MPLS Forum 20.0.0*, Outubro, 2008

Referências

[17] *Broadband Forum*, Junho 2010, URL: http://www.broadband-forum.org/

[18] "*WinPath3 Family*", *Product brief from Wintegra*, Junho 2010, URL: http://www.wintegra.com/page/winpath3-overview

[19] *Wintegra*, Junho 2010, URL: http://www.wintegra.com/

[20] *MPLS for Linux*, Março 2010, URL: http://mpls-linux.sourceforge.net/

[21] *Metaswitch*, Maio 2010, URL: http://www.metaswitch.com/

[22] *IP Infusion*, Maio 2010, URL: http://www.ipinfusion.com/

[23] *Documentation for Quagga*, Maio 2010, URL: http://www.quagga.net/docs.php

[24] "*LDP Conformance Implementation Agreement*", *MPLS Forum 3.0*, 4 de Dezembro, 2002

[25] "*MPLS Conformance and Performance Testing*", *White Paper from IXIA*, 2004

[26] "RFC4447, *Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)*", Abril, 2006

[27] *Address Family Numbers*, Junho 2010, URL: http://www.iana.org/assignments/address-family-numbers/

[28] "*Internetworking Technology Overview*", *Part 6* "*Routing Protocols*", Cisco Systems Inc., *Kim Lew*, Junho, 1999

[29] *Configuration Guide*: "*MPLS Label Distribution Protocol (LDP)*", Cisco Systems Inc., Junho 2010, URL: http://www.cisco.com/

[30] Juniper Networks, Junho 2010, URL: http://www.juniper.net/