UNIVERSIDADE SALVADOR – UNIFACS
PROGRAMA DE PÓS-GRADUAÇÃO EM REDES DE COMPUTADORES
MESTRADO PROFISSIONALIZANTE EM REDES DE COMPUTADORES

MARCOS GUIMARÃES FONSECA

RE-ROTEAMENTO RÁPIDO EM REDES MPLS (MULTIPROTOCOL LABEL SWITCHING):
UM MÉTODO DE PROTEÇÃO DE MÚLTIPLOS SEGMENTOS

Salvador
2005

MARCOS GUIMARÃES FONSECA

RE-ROTEAMENTO RÁPIDO EM REDES MPLS (MULTIPROTOCOL LABEL SWITCHING):
UM MÉTODO DE PROTEÇÃO DE MÚLTIPLOS SEGMENTOS

Dissertação apresentada ao Curso de Mestrado Profissional em Redes de Computadores, Universidade Salvador - UNIFACS, como requisito parcial para obtenção do grau de Mestre.

Orientador: Prof. Dr. José Augusto Suruagy Monteiro.

Salvador
2005

Ficha Catalográfica
(Elaborada pelo Sistema de Bibliotecas da Universidade Salvador - UNIFACS)

Fonseca, Marcos Guimarães

Re-roteamento rápido em redes MPLS (Multiprotocol Label Switching): um método de proteção de múltiplos segmentos. / Marcos Guimarães Fonseca. – Salvador, 2005.

81f. : il.

Orientador: Prof. Dr. José Augusto Suruagy Monteiro.

Dissertação apresentada ao Curso de Mestrado em redes, Universidade Salvador – UNIFACS, como requisito parcial para a obtenção do grau de Mestre.

1. Redes de Computadores. 2. Engenharia de Tráfego. 3. Padrão MPLS. I. Monteiro, José Augusto Suruagy, orient. II. Universidade Salvador – Unifacs. III.Título.

CDD: 004.6

TERMO DE APROVAÇÃO

MARCOS GUIMARÃES FONSECA

RE-ROTEAMENTO RÁPIDO EM REDES MPLS (MULTIPROTOCOL LABEL SWITCHING): UM MÉTODO DE PROTEÇÃO DE MÚLTIPLOS SEGMENTOS

Dissertação aprovada como requisito parcial para obtenção do grau de mestre em Redes de Computadores, Universidade Salvador – Unifacs, pela seguinte banca examinadora:

José Suruagy Monteiro (Orientador) _____
Doutor em Computer Scince, University of Califórnia Los Angeles, U.C.L..A, Estados Unidos.
Universidade Salvador - UNIFACS

William Ferreira Giozza _____
Doutor em Systemes D`Informatique, Universite de Paris VI, UP VI, França.
Universidade Salvador - UNIFACS

Waslon Terllizzie Araújo Lopes _____
Doutor em Engenharia Elétrica, Universidade Federal de Campina Grande (UFCG), Brasil.
Faculdade de Ciência e Tecnologia - Área1

Salvador, de de 2005.

.

AGRADECIMENTOS

Quero agradecer, em primeiro lugar, à minha família por seu exemplo de vida, pelo apoio, amor e dedicação sempre demonstrados. Tenho certeza de que sem eles não teria vencido esta etapa.

Também, como não poderia deixar no esquecimento, quero agradecer a Deus e aos irmãos de luz.

Registro, de forma singela, meus sinceros agradecimentos a todos os professores do Curso de Mestrado em Redes de Computadores da UNIFACS, em especial ao meu orientador Prof. Dr. José Augusto Suruagy Monteiro, pela atenção e disposição sempre demonstradas durante o andamento deste trabalho.

À direção da Fundação Visconde de Cairu, pela bolsa de estudo concedida.

Aos colegas de Mestrado, em especial a Messias Bittencourt Figueredo, que foi de grande importância para que eu pudesse alcançar meus objetivos neste trabalho.

RESUMO

O aprimoramento e desenvolvimento de técnicas de recuperação de falhas em redes de computadores são de suma importância para diversas aplicações, especialmente, aquelas em tempo real. O uso combinado de comutação de rótulos multiprotocolo *Multiprotocol Label Switching* (MPLS) com a Engenharia de Tráfego *Traffic En*gineering (TE) provê métodos de recuperação de falhas em redes mais rápidos e eficazes do que outros protocolos e tecnologias. Isso faz com que o atraso e a perda de pacotes sejam reduzidos de forma significativa. Nessa dissertação são analisados três métodos de proteção de falhas em redes que utilizam a tecnologia MPLS, a saber: Proteção de Caminho Global (MPCG), Proteção de Caminho Reverso (MPCR) e Proteção de Múltiplos Segmentos (MPMS). Esses métodos foram implementados utilizando o simulador de rede Network Simulator (NS). A partir dos resultados de simulação, pode-se concluir que o método de proteção MPMS apresenta melhor eficiência em relação a perdas de pacotes, quando comparado ao método MPCG, bem como melhor eficiência em relação ao atraso, quando comparado ao método MPCR.

Palavras-Chaves: *Multiprotocol Label Switching* (MPLS); Engenharia de Tráfego; Proteção.

ABSTRACT

The research of fault recovery techniques in computer networks plays an important role for many applications, specially for the real-time ones. The combined use of Multiprotocol Label Switching (MPLS) and Traffic Engineering (TE) provides faster and more efficient fault recovering techniques than those obtained through other protocols and techniques. As a consequence, packet delays and losses are significantly reduced. Three methods based on MPLS technology are discussed in this master thesis: The Global Path Protection Method (GPPM), the Reverse Path Protection Method (RPPM), and the Multiple Segments Protection Method (MSPM). These methods were implemented by using the Network Simulator (NS2). From simulation results, one can see that MSMP is more efficient than GPPM in terms of packet loss and more efficient than RPPM in terms of packet delay.

Keywords: *Multiprotocol Label Switching* (MPLS); Traffic Engineering; Protection.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS E ABREVIATURAS

ADM Multiplexador *add/drop*

ATM *Asynchronous Transfer Mode* = Modo de transferência assíncrono

BGP *Border Gateway Protocol* = Protocolo de gateway de borda

CBR *Constraint Based Routing* = Roteamento baseado em restrições

CR-LDP *Constraint-based Routing LDP* = Roteamento restrito de LDP

CR-LSP *Constraint-based Routing LSP* = Roteamento restrito de LSP

CSPF *Constrained Shortest Path First* = Caminho restrito mais curto primeiro

ERB *Explicit Route Information Base* = Base de informações para roteamento explícito

ER-LSP *Explicit Routed Label Switched Path* = Roteamento explícito de LSP

EXP *Experimental field* = Campo experimental

FEC *Forwarding Equivalence Class* = Classe de equivalência de encaminhamento

FIS *Fault Indication Signal* = Sinal de indicação de falha

FTN *FEC to NHLFE*

FTP *File Transfer Protocol* = Protocolo de transferência de arquivos

IGP *Internal Gateway Protocol* = Protocolo de gateway interno

IP *Internet Protocol* = Protocolo da internet

ILM *Incoming Label Mapping* = Mapa de rótulos de entrada

LDP *Label Distribution Protocol* = Protocolo de distribuição de rótulos

LER *Label Edge Router* = Roteador de borda de rótulo

LIB *Label Information Base* = Base de informações de rótulos

LSP *Label Switched Path* = Caminho de comutação de rótulos

LSR *Label Switching Routers* = Roteadores de comutação de rótulos

MNS_V2 MPLS Network Simulator Version 2

MPCG Método de proteção de caminho global

MPCR Método de proteção de caminho reverso

MPLS *Multiprotocol Label Switching* = Comutação de rótulos multiprotocolo

MPMS Método de proteção de múltiplos segmentos

MPOA *Multiprotocol* sobre ATM

NAM *Network Animator*

NHLFE *Next Hop Label Forwarding Entry* =Registro de encaminhamento por rótulos para o próximo salto

NS *Network Simulator* = Simulador de rede

PFT *Partial Forwarding Table* = Tabela parcial de encaminhamento

PHP *Penultimate Hop Popping* = Retirada do penúltimo salto

PIL *Protection Ingress LSR*

PML *Protection Merging LSR*

QoS *Quality of Service* = Qualidade de serviço

RFC *Request For Comments*

RSVP *Resource Reservation Protocol* = Protocolo de reserva de largura de banda

SPF *Shortest Path First* = Primeiro o caminho mais curto

TCP *Transmission Control Protocol* = Protocolo de controle de transmissão

TE *Traffic Engineering* = Engenharia de tráfego

TT *Traffic Trunks*

TTL *Time to live* = Tempo de vida

UDP *User Datagram Protocol* = Protocolo de datagrama do usuário

VC V*irtual Circuit* = Circuito virtual

VPN *Virtual Private Networks* = Rede privada virtual

SUMÁRIO

1 INTRODUÇÃO

Os serviços baseados em rede IP, tais como presença virtual, redes privadas virtuais, vídeo e voz necessitam de níveis elevados de garantias de serviços confiáveis e utilizáveis para os clientes. Neste contexto, as conseqüências de uma falha na rede tornam-se mais pronunciadas.

Em relação aos roteadores, existem dois tipos de falhas em uma rede de computadores. Falhas de enlace e falhas de nó. Uma falha no enlace pode ser uma fibra cortada, um problema com Multiplexador add/drop (ADM) ou vários outros problemas. Uma falha de nó pode ser algo desde um problema de alimentação até uma falha no roteador ou um roteador sendo desligado para manutenção preventiva. Não importa qual a causa, todas as falhas são uma falha de enlace ou uma falha de nó (OSBORNE; AJAY, 2002).

As falhas eram, até recentemente, preocupação exclusiva de projetistas de sistemas críticos como aviões, sondas espaciais e controle industrial de tempo real. Com a espantosa popularização das redes de computadores fornecendo os mais variados serviços, aumentou-se a dependência tecnológica de uma grande parcela da população aos serviços oferecidos. Falhas nesses serviços podem ser catastróficas para a segurança da população ou para a imagem e reputação das empresas.

O uso combinado de comutação de rótulos multiprotocolo *Multiprotocol Label Switching* (MPLS) com a Engenharia de Tráfego - *Traffic Engineering* (TE), de acordo com Awduche e outros autores (1999) provê métodos de recuperação de falhas em enlaces e em nós mais rápidos e eficazes que outros protocolos e tecnologias, o que permite reduzir o atraso e a perda de pacotes.

Segundo Rosen, Viswanathan e Callon (2001), a tecnologia MPLS combina a flexibilidade de roteamento IP e a eficiência do chaveamento no nível de enlace. A Figura 1 representa uma malha MPLS com seus principais componentes. Nessa tecnologia, cada pacote recebe um rótulo (Label) do roteador de borda de rótulo *Label Edge Router* (LER). Os pacotes são encaminhados através de um caminho comutado por rótulo, *Label Switch Path* (LSP), formado por roteadores de

comutação por rótulos *Label Switch Routers* (LSRs), e cada LSR toma decisões de encaminhamento baseado apenas no rótulo de 32 bits do pacote MPLS, utilizando uma busca exata em tabelas indexadas de porte bem menor do que as tabelas do roteamento IP. Em cada salto o LSR retira o rótulo existente e aplica um novo que informa ao próximo salto como encaminhar o pacote. O protocolo de distribuição de rótulos - *Label distribution Protocol* (LDP) é definido para distribuir os rótulos e estabelecer um LSP. A tecnologia MPLS pode também estabelecer um roteamento explícito de LSP - *Explicit Routed LSP* (ER-LSP).

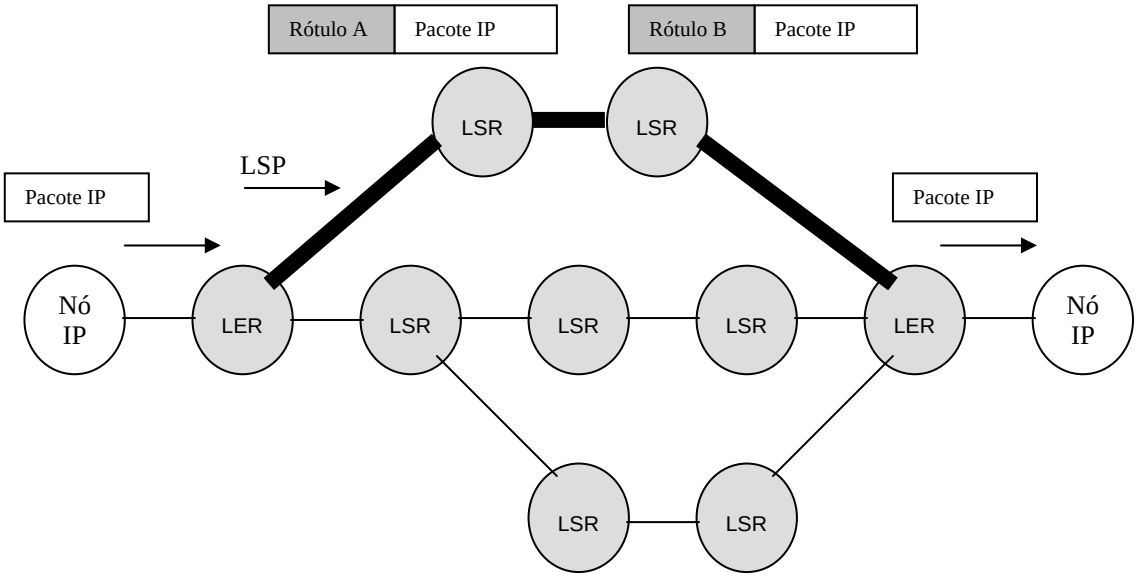


Figura 1 - Malha MPLS
Nota: Elaboração própria.

O roteamento explícito é aplicado na engenharia de tráfego, que se refere à tarefa de garantir que recursos suficientes estejam disponíveis em uma rede para atender às demandas impostas sobre ela. O controle exato do percurso por onde o tráfego flui é uma tarefa importante da engenharia de tráfego. O roteamento explícito também pode ajudar a tornar as redes mais tolerantes em casos de falha, usando uma capacidade chamada de proteção ou re-roteamento rápido.

A recuperação de falhas em redes MPLS é uma técnica para re-rotear o tráfego em torno de uma falha ou congestionamento em um LSP. De maneira geral, existem dois modelos básicos de recuperação: proteção e restauração (DAHAI XU, 2003). Na proteção, os LSPs são preestabelecidos. Na restauração, os LSPs de recuperação são criados e roteados em reação a erros na rede, ou seja, são dinamicamente criados.

Modelos de proteção podem, em geral, recuperar falhas mais rapidamente, porém são menos eficientes quanto à otimização de recursos. Por outro lado, modelos de restauração podem resistir a uma ou múltiplas falhas, mas não podem garantir o tempo de recuperação nem a quantidade de informações perdidas, tornando-os inadequados para aplicações em tempo real.

No Método de Proteção de Caminho Global (MPCG) apresentado na Figura 2, um nó de ingresso ativa o caminho de recuperação quando receber o sinal de indicação de falha *Fault Indication Signal* (FIS). Isto requer um caminho de recuperação alternativo para cada caminho de trabalho. O nó de ingresso é onde o processo de proteção é iniciado, independente da localização da falha ao longo do caminho de trabalho.

A Figura 2 representa uma topologia composta de nove nós, um caminho de trabalho formado pelos nós (1, 3, 5, 7 e 9) e um caminho de recuperação formado pelos nós (1, 2, 4, 6, 8 e 9). O nó 1 definido como *Protection Ingress LSR* (PIL), é o responsável pela função de transição quando a falha for identificada. O nó 9 é definido como *Protection Merging LSR* (PML), responsável por fazer a união entre o caminho de trabalho e o caminho de recuperação. Também está representada nessa Figura, uma falha no enlace entre os nós 5 e 7. No instante que o nó 5 detecta a falha, envia um sinal FIS em direção ao nó 1.

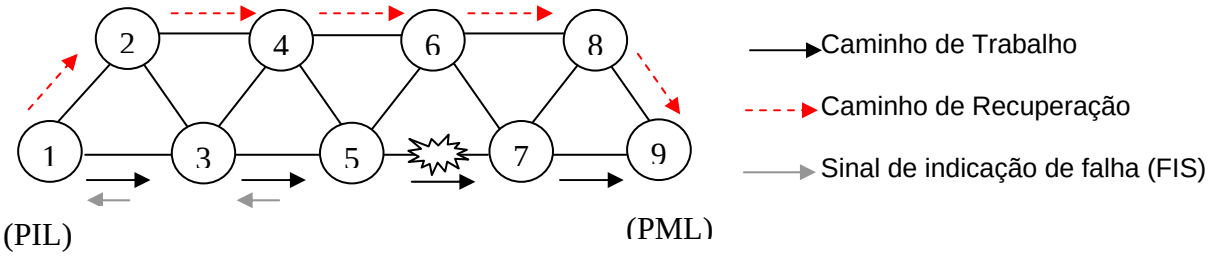


Figura 2 – Método Proteção de Caminho Global (MPCG)

O Método de Proteção de Caminho Reverso (MPCR) (HASKIN; KRISHNAN, 2000) como mostra a Figura 3, inverte o tráfego próximo ao local da falha de volta à fonte (nó de ingresso) do caminho que está sendo protegido, via um LSP de proteção Reversa. Assim que uma falha é detectada, o nó de início do enlace com

falha, redireciona o tráfego para o LSP de proteção na direção oposta, de volta para o nó de ingresso.

Na Figura 3, o LSP de proteção é representado pelos nós (5, 3, 1, 2, 4, 6, 8 e 9), o nó 1 representa a fonte (nó de ingresso). No instante que o nó 5 detecta a falha, redireciona o tráfego na direção oposta, para o LSP de proteção.

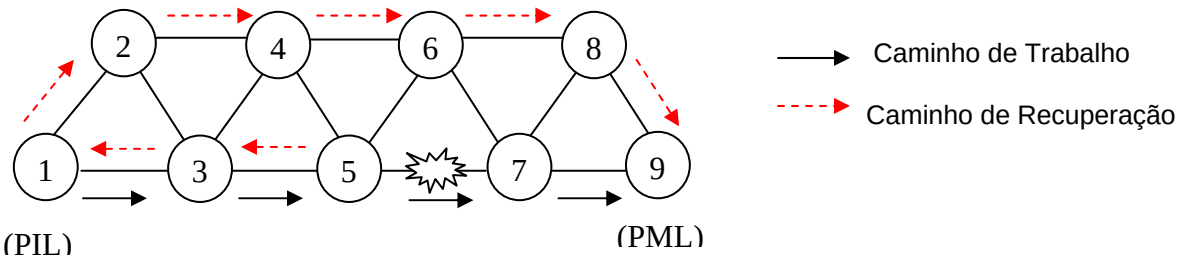
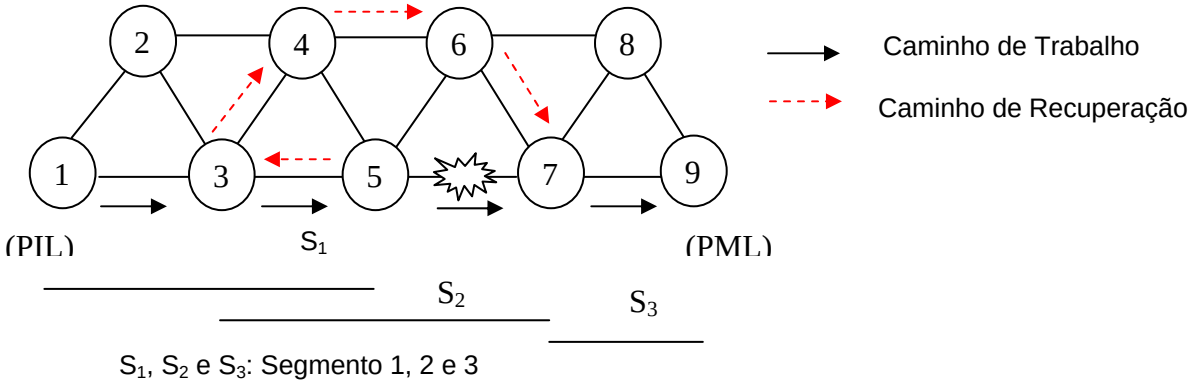


Figura 3 – Método de Proteção de Caminho Reverso (MPCR)

No Método de Proteção de Múltiplos Segmentos (MPMS) (DAHAI XU, 2003) como mostra a Figura 4, o caminho de trabalho é dividido em diversos segmentos de trabalho e cada um desses segmentos é, então, protegido com um segmento de proteção, em vez de proteger o caminho de trabalho como um todo como é feito pelos métodos de proteção de caminho. Na Proteção MPMS, o tráfego re-roteado necessita passar apenas pelo nó que inicia o segmento de proteção, em oposição à fonte como na Proteção MPCR (HASKIN; KRISHNAN, 2000).

A Figura 4 representa uma topologia cujo caminho de trabalho foi dividido em três segmentos de trabalho. S1 formado pelos nós (1, 3 e 5), S2 (3, 5 e 7) e S3 (7 e 9), protegidos respectivamente pelos segmentos de recuperação formados pelos nós (1, 2, 4 e 5), (3, 4, 6 e 7) e (7, 8 e 9). No instante que o nó 5 detecta a falha, redireciona o tráfego para o segmento de recuperação na direção oposta, de volta ao nó 3, que é o nó de início do segmento de recuperação.



S_1, S_2 e S_3: Segmento 1, 2 e 3

Figura 4 – Método de Proteção de Múltiplos Segmentos (MPMS)

O objetivo principal deste trabalho é comparar a eficiência dos métodos de proteção de caminho com o Método de Proteção de Múltiplos Segmentos (MPMS), utilizando como medidas de desempenho o número de pacotes perdidos e o atraso. Para isso, foram atingidos os seguintes objetivos específicos: O aprofundamento do conhecimento das tecnologias que envolvem esta modelagem e a implementação do Método de Proteção de Múltiplos Segmentos no simulador de redes - *Network Simulator (NS*).

O texto está organizado da seguinte forma: discute-se no Capítulo 2 a tecnologia MPLS, através de uma revisão geral dos diversos componentes e conceitos que a envolvem. No Capítulo 3, são apresentados os métodos de proteção e restauração, focando suas vantagens e desvantagens. O Capítulo 4 traz um breve tutorial do NS e do módulo MPLS do simulador de redes - *MPLS Network Simulator (MNS*) e explica a implementação realizada no simulador, além de obter e comparar os resultados das simulações. O Capítulo 5 apresenta as Conclusões do trabalho. Os APÊNDICES A e B apresentam, respectivamente, o *Script* de simulação para a topologia 1 e o Script de simulação para a topologia 2.

2 A TECNOLOGIA MPLS E ENGENHARIA DE TRÁFEGO

No roteamento IP convencional, tipo salto-por-salto, quando um pacote IP chega a um roteador, examina-se o endereço de destino no cabeçalho IP, realiza-se uma pesquisa de rota com base no prefixo de rede mais longo que coincida com o endereço de destino e encaminha-se o pacote para o próximo salto (MAGALHÃES; CARDOSO, 2001). Para cada pacote, faz-se todo este processamento, mesmo que uma série consecutiva de pacotes tenha o mesmo destino. Este procedimento pode ser fonte de dois problemas: o atraso de encaminhamento dos pacotes e o congestionamento da rede. Isso pode comprometer de forma significativa o crescimento da rede e limitar o número de aplicações disponibilizadas pela rede. A comutação IP é solução para o problema do atraso de encaminhamento, enquanto a engenharia de tráfego é solução para o problema do congestionamento. A tecnologia MPLS integra ambas as soluções (MAGALHÃES; CARDOSO, 2001).

As primeiras tecnologias para a comutação IP por rótulos utilizavam hardware do modo de transferência assíncrona - *Asynchronous Transfer Mode (ATM*) e pertenciam ao modelo Sobreposto (Overlay). Essas tecnologias preservavam a sinalização ATM e forneciam soluções para operação de protocolos de enlace ou de rede sobre ATM. A *Multiprotocol Over ATM (MPOA)* e IP-Clássico sobre ATM são representantes destas iniciativas.

Enquanto as entidades de padronização investiam no modelo Sobreposto, a indústria deixava claro, sua preferência pelo modelo Par (*Peer*). Surgiram produtos como *IP Switch, Tag Switching e IP Navigator*. Estas tecnologias utilizavam a velocidade do hardware ATM e substituíam os protocolos de sinalização ATM por protocolos de sinalização IP, mais integrados com o roteamento e endereçamento IP. As tentativas de padronização dessas tecnologias resultaram na combinação de várias tecnologias, gerando a comutação de rótulos multiprotocolo - *Multiprotocol Label Switching (MPLS)* (OSBORNE; AJAY, 2002).

Uma rede MPLS consiste de equipamentos de comutação denominados roteadores de comutação por rótulos - *Label Switch Router (LSR)*. Na rede MPLS, os caminhos comutados são estabelecidos segundo orientação por topologia

denominados caminhos comutados por rótulos - *Label Switch Path (LSP)*. Os LSPs são estabelecidos por ação de protocolos do plano de controle, ou por ação de gerência de rede. A rota estabelecida para um LSP pode ser determinada com o auxílio de protocolos de roteamento convencionais. Ou segundo um roteamento baseado em restrições como, por exemplo, roteamento na origem ou roteamento com qualidade de serviço.

Cada LSP está associado a uma classe equivalente de encaminhamento - *Forwarding Equivalent Class (FEC).* Uma FEC determina quais pacotes serão encaminhados pelo LSP. O LSR de ingresso ao receber um pacote, verifica se o mesmo pertence a uma FEC. Em caso afirmativo o pacote é encaminhado através do LSP associado a FEC. Caso contrário o pacote recebe o encaminhamento IP convencional (salto-por-salto).

A tecnologia MPLS proporciona designação, encaminhamento e comutação eficiente de fluxos de tráfego através da rede. A informação em uma rede MPLS pode ser designada a uma determinada classe de serviço, e os dados são encaminhados através de caminhos estabelecidos anteriormente, sendo feita apenas comutação, e não roteamento. Entre as suas principais características pode-se citar:

a) A agilidade no encaminhamento de pacotes proporcionada pela inspeção de rótulos no denominado roteamento explícito, onde os pacotes são analisados somente na borda de um domínio MPLS;
b) Implementação de orientação a conexão em redes IP, onde a partir da ligação entre os rótulos e classes de serviços, os pacotes são encaminhados em “caminhos virtuais”, que propicia a Engenharia de Tráfego e o Re-roteamento Rápido em caso de falhas na rede;
c) Suporte otimizado às arquiteturas de IP QoS como o *IntServ e DiffServ*;
d) Simplificação na interoperabilidade de redes com tecnologias heterogêneas, como redes IP e redes ATM.

Segundo Rosen, Viswanathan e Callon (2001), MPLS é uma tecnologia de encaminhamento de pacotes, que inclui extensões no plano de controle do protocolo de encaminhamento convencional IP.

2.1 COMPONENTES DO MPLS

A Figura 1 ilustra os elementos fundamentais da arquitetura MPLS. Entre a origem e o destino das mensagens, representados por nós IP, tem-se a rede MPLS propriamente dita, compostas por roteadores MPLS, representados na cor cinza. A seguir serão apresentados os principais componentes da tecnologia MPLS.

2.1.1 Roteadores

Os roteadores são equipamentos com funções de roteamento que estabelecem LSPs que podem requerer ou não reserva de recursos da rede. Basicamente existem dois tipos: roteadores de borda de rótulo - *Label Edge Routers (LER)* e roteadores de comutação de rótulos - *Label Switching Routers (LSR)*.

Os LERs ligam diversas sub-redes *(Ethernet, Frame Relay* e *ATM)* à rede MPLS e são responsáveis pela designação e retirada de pilhas inteiras de rótulos dos pacotes no tráfego de entrada e saída na rede MPLS.

Os LSRs possuem alta velocidade e têm a responsabilidade de fazer a troca, retirada e inserção de rótulos do topo da pilha, passando o pacote para o próximo roteador e assim por diante.

2.1.2 Rótulo

Os rótulos são identificadores, de tamanho fixo, no qual o encaminhamento MPLS é baseado. O rótulo MPLS deve ser posicionado depois de qualquer cabeçalho de camada 2 e antes de qualquer cabeçalho de camada 3. Seu tamanho é definido em 4 octetos. O rótulo associa pacotes às respectivas conexões. Um rótulo pode ser pensado como uma forma abreviada para o cabeçalho do pacote, de forma a indicar ao pacote a decisão de remessa que um roteador faria. Seu formato, como indicado na Figura 5, é o seguinte:

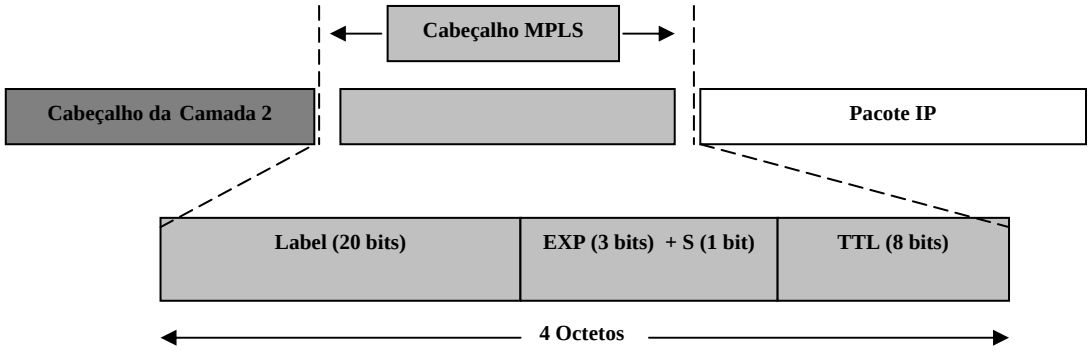


Figura 5 - Rótulo MPLS

1. O campo *Label* (20 bits) contém o valor atual do rótulo MPLS;
2. O campo EXP (3 bits) são reservados para uso experimental;
3. O campo S (*Stack*) (1 bit) identifica a última entrada na pilha do rótulo;
4. O campo tempo de vida *Time to Live (TTL)* (8 bits) fornece funcionalidades de um TTL IP convencional. Especifica um limite de saltos que o pacote pode atravessar antes de ser descartado.

2.1.3 Pilhas de Rótulos

A comutação de rótulos foi projetada para ser usada em *backbones* de redes de grande porte. MPLS suporta essa comutação com operações hierárquicas, baseadas na possibilidade de o pacote possuir mais de um rótulo. O empilhamento de rótulos permite que LSRs troquem informações entre si e ajam como nós de borda para um grande domínio de redes e outros LSRs.

O processamento de um pacote rotulado é completamente independente do nível de hierarquia, ou seja, o nível do rótulo é irrelevante para o LSR. O processamento é sempre baseado no rótulo do topo, sem levar em conta os outros rótulos que pode haver abaixo deste.

2.1.4 Classe de equivalência de encaminhamento (FEC)

Uma classe de equivalência de encaminhamento – *Forwarding Equivalency Class (FEC)* consiste em um conjunto de pacotes que serão encaminhados da

mesma maneira em uma rede MPLS. Pacotes de um mesmo fluxo de dados geralmente pertencem à mesma FEC. Requisitos de qualidade de serviço - *Quality of Service (QoS)* também podem ser definidos através da atribuição de FECs. A FEC é representada por um rótulo, e cada LSP é associado a uma FEC. Ao receber um pacote, o LER verifica a qual FEC ele pertence e o encaminha através do LSP correspondente. Portanto, há uma associação pacote - rótulo FEC – LSP. Essa associação acontece apenas uma vez, quando o pacote entra na rede MPLS. Esta funcionalidade, proporciona grande escalabilidade do ponto de vista de fluxos de tráfego e flexibilidade na introdução de novos serviços devido à possibilidade de atribuição de FECs baseado em requisitos de QoS.

2.1.5 Protocolo de distribuição de rótulos (LDP)

O *Label Distribution Protocol (LDP),* padronizado na RFC 3036, é um conjunto de procedimentos através do qual um LSR informa aos outros LSRs a respeito do significado dos rótulos usados para encaminhar o tráfego entre a através deles. Esse conjunto de procedimentos e mensagens oferece meios para os LSRs estabelecerem LSPs através de uma rede. O LDP oferece uma associação de FEC a cada LSP que cria. Essa associação especifica quais pacotes são mapeados para qual LSP. Dois LSRs que se comunicam para a troca de informações de rótulos são chamados “pares LDP” (LDP *Peers*), e diz-se haver uma sessão LDP entre eles.

Há quatro categorias de mensagens LDP:

1) Mensagens de descoberta: são empregadas inicialmente para anunciar a presença de um LSR em uma rede e posteriormente para ratificar que o LSR continua presente.
2) Mensagens de sessão: são usadas para estabelecer, manter e terminar sessões LDP entre dois LSRs.
3) Mensagens de anúncio de rótulo: são usadas para criar, modificar e suprimir associações de rótulo FEC.
4) Mensagens de notificação: são usadas para prover informações de estado da rede e sinalizar erros.

A arquitetura MPLS não define um único método de distribuição de rótulos. Alguns protocolos foram estendidos como protocolo de gateway de borda - *Border Gateway Protocol (BGP)* e protocolo de reserva de largura de banda - *Resource Reservation Protocol (RSVP)* (AWDUCHE, 2002), enquanto novos foram definidos como o roteamento restrito de LDP - *Constrained Based Routing (CR-LDP)* (JAMOUSSI, 2002).

2.1.6 Protocolo de roteamento restrito de LDP (CR–LDP)

O *Constrained Based Routing - LDP (CR-LDP LDP)* é uma extensão do protocolo LDP para roteamento baseado em restrições (JAMOUSSI, 2002) e adiciona novas funções ao LDP, tais como:

1) Roteamento explícito: permite a indicação explícita da rota a ser seguida.
2) Estabelecimento de LSP com qualidade de serviço: permite reservar recursos para um dado LSP.
3) Preempção entre LSPs: permite atribuir prioridades aos LSPs. O estabelecimento de uma LSP com prioridade maior que outra já estabelecida no mesmo caminho, causa a extinção da LSP já existente (PORTNOI, 2005).

2.1.7 Estruturas de dados

Para que o encaminhamento MPLS ocorra corretamente, são necessárias algumas estruturas de dados para auxiliar na interpretação de rótulos e no seu processamento. Em geral, existem três estruturas de dados como mostra a Figura 6. Uma estrutura para ajudar a inserir rótulos de saída (rótulos em pacotes que estão saindo do LSR ou LER de ingresso), chamada de registro de encaminhamento por rótulos para o próximo passo - *Next Hop Label Forwarding Entry (NHLFE*), uma segunda estrutura para ajudar a interpretar rótulos de entrada (rótulo de pacotes que estão entrando no LSR ou LER de egresso), chamada de mapa de rótulos de entrada - *Incomming Label Map (ILM)* e uma terceira estrutura utilizada por LER`s de

ingresso para descobrir que rótulo adicionar a um pacote proveniente do encaminhamento de nível três, chamada de FTN (FEC To NHLFE).

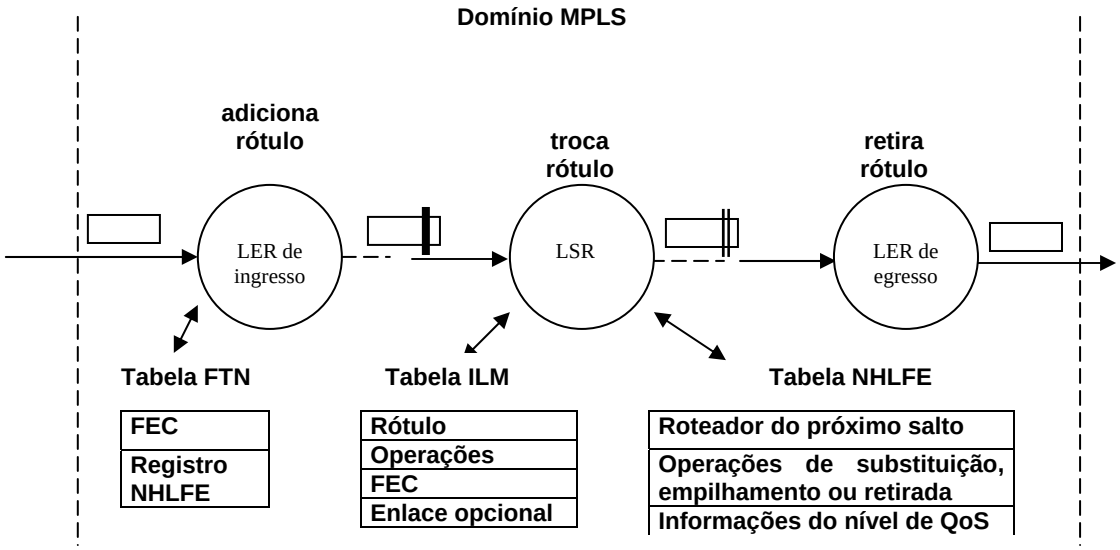


Figura 6 – Tipos de tabelas utilizadas em um domínio MPLS

A tabela NHLFE consiste de todos os rótulos que podem ser inseridos em pacotes pelo roteador. O conteúdo da tabela NHLFE é:

a) O roteador do próximo salto do pacote.

b) Operação a ser realizada na pilha de rótulos do pacote (substituição, empilhamento ou retirada).

c) Informações do nível de QoS associadas ao pacote.

A tabela ILM é implementada somente nos roteadores de núcleo, pois trabalha apenas com pacotes já rotulados. O conteúdo de cada entrada ILM é: rótulo, operações a serem realizadas no pacote, FEC e um enlace opcional para uma estrutura de rótulo de saída. A ordem de operações efetuadas em pacotes rotulados que chegam ao roteador é:

1) Extração do rótulo do topo da pilha.

2) Busca do rótulo na tabela ILM.

3) Processamento adicional do pacote baseado nas operações descritas na ILM.

A tabela FTN tem como função o mapeamento de cada FEC em um conjunto de NHLFEs. Um registro FTN consiste em FEC e registro NHLFE. A ordem de operações da FTN é:

1) Decidir a qual FEC um pacote pertence.

2) Encontrar a FEC na tabela FTN.

3) Encaminhar o pacote para o registro NHLFE obtido na FTN.

2.2 ROTEAMENTO DE LSPs

Roteamento de LSPs é a seleção da rota durante o estabelecimento de um LSP. A arquitetura MPLS prevê duas opções para o estabelecimento de rotas: Roteamento salto-por-salto e Roteamento explícito. No roteamento salto-por-salto, cada nó determina de forma independente o próximo salto para cada FEC, o que corresponde à forma tradicional do roteamento IP.

No caso do LSP roteado explicitamente, os LSRs não são autônomos na escolha do próximo salto, ou seja, um único LSR, normalmente o LSR de ingresso ou o LSR de egresso, especifica todas (ou parte) dos LSRs que compõem a rota do LSP. As rotas explícitas não precisam ser calculadas por um operador da rede. Existem diversos algoritmos que os roteadores podem usar para calcular rotas explícitas automaticamente. O mais comum deles é chamado caminho restrito mais curto primeiro – *Constrained Shortest Path First (CSPF),* que é semelhante aos algoritmos por estado de enlace, mas também leva em consideração as restrições (PETERSON; BRUCE, 2004).

O roteamento explícito é aplicado na engenharia de tráfego e também pode ajudar a tornar as redes mais tolerantes em caso de falhas, usando uma capacidade chamada re-roteamento rápido.

2.3 ENGENHARIA DE TRÁFEGO

Ao lidar com o crescimento e a expansão da rede, existem dois tipos de engenharia: Engenharia de Rede e Engenharia de Tráfego. Engenharia de Rede é a

manipulação da rede para se ajustar ao seu tráfego. Enquanto engenharia de tráfego é a manipulação do tráfego para se ajustar à rede (OSBORNE; AJAY, 2002).

O maior problema de qualidade de serviço na rede está associado à questão do congestionamento. Este ocorre devido a dois fatores: Carência de recursos e má distribuição de recursos. Quando o problema do congestionamento está relacionado à carência de recursos, deve-se fazer a atualização da infra-estrutura da rede de modo a introduzir recursos adicionais. Quando o congestionamento ocorre em função de uma má distribuição da utilização dos recursos, tem-se um problema de engenharia de tráfego (MAGALHÃES; CARDOSO, 2001).

A engenharia de tráfego é o aspecto das redes que está ligado à otimização de desempenho. Segundo Awduche e Bijan, (2002), a engenharia de tráfego engloba tecnologia e aplicação de princípios científicos para medir, modelar, caracterizar e controlar o tráfego na rede, de forma a facilitar a operacionalidade e resultar em uma utilização balanceada dos recursos da rede e minimização da ocorrência de congestionamentos.

Através da engenharia de tráfego, podem-se alcançar benefícios como a capacidade de evitar pontos de congestionamento, roteamento rápido dos fluxos em caso de falhas, uso mais eficiente da banda disponível e possibilidade de oferecimento de melhores garantias de Qualidade de serviços.

Uma das tarefas da engenharia de tráfego é movimentar o tráfego de um enlace congestionado para a capacidade não usada de outro enlace. Como resultado direto da ação da engenharia de tráfego, é possível mover o tráfego para caminhos diferentes do caminho mais curto determinado pelo protocolo de gateway interno - *Internal Gateway Protocol (IGP).* Consequentemente é viável equilibrar o tráfego de modo a ocupar os vários enlaces e os elementos de rede (comutadores e roteadores) de modo que nenhum destes componentes encontrem-se super ou sub-utilizados. A engenharia de tráfego pode ser classificada como uma atividade complementar à infra-estrutura de roteamento de modo a oferecer informações adicionais para utilização no encaminhamento do tráfego, através de caminhos alternativos (MAGALHÃES; CARDOSO, 2001).

2.3.1 Engenharia de Tráfego em Redes IP

A engenharia de tráfego não é uma aplicação específica do MPLS, pois, na prática, pode ser, implementada também nas tecnologias IP e ATM. Os princípios e arquiteturas da engenharia de tráfego em Redes IP, assim como, as metodologias para avaliação e otimização de desempenho operacional das redes IP, são discutidos na RFC 3272 (AWDUCHE, 2002).

Com a engenharia de tráfego na rede IP, não existe uma maneira razoável de controlar o caminho do tráfego com base na origem do tráfego, mas apenas com base no destino do tráfego. A principal maneira de se controlar o caminho que o pacote IP segue na rede é mudar o custo em um enlace específico. Ao fazer o caminho mais longo custar menos do que o caminho mais curto, todo o tráfego passará pelo caminho mais longo. Isto não resolve o problema, simplesmente muda de lugar. Outra alternativa é mudar os custos de enlace de modo que o caminho curto e o caminho longo tenham o mesmo custo. Esta solução só funciona em redes pequenas. Em redes grandes, tentar configurar os custos de enlace de modo que todos os caminhos sejam usados, é extremamente difícil, quando não impossível.

2.3.2 Engenharia de Tráfego em Redes ATM

A engenharia de tráfego, em redes ATM, permite incluir circuitos virtuais - *Virtual Circuits (VC)* a partir de uma fonte para um destino, podendo, assim, obter mais controle sobre o fluxo de tráfego na rede. Isto é possível montando uma malha completa de VCs entre um conjunto de roteadores, redimensionando e redirecionando periodicamente esses VCs com base nas demandas do fluxo que atravessa (OSBORNE; AJAY, 2002). Entretanto, quando ocorre mudanças na topologia da rede, aumenta a quantidade de trabalho dos roteadores para manter cada um dos outros roteadores informados sobre essas mudanças. Isto significa que, quando um enlace é interrompido em um ambiente de malha completa, os dois roteadores em cada ponta desse enlace informam a todos os seus vizinhos a respeito do enlace interrompido, e cada um desses vizinhos informa à maioria dos seus vizinhos. E quando um roteador falha, todos os seus vizinhos informam a todos

os outros roteadores aos quais estão conectados, a respeito da falha, e os roteadores que recebem essa informação, retransmitem-na para seus vizinhos (PETERSON; BRUCE, 2004).

2.3.3 Engenharia de Tráfego em Redes MPLS (MPLS TE)

MPLS TE combina as capacidades de engenharia de tráfego disponível originalmente na tecnologia ATM com a flexibilidade e a diferenciação de classe de serviço do IP. MPLS TE permite montar LSPs através da rede, por onde o tráfego será encaminhado.

A utilização da MPLS TE traz várias vantagens, tais como:

1) MPLS permite esquemas sofisticados de roteamento baseados na capacidade de estabelecimento de LSPs, explicitamente roteados.

2) Evita os problemas de inundação, apresentado pelo ATM. Uma vez que diminui o número de adjacências, sem formar uma malha completa de vizinhos de roteamento.

A Figura 7 representa um conjunto de roteadores interconectados por VCs em uma rede ATM, uma configuração que é chamada de overlay (ou sobreposta). Nesta rede, cada roteador é conectado a cada um dos outros roteadores por um VC. Neste caso, mostram-se apenas os circuitos de R1 para todos os seus roteadores emparelhados. R1 possui cinco vizinhos de roteamento, e precisa trocar mensagens de protocolo de roteamento com todos eles. Diz-se que R1 possui cinco adjacências de roteamento. Ao contrário, na Figura 8, os switches ATM foram substituídos por LSRs. Não há mais VCs interconectando os roteadores. Assim, R1 tem apenas uma adjacência, com LSR1.

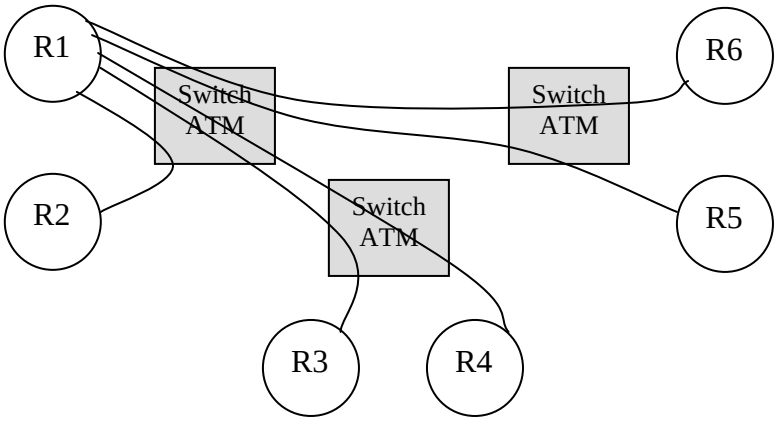


Figura 7 – Roteadores conectados entre si usando um overlay de VCs
Fonte: Peterson e Bruce (2004).

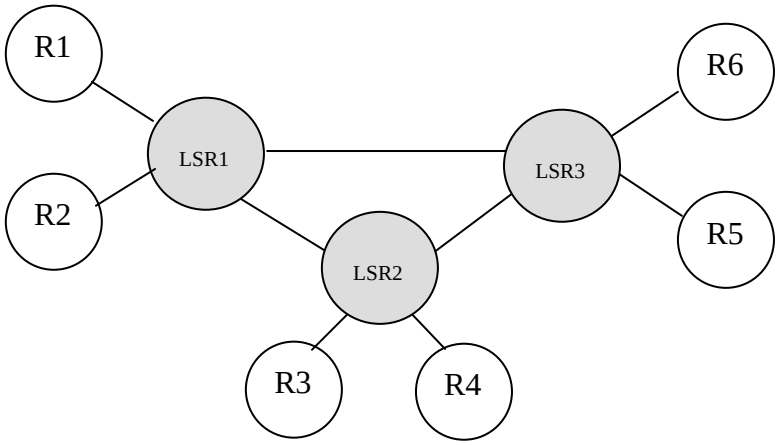


Figura 8 – Roteadores emparelhados diretamente com LSRs
Fonte: Peterson e Bruce (2004).

A RFC 3272 considera a tecnologia MPLS muito poderosa para a engenharia de tráfego na Internet, porque apóia a criação de LSP explícito, que permite o roteamento baseado em restrições, para ser implementado eficazmente em redes IP (AWDUCHE, 2002). As exigências para a engenharia de tráfego em redes MPLS são descritas na RFC 2702 (AWDUCHE, 1999). As extensões do protocolo RSVP, que apóiam a criação de LSP explícito, são discutidos na RFC 3209 (AWDUCHE, 2001). E as extensões do protocolo LDP, conhecido como CR-LDP, que apóiam a criação de LSP explícito, são apresentadas na RFC 3212 (JAMOUSSI, 2002).

3 PROTEÇÃO E RESTAURAÇÃO DE FALHAS

Neste capítulo, é apresentado o conceito geral de defeito, erro e falha, o re-roteamento com IGP, além dos métodos de proteção, focando suas vantagens e desvantagens.

3.1 FALHAS, ERROS E DEFEITOS

Um defeito é definido como um desvio da especificação. Defeitos podem ser tolerados, mas devem ser evitados. Define-se que um sistema está em estado errôneo ou em erro se o processamento posterior a partir desse estado pode levar a defeito. Finalmente define-se falha (ou falta) como a causa física ou algorítmica do erro (WEBER, 2004).

A Figura 9 mostra uma simplificação para os conceitos de falha, erro e defeito. Falhas estão associadas ao universo físico, erros ao universo da informação e defeitos ao universo do usuário. Assim um chip de memória, que apresenta uma falha do tipo grudado-em-zero (*stuck-at-zero*) em um de seus bits (falha no universo físico), pode provocar uma interpretação errada da informação armazenada em uma estrutura de dados (erro no universo da informação) e, como resultado, o sistema pode negar autorização de embarque para todos os passageiros de um vôo (defeito no universo do usuário). É interessante observar que uma falha não necessariamente leva a um erro (aquela porção da memória pode nunca ser usada) e um erro não necessariamente conduz a um defeito (no exemplo, a informação de vôo lotado poderia eventualmente ser obtida a partir de outros dados redundantes da estrutura) (WEBER, 2004).

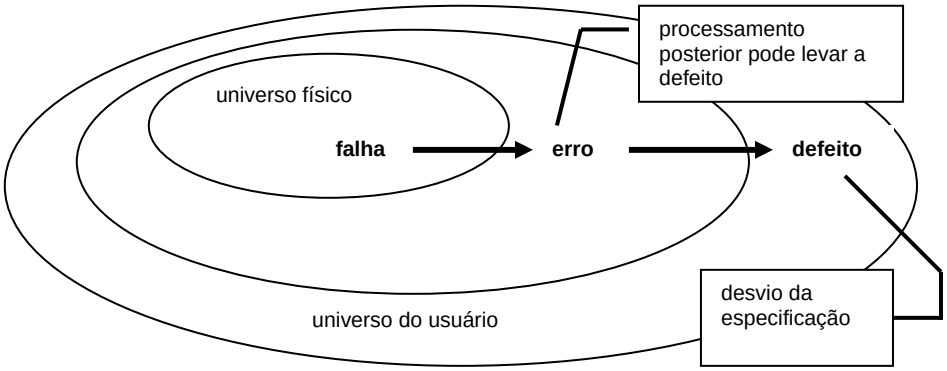


Figura 9 – Modelo de 3 universos: falha, erro e defeito
Weber (2004).

As principais causas de falhas são problemas de especificação, problemas de implementação, componentes defeituosos, imperfeições de manufatura, fadiga dos componentes físicos além de distúrbios externos como radiação, interferência eletromagnética, variações ambientais (temperatura, pressão, umidade) e também problemas de operação.

Nas redes de computadores existem dois tipos de falhas: falha de enlace e falha de nó (OSBORNE; AJAY, 2002). É muito importante que se reduza os efeitos negativos de tais falhas, como a perda de pacotes. A MPLS TE com sua capacidade de direcionar o tráfego para fora do caminho mais curto, derivado do protocolo de gateway interno - *Internal Gateway Protocol (IGP),* ajuda a aliviar a perda de pacotes associada a falhas de enlace ou nó na rede. Esta capacidade é conhecida como Re-roteamento Rápido ou Proteção.

Segundo Osborne e Ajay (2002, p.45),

> A proteção, no contexto da recuperação rápida, é ter procedimentos preparados que, quando aplicados a recursos selecionados, garantem a perda mínima de tráfego na falha. Os recursos protegidos podem ser vistos como recursos físicos (enlace ou nó) ou recursos lógicos (os LSPs que atravessam um enlace ou nó).

Um aspecto crucial no desenvolvimento de um sistema de recuperação de falhas é a criação e roteamento dos caminhos de recuperação. Isto pode ser alcançado tanto estática quanto dinamicamente. No caso estático, conhecido como Proteção, é determinado um desvio em torno de uma possível falha no momento do estabelecimento da conexão ou projeto da rede, ou seja, antes da falha. No caso dinâmico, conhecido como Restauração, o desvio é determinado dinamicamente

após a ocorrência da falha. Em conseqüência, esquemas de proteção podem, em geral, se recuperar de uma falha mais rapidamente (desde que o desvio não seja afetado por quaisquer outras falhas), mas são menos eficientes em otimização de recursos adicionais do que os métodos de restauração. Por outro lado, métodos de restauração podem resistir a uma ou múltiplas falhas (desde que o destino ainda seja alcançável, com conectividade e largura de banda suficiente), mas não podem garantir o tempo de recuperação e/ou a quantidade de informações perdidas.

3.2 RE-ROTEAMENTO COM IGP

O protocolo de gateway interno (IGP) é tradicionalmente utilizado para o re-roteamento em caso de falhas na rede IP. Entretanto, possui algumas deficiências (OSBORNE; AJAY, 2002):

a) O IGP em uma rede grande pode levar muito tempo (5 a 10 segundos) para convergir.

b) Uma falha de enlace pode levar ao congestionamento de algumas partes da rede, enquanto deixa outras partes livres de congestionamento.

c) A configuração do IGP para convergir rapidamente pode torná-lo extremamente sensível à pequena perda de pacote, causando alarmes falsos e convergência do IGP sem qualquer motivo.

3.3 MÉTODOS DE PROTEÇÃO

Os métodos de proteção iniciam com a identificação da falha e terminam com a recuperação do enlace ou nó. Esta cadeia de eventos compreende três componentes:

1) Configuração antes da falha: É necessário um método para selecionar os caminhos de trabalho e recuperação, e um método para configurar o caminho de trabalho para usar o caminho de recuperação no caso de falha.

2) Detecção da falha: Método para detecção e notificação de falhas, traz informações sobre a ocorrência de falha para a entidade da rede

responsável por tomar a ação corretiva apropriada, por exemplo, transmitir um sinal de indicação de falha - *Fault Indication Signal (FIS).*Restauração da conectividade: É necessário um método de transição para mover o tráfego do caminho de trabalho para o caminho de recuperação.

De modo a fornecer certos aspectos de proteção, dois novos graus de nós são necessários: um nó responsável pela função de transição, uma vez que a falha seja identificada; e um nó onde o caminho de trabalho e o caminho de recuperação confluem. Em MPLS, estes dois nós são definidos como *Protection Ingress LSR (PIL)* e *Protection Merging LSR (PML)*, respectivamente.

Dependendo de onde o desvio se origina, os esquemas de proteção podem ser classificados em proteção de enlace, caminho ou segmento. Na proteção de enlace, para cada enlace carregando tráfego em situação normal, chamado enlace de trabalho, um segmento de recuperação é usado como desvio de uma extremidade do enlace para a outra extremidade. Na proteção de caminho, para cada caminho de trabalho da fonte até o destino, um caminho de recuperação é usado como desvio. E, na proteção de segmento, para cada segmento de trabalho, um segmento de recuperação é usado como desvio.

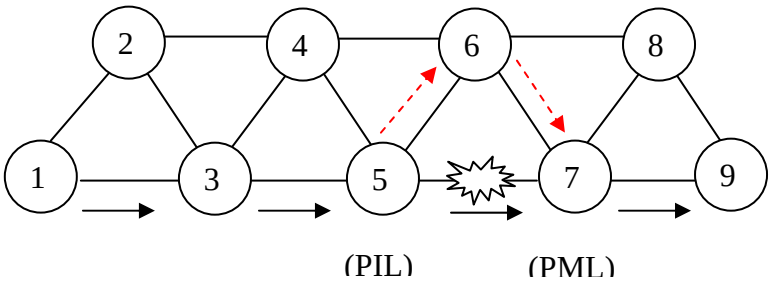
3.3.1 Método de Proteção de Caminho Global (MPCG)

Neste modelo (ver Figura 2), o nó de ingresso é responsável pelo caminho de restauração quando o FIS chega. Isto requer um caminho de recuperação não conectado alternativo para cada caminho de trabalho. O nó de ingresso é onde o processo de proteção é iniciado, independentemente da localização da falha ao longo do caminho de trabalho.

A vantagem deste método é que estabelece apenas um caminho de recuperação por caminho de trabalho. Além disso, é um método de proteção centralizado, o que significa que para cada possível origem de LSPs apenas um *Label Switch Router (LSR)* tem que ser fornecido com funções PIL. Por outro lado, este método tem um custo mais alto (em termos de tempo), dado que o FIS é enviado para o nó de ingresso. Isto é, implica em perdas maiores de pacote durante o tempo de re-roteamento.

3.3.2 Método de Proteção de Caminho Local

Com este método, a recuperação começa no ponto mais próximo da falha (Figura 10). É um método local e não envolve necessariamente o nó de ingresso. A principal vantagem é que oferece um tempo de recuperação mais rápido do que o modelo de reparo global, bem como uma redução significativa na perda de pacotes. Por outro lado, cada nó que requer proteção precisa ser provido com uma função de transição PIL. E um PML também precisa ser fornecido para cada LSP de trabalho. Outra desvantagem é a manutenção e criação de múltiplas recuperações (uma por domínio protegido). Isto pode levar à baixa utilização de recursos e aumento da complexidade. Uma solução intermediária estabelece proteções locais apenas para segmentos com altos requisitos de confiabilidade.



. ⟶ Caminho de Trabalho (1-3-5-7-9)

. ⇢ Caminho de Recuperação (5-6-7)

Figura 10 - Proteção de Caminho Local

3.3.3 Método de Proteção de Caminho Reverso (MPCR)

A principal característica da proteção MPCR é o fato de inverter o tráfego próximo ao ponto da falha de volta ao nó fonte (nó de ingresso) do caminho que está sendo protegido, via um LSP de recuperação reversa (ver Figura 3). Assim que uma falha é detectada, o LSR no início do enlace com falha re-roteia o tráfego para o LSP de recuperação na direção oposta, de volta para o nó de ingresso. Haskin e Krishnan (2000) propuseram para pré-estabelecer o caminho de recuperação reverso, usar o mesmo nó do caminho de trabalho, simplificando assim o processo de sinalização.

Este método, assim como o de reparo local, é especialmente indicado contra as perdas de tráfego. Outra vantagem é a indicação de falha simplificada, uma vez

que a recuperação reversa transmite o FIS para o nó de ingresso e o caminho de recuperação ao mesmo tempo. Uma das desvantagens é a alta utilização de recursos.

3.3.4 Método de Proteção de Múltiplos Segmentos (MPMS)

Na proteção de múltiplos segmentos (ver Figura 4), um caminho de trabalho é dividido em diversos segmentos de trabalho e cada segmento de trabalho é, então, protegido com um segmento de recuperação (em vez de proteger o caminho de trabalho como um todo, como nos esquemas de proteção de caminho). Os esquemas de proteção baseados em segmentos são também similares às proteções de caminho local (ou proteção de enlace), mas o tráfego re-roteado necessita passar apenas pelo nó que inicia o segmento de recuperação, em oposição à fonte como na proteção de caminho (ou o nó de início do enlace que falhou, como na proteção de enlace (DAHAI XU, 2003).

A idéia básica do esquema de proteção baseado em múltiplos segmentos é prover proteção para um segmento de trabalho ao tempo em que usa um desvio, chamado segmento de recuperação. Esse segmento de recuperação inicia e terminamos mesmos dois nós do segmento de trabalho.

No esquema proposto em (DAHAI XU, 2003), um conjunto válido de segmentos de trabalho para um determinado caminho de trabalho está sujeito a três restrições:

1) Cada enlace ao longo do caminho de trabalho pertence, no mínimo, a um segmento de trabalho e, no máximo, a dois;
2) Um segmento de trabalho não pode ser um subconjunto de outro segmento de trabalho;
3) Adicionalmente cada enlace pertencente a dois segmentos de trabalho sobrepostos precisa ser protegido apenas pelo segmento de recuperação correspondente ao segundo (posterior) segmento de trabalho.

As Figuras 11 e 12, mostram dois exemplos de divisão de um caminho de trabalho formado por sete enlaces, em segmentos de trabalho, conforme as restrições do esquema proposto em (DAHAI XU, 2003).

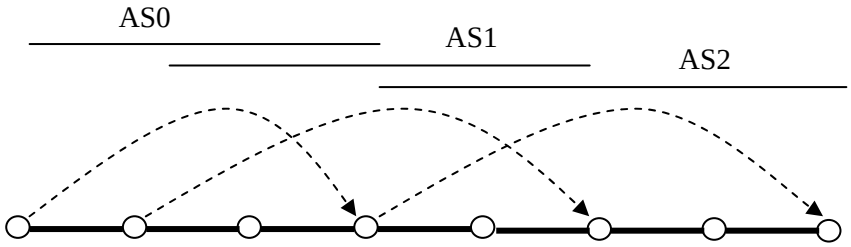


Figura 11 – Divisão do caminho de trabalho em segmentos de trabalho

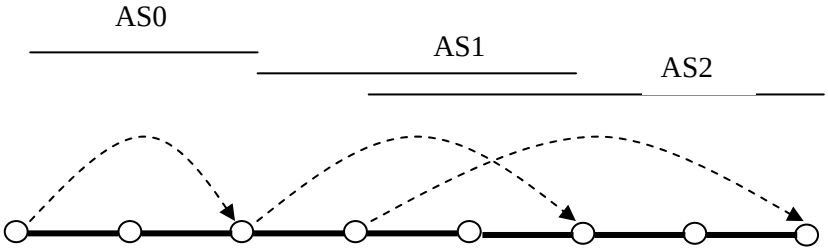


Figura 12 – Divisão do caminho de trabalho em segmentos de trabalho (exemplo 02)

A Proteção MPMS, devido à sua flexibilidade inerente em fracionar um caminho de trabalho e protegê-lo com múltiplos segmentos de recuperação, apresenta vantagens sobre as proteções de caminho compartilhado em dois aspectos: a eficiência de otimização de largura de banda e o tempo de recuperação.

Em relação ao primeiro aspecto, a grande flexibilidade oferecida em (DAHAI XU, 2003) pode levar a graus mais elevados de compartilhamento de largura de banda de recuperação, não apenas entre segmentos de recuperação para diferentes caminhos de trabalho (chamado intercompartilhamento), como mostra a Figura 13 onde dois segmentos de recuperação BS1,1 e BS2,1 podem compartilhar largura de banda de recuperação, mas também entre segmentos de recuperação para o mesmo caminho de trabalho (chamado intracompartilhamento), como mostra a Figura 14. Onde BS1 e BS2 partilham largura de banda de recuperação no enlace c.

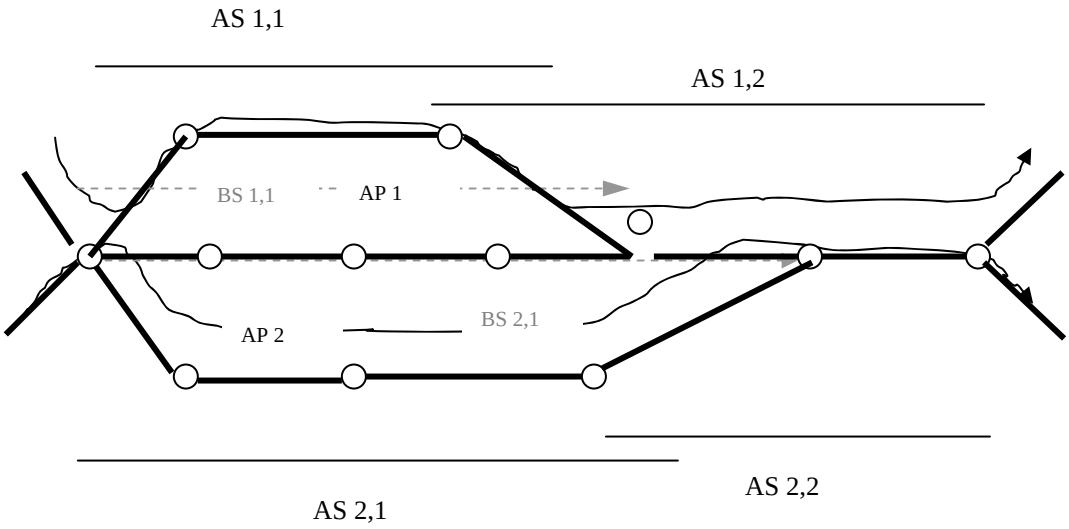


Figura 13 – Intercompartilhamento

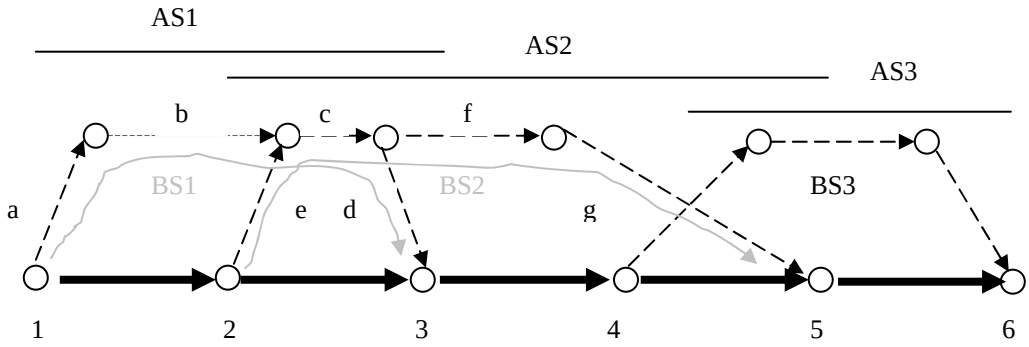


Figura 14 - Intracompartilhamento

Em relação ao tempo de recuperação, a Proteção MPMS tem melhores garantias de qualidade de serviço - *Quality of Service (QoS)*. Isso ocorre porque protege cada segmento de trabalho, usando um segmento de recuperação mais curto em vez de proteger o caminho de trabalho inteiro, usando um caminho de recuperação mais longo (como na Proteção MPCR).

Convém destacar que o método proposto em (DAHAI XU, 2003) tem outros benefícios, como, por exemplo, pode tolerar mais falhas múltiplas do que a Proteção MPCR, com igual ou menor consumo de largura de banda. Cumpre salientar que estes benefícios adicionais requerem controles de sinalização mais complexos, comparados aos métodos de proteção de caminho.

4 SIMULAÇÃO DA PROTEÇÃO DE FALHAS NO NS

O simulador de rede *Network Simulator (NS)* é uma ferramenta que tem o propósito de executar modelagens e simulações de situações reais, focadas para o desenvolvimento de pesquisas em redes de computadores (VINT, 2004). Foi desenvolvida na Universidade *Berkeley* da Califórnia, nas linguagens C++ e Otcl e tem como vantagens o fato de ter seu código aberto e bibliotecas que o auxiliam na simulação de diversos tipos de redes.

Para o desenvolvimento desse trabalho, utilizou-se o NS versão 2.1b6, e duas ferramentas complementares: O Network Animator (NAM) e o Trace Graph (MALEK, 2004). Esta versão não é a mais recente, porém foi escolhida porque possui o pacote completo do *Network Simulator MPLS* (MNS_V2).

Basicamente, o funcionamento do NS corresponde à elaboração de um Script feito em Otcl, no qual são inseridos todos os parâmetros necessários à simulação. A partir do script, um relatório ".tr" é gerado pelo NS e as suas ferramentas podem, a partir do próprio script, ou posteriormente, ser executadas para demonstrar a animação ou os gráficos de informações (TAUMATURGO, 2003).

O NS pode ser utilizado para a simulação de diversas tecnologias como, por exemplo, IP, MPLS e Rede sem fio. Para cada uma dessas tecnologias, os pacotes que permitem as suas simulações e implementações, são incorporados ao NS pelos seus usuários.

Como a tecnologia utilizada nesse trabalho diz respeito ao MPLS, o pacote do NS mais adequado para realizar as implementações é o MNS_v2, desenvolvido inicialmente na Coréia por *Gaeil Ahn* do departamento de Engenharia da Computação da Universidade Nacional de *Chungnam*.

O NS foi construído para rodar preferencialmente em plataformas Unix (FreeBSD, SunOS, Solaris e Linux). Pode também rodar em plataforma *Microsoft Windows*. Este capítulo descreverá características do simulador em sistema operacional Linux (distribuição RedHat 8.0) e arquitetura Intel PC.

Fornecido em arquivos fontes, o NS é compilado durante o processo da instalação. Assim, faz-se necessário um compilador C++ no computador onde será instalado (VINT, 2004). Como o simulador é compilado na instalação, ele pode, à princípio, ser executado em qualquer arquitetura de computador, como Intel/AMD ou RISC.

O pacote do simulador é composto dos seguintes módulos básicos:

a) Tcl/Tk: Interpretador de linguagem Tcl, que é a interface do simulador com o usuário.

b) Otcl: suplemento de orientação a objetos para o Tcl.

c) Tclcl: implementação de classes para Tcl.

d) ns: classes do simulador propriamente dito.

e) nam-1: visualizador e animador gráfico de topologias de rede e simulação.

f) xgraph: ferramenta de plotagem de gráficos.

g) cweb e SGB: bibliotecas requeridas para sgb2-ns e gt-itm.

h) Gt-itm, gt-itm e sgb2-ns: gerador de topologias.

i) zlib: ferramenta para compressão de arquivos.

Há duas formas de instalar o NS: através do ns-allinone, que reúne todos os pacotes acima descritos, ou obtendo cada pacote e fazendo sua instalação separadamente. Nem todos os módulos acima são requeridos para que o simulador funcione, portanto pode-se preparar ambientes com instalações mínimas e outros com mais ferramentas. Outros módulos externos (como Perl) são necessários para realizar certas funções. Detalhes sobre o processo de instalação do NS, são fornecidos em Coutinho (2003).

4.1 ESTRUTURA DO NS

A interface entre o usuário e o NS se dá através da linguagem script Otcl. Segundo os seus desenvolvedores, a divisão em duas linguagens (Otcl e C++) dá

ao simulador velocidade e facilidade de mudança de parâmetros. O núcleo do simulador é escrito em C++, conferindo velocidade, mas esta linguagem torna-se lenta para manipulação constante ou mudança de parâmetros. Otcl, por ser interpretada, é bem mais lenta, porém pode ser facilmente alterada. A Figura 15 mostra a construção geral do NS. Um usuário comum atua no perímetro “tcl”, escrevendo scripts em Otcl e executando simulações. Os escalonadores de eventos e os componentes de rede são implementados em C++. Todo o conjunto constitui-se no NS, que é um interpretador de Otcl com bibliotecas de simulação para redes de computadores.

Figura 15 - Arquitetura do NS
Fonte: VINT (2004).

4.2 OPERAÇÃO BÁSICA DO NS

Para montar uma simulação no NS, é preciso escrever um script em Otcl. Este script contém as seguintes partes básicas:

1) Criação do objeto simulador;
2) Abertura de arquivos de dados de saída;
3) Criação da topologia de rede:
    a) Criação de nós ou nodos;
    b) Conexão dos nós entre si (enlace);
    c) Criação das filas de saídas;
4) Criação dos agentes da camada de transporte e conexão com nós;
5) Criação dos geradores de tráfego (nível de Aplicação) e conexão com agentes de 4ª camada (nível de Transporte);

6) Programação dos escalonadores e temporizadores;

7) Encerramento da simulação, animação e estatísticas.

O processo de simulação pode ser assim resumido:

1) Confecção do script (arquivo texto);
2) Execução do script com o comando “ns nomedoscript.tcl”;
3) Após conclusão da simulação:
    a) Imprimir estatísticas calculadas no script;
    b) Visualizar os eventos com o NAM;
    c) Analisar resultados através dos arquivos de dados com apoio de ferramentas apropriadas (*Trace Graph* ou awk).

Existem dois tipos básicos de arquivos de dados no NS. O arquivo com as saídas de dados da simulação, que contém todas as informações dos pacotes enviados e recebidos em todos os nós, e o arquivo com os dados de saída para a simulação gráfica no NAM, que contém os dados necessários para a animação.

É importante observar que o NS não fornece estatísticas de simulação de modo automático. Estas devem ser obtidas através de procedimentos matemáticos no script ou pela manipulação de objetos especiais chamados monitores. Pode-se, ainda, usar ferramentas para análise dos arquivos de dados gerados durante a simulação, conforme a Figura 16. Estes arquivos, com formatação específica, registram cada evento gerado pelos escalonadores. As ferramentas para análise dos arquivos de dados devem, então, ser capazes de ler os dados gravados nestes arquivos e efetuar os cálculos desejados. Duas destas ferramentas, muito utilizadas, são a *Trace Graph* e o awk, linguagens projetadas para buscar padrões dentro de um arquivo e efetuar ações programadas. O animador NAM pode também ser usado para analisar visualmente a simulação e obter algumas estatísticas, mas não é apropriado para análises mais profundas. Se nada for feito, o simulador apenas rodará o script, gerará os arquivos de saída de dados e encerrará, sem nada mostrar ao usuário.

Figura 16 - Esquema de utilização do NS
Fonte: VINT (2004).

4.3 ARQUITETURA MPLS NO NS

O MPLS *Network Simulator* (MNS) é uma extensão do NS que permite a simulação de redes MPLS. Apesar de ter sido originalmente desenvolvido para ambientes Unix da Sun, o MNS se comporta bem no ambiente *Linux RedHat* 8.0. O MNS foi criado para funcionar na versão 2.1b6 do NS e possui as seguintes funções do MPLS:

1) Comutação de rótulos - operações de troca e empilhamento de rótulos, tratamento do campo tempo de vida - *Time to live (TTL)* e *r*etirada do penúltimo salto - *Penultimate Hop Popping (PHP).*

2) Protocolo LDP - tratamento de mensagens LDP, tais como, mapeamento, remoção, liberação e notificação;

3) Protocolo CR-LDP - tratamento de mensagens de requisição e mapeamento.

4) Agregação de fluxo de pacotes.

5) Controle e configuração de LSPs - distribuição e alocação de rótulos e configuração de LSPs pré-definidos pelo usuário.

6) Implementação do roteamento baseado em restrições, que permite estabelecer CR-LSPs usando informações sobre reserva de recursos da rede.

7) Re-roteamento com suporte a esquemas de proteção de rotas pré-negociadas.

A arquitetura MPLS no NS consiste em três classificadores: classificador MPLS, classificador de endereço e classificador de serviço e três tabelas de informações: tabela parcial de encaminhamento - *Partial Forwarding Table (PFT)*, base de informações de rótulo *Label Information Base (LIB)* e base de informações de caminho explícito - *Explicit Route Information Base* (ERB).

As tabelas ILM e FTN do MPLS, descritas no Capítulo 2, respectivamente, são implementadas nas seguintes tabelas do NS: LIB, ERB e PFT:

1) Na tabela LIB, são mapeados os conjuntos de rótulos e interfaces de entrada e rótulos e interfaces de saída dos nós utilizados pelos fluxos MPLS.

2) Na tabela ERB, são realizados os mapeamentos das FECs e dos LSPs.

3) A tabela PFT é de suma importância, e funciona como um apêndice da ERB. Nela é realizado o mapeamento da FEC e do fluxo de dados com a tabela LIB.

A tabela NHLFE, definida no Capítulo 2, não possui uma estrutura correspondente específica no MNS. Uma parte dela está implementada ao longo do código, no procedimento get-cr-mapping (no arquivo ns-mpls-ldpagent.tcl), enquanto que a informação de próximo passo está armazenada na tabela LIB.

As Figuras 17 e 18 ilustram as relações e interações entre os classificadores e tabelas de informações.

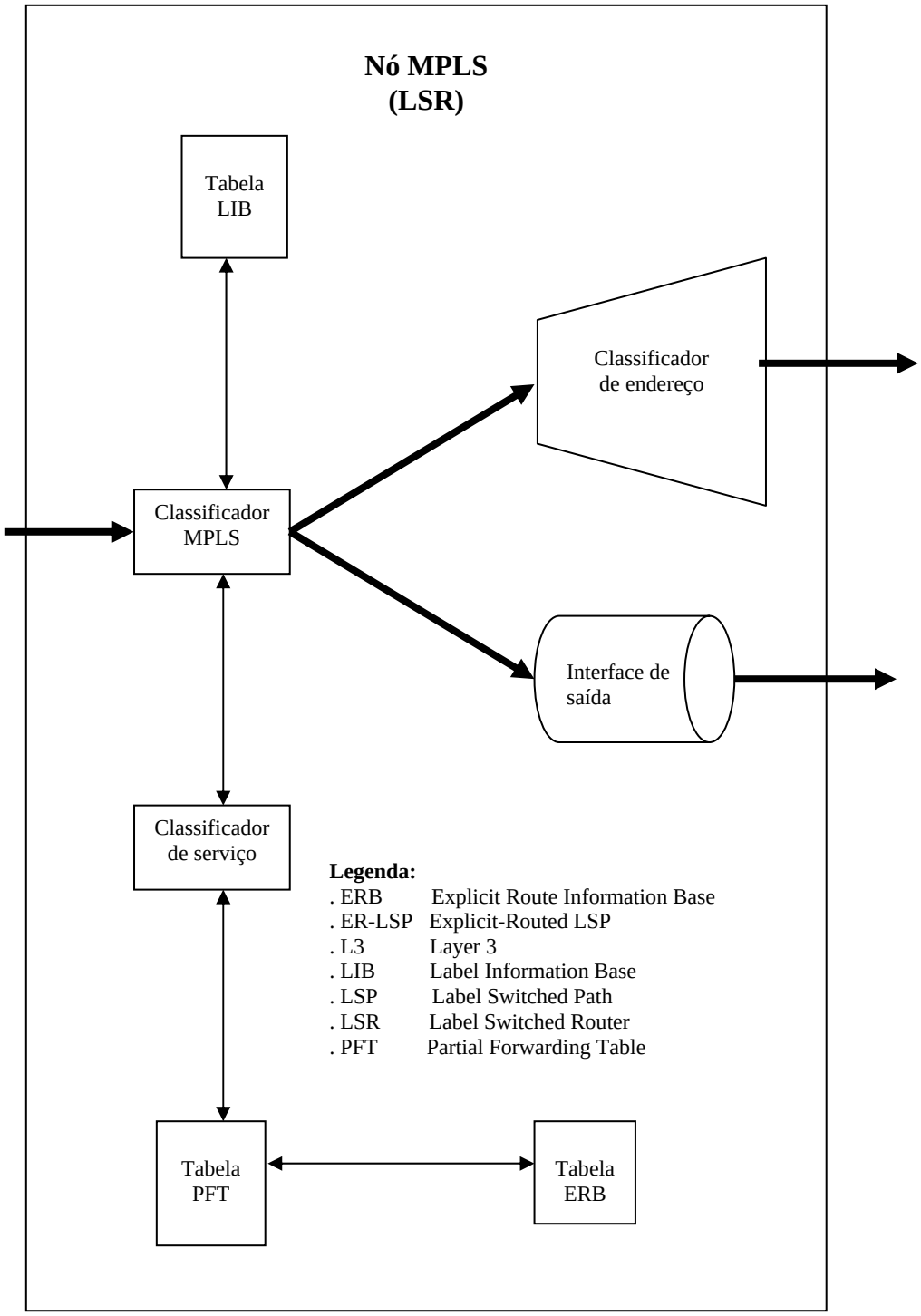


Figura 17 - Nó MPLS no NS
Fonte: Danny (2002).

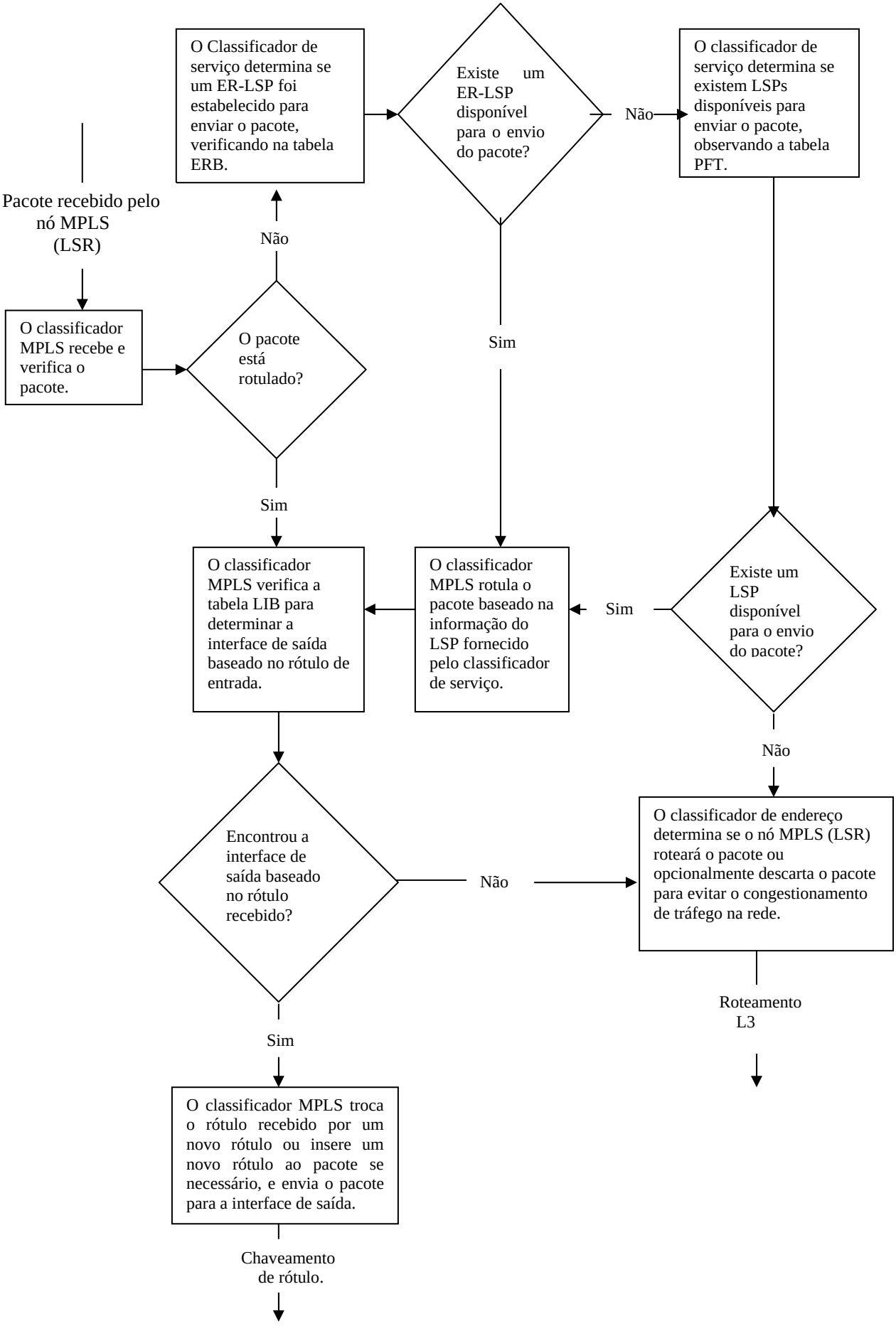


Figura 18 - Operação gráfica do nó MPLS no NS
Fonte: Danny (2002).

Observando a Figura 18 quando um nó mpls (lsr) recebe um pacote, o lsr executa as seguintes operações:

1) O Classificador MPLS verifica se o pacote recebido está com cabeçalho de rótulo MPLS

2) Se estiver rotulado, o classificador MPLS verifica na tabela LIB o rótulo e a interface de saída do pacote. Executa a troca de rótulo. A tabela LIB possui informações de todos os LSPs estabelecidos.

3) Se não estiver rotulado, mas se um caminho explícito LSP (ER-LSP) foi pré-estabelecido para determinar a rota dos pacotes, o classificador de serviço vai devolver a informação sobre ER-LSP para o classificador MPLS, observando a tabela ERB, e o classificador MPLS rotula o pacote baseado nessas informações. A tabela ERB mantém informações de todos os ER-LSP estabelecidos.

4) Se não estiver rotulado, mas há um LSP cujo caminho pode enviar o pacote, então depois de verificar na tabela PFT, o pacote será rotulado com cabeçalho de rótulo apropriado e será tratado como um pacote rotulado pelo classificador MPLS.

5) Se não, o nó MPLS irá se comportar como um roteador IP e o classificador de endereço, encaminhará o pacote, usando o protocolo de roteamento de nível 3 ou descarta o pacote para evitar o congestionamento de tráfego na rede.

4.4 IMPLEMENTAÇÃO DOS MÉTODOS DE PROTEÇÃO NO NS

Esta seção, explica a implementação realizada para permitir a simulação dos métodos de proteção de caminho global (MPCG), proteção de caminho reverso (MPCR) e proteção de múltiplos segmentos (MPMS) em redes MPLS.

Durante os testes das implementações, realizou-se a medição de perdas de pacotes e atraso e a comparação entre os métodos de proteção. Estas medidas são utilizadas como parâmetros de qualidade de serviços para aplicações de rede. Como exemplo, aplicações de áudio altamente interativas, como o telefone por Internet,

podem tolerar atrasos entre 150 e 400 milissegundos e perdas de pacote entre 1 e 20 por cento (KUROSE; KEITH, 2003).

A partir das simulações, pôde-se visualizar e analisar as operações dos métodos de proteção, durante a ocorrência de falhas na rede.

Este trabalho baseou-se totalmente no pacote MNS_V2, tendo em vista a especificidade da tecnologia MPLS. Para tanto, os scripts de exemplo do Método de Proteção de Caminho Global (MPCG) e do Método de Proteção de Caminho Reverso (MPCR) que acompanham o pacote MNS_V2, foram modificados para trabalhar em conformidade com o Método de Proteção de Múltiplos Segmentos (MPMS). Para análise dos arquivos de dados e medições dos resultados, foi utilizada a ferramenta *Trace Graph*.

Depois de criados os scripts no NS (ver APÊNDICES A e B) e implementadas as mudanças correspondentes ao procedimento *Protection Option*, para a criação dos caminhos de trabalho e recuperação e re-roteamento explícito, para cada uma das opções de proteção, houve a realização dos testes de re-roteamento rápido.

4.4.1 Métricas e parâmetros de avaliação

1) Medidas de desempenho:
    a) Perda de pacotes.
    b) Atraso.
2) Parâmetros do Sistema:
    a) Enlace (velocidade, atraso e tipo de fila).
    b) Topologia (hosts, LSRs e LERs).
    c) Local da falha.
3) Parâmetros de Carga:
    a) Tráfego.
4) Fatores e suas faixas de valores:
    a) Tráfego:
        - Exponencial sobre UDP, tamanho constante dos pacotes gerados 200 bytes, taxa de transmissão 500 kbps;

- Taxa constante de bits (CBR - Constant Bit Rate) equivalente à difusão de vídeo contínuo sobre UDP, tamanho constante dos pacotes gerados 2 kbytes, taxa de transmissão 1 Mbps.

5) Topologias:

a) Topologia 1:

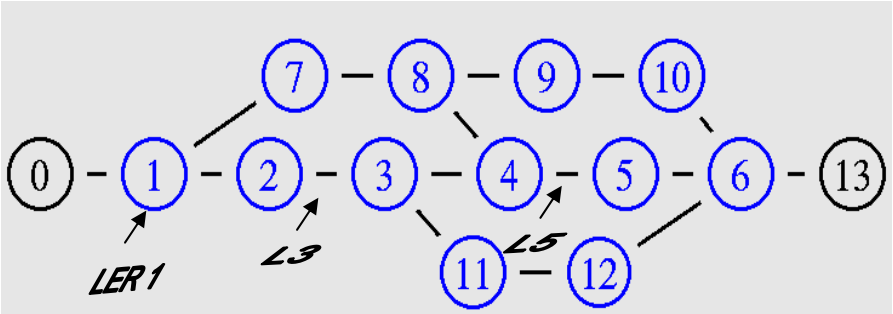


Figura 19 - Topologia 1

- Hosts: 0 e 13;
- LSR's: 2, 3, 4, 5, 7, 8, 9, 10, 11 e 12;
- LER's: 1 e 6;

b) Topologia 2:

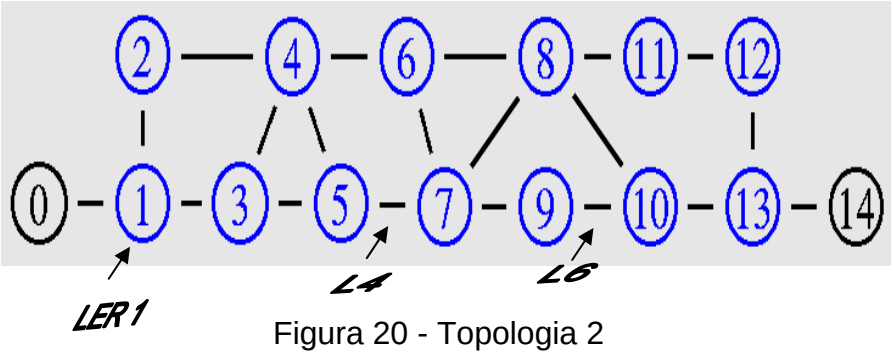


Figura 20 - Topologia 2

- Hosts: 0 e 14;
- LSR's: 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 e 12;
- LER's: 1 e 13;

6) Locais da falha:

- Topologia 1: Enlaces L3 e L5.
- Topologia 2: Enlaces L4 e L6.

4.4.2 Teste do Método de Proteção de Caminho Global (MPCG)

4.4.2.1 Topologia 1, falha no enlace L3

Para esta simulação, estabeleceu-se um LSP de trabalho, seguindo a rota 1_2_3_4_5_6. Após uma falha no enlace L3, o LSR2 transmite um sinal de indicação de falha (FIS) para o LER1, conforme a Figura 21. Em seguida, o tráfego é re-roteado para o LSP de recuperação, seguindo a rota 1_7_8_9_10_6, como mostra a Figura 22.

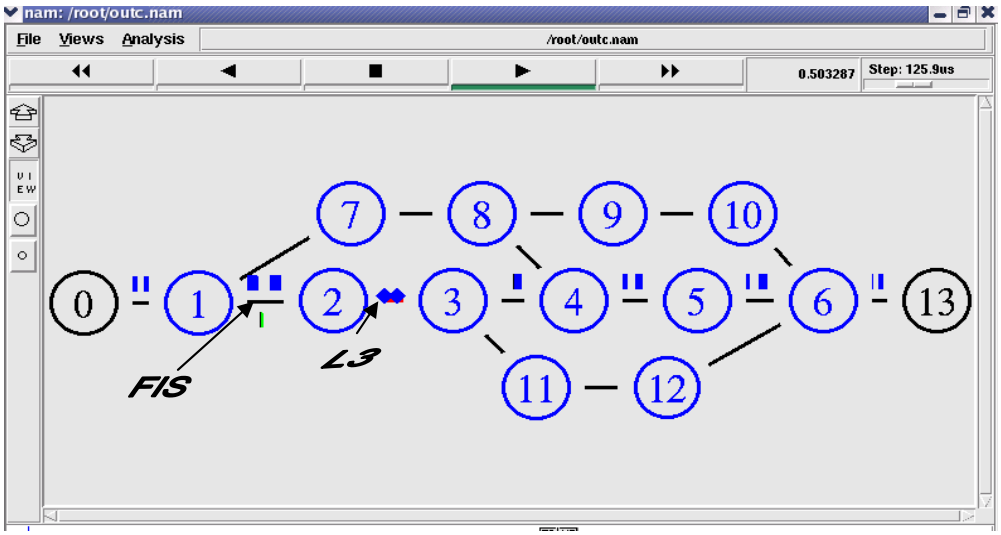


Figura 21 – Sinal de indicação de falha – FIS

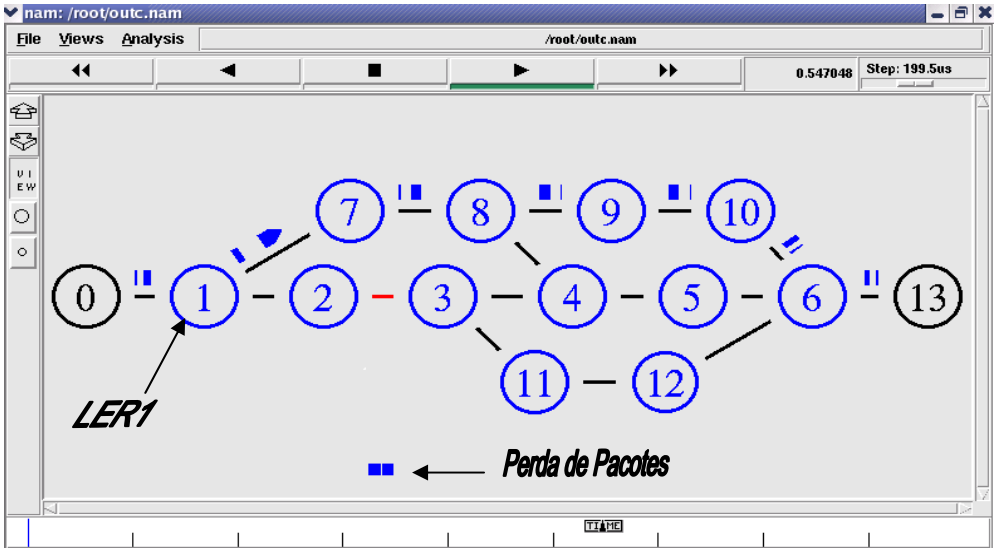


Figura 22 - Proteção de Caminho Global (MPCG), falha no enlace L3

A Figura 23 mostra o trecho de código do procedimento *Protection Option* para seleção dos caminhos de trabalho e recuperação e o re-roteamento explícito de LSP`s. A função *setup-erlsp* é utilizada para criar os LSPs de trabalho e recuperação, a função *bind-flow-erlsp* é responsável pela associação de uma FEC ao LSP e a função *reroute-lsp-binding* é utilizada para re-rotear o tráfego.

```
# Protection Option

 makam       {
                    # The setup of working LSP
                    $ns at 0.0  "$LSR1  setup-erlsp  6   2_3_4_5_6   1000"

                    # The setup of alternative LSP
                    $ns at 0.1  "$LSR1  setup-erlsp  6   7_8_9_10_6  2000"

                    # bind a flow to LSP
                    $ns at 0.3  "$LSR1 bind-flow-erlsp   13   100   1000"

                    # binding working LSP to alternative LSP
                    $ns at 0.3  "$LSR1 reroute-lsp-binding     1000    2000"
                  }
```

Figura 23 – *Protection Option* – Proteção MPCG (topologia 1, falha no enlace L3)

4.4.2.2 Topologia 1, falha no enlace L5

Houve a realização de outro teste em que ocorreu uma falha no enlace L5, como mostra a Figura 24. Nesta situação, o LSR4 assume a função de transmitir o FIS para o LER1, para que em seguida o tráfego seja re-roteado ao LSP de recuperação pré-estabelecido que corresponde à rota 1_7_8_9_10_6.

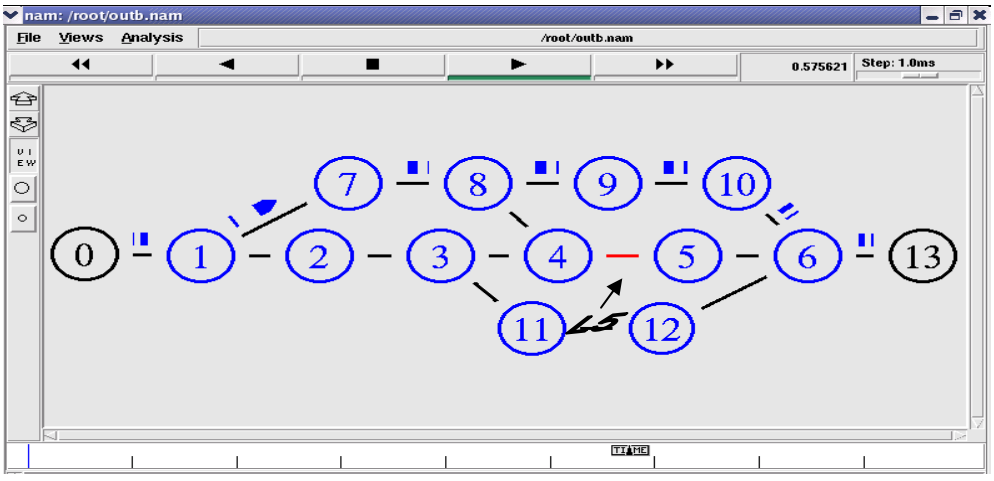


Figura 24 - Proteção (MPCG), falha no enlace L5

4.4.2.3 Topologia 2, falha no enlace L4

Para esta simulação, estabeleceu-se um LSP de trabalho, conforme Figura 25, seguindo a rota 1_3_5_7_9_10_13. Após uma falha no enlace L4, o LSR5 transmite um sinal de indicação de falha (FIS) para o LER1. Em seguida, o tráfego é re-roteado para o LSP de recuperação, seguindo a rota 1_2_4_6_8_11_12_13.

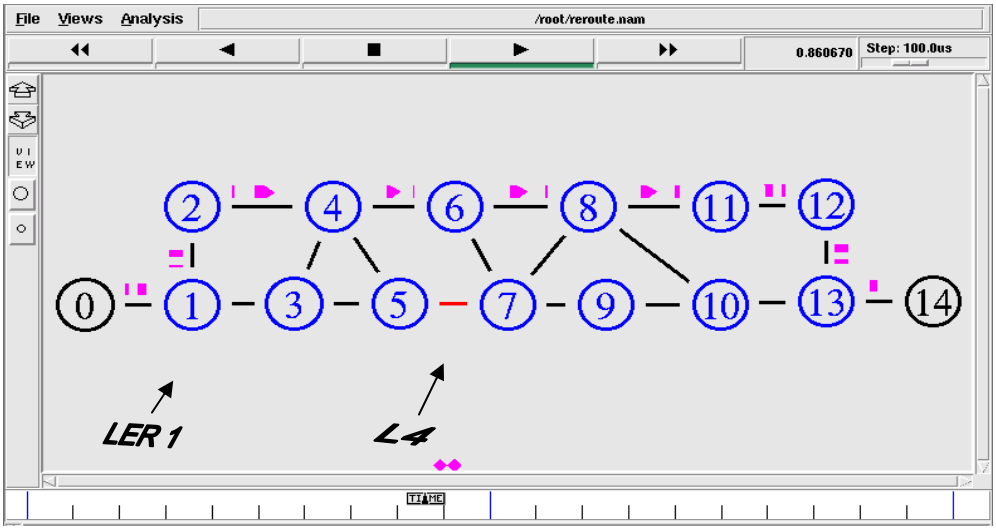


Figura 25 - Proteção (MPCG), falha no enlace L4

4.4.2.4 Topologia 2, falha no enlace L6

Como mostra a Figura 26, ocorreu uma falha no enlace L6. Nesta situação, o LSR9 assume a função de transmitir o FIS para o LER1, para que em seguida o tráfego seja re-roteado ao LSP de recuperação pré-estabelecido que corresponde à rota 1_2_4_6_8_11_12_13.

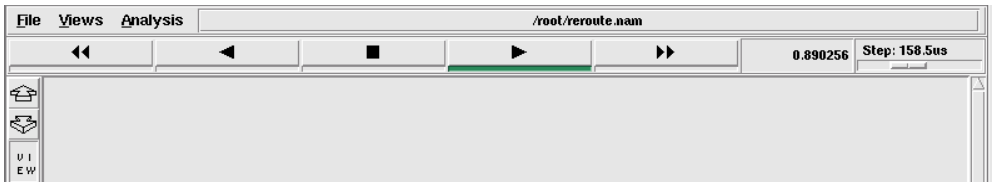


Figura 26 - Proteção (MPCG), falha no enlace L6

A Tabela 1 mostra os resultados de perda de pacotes e atraso da simulação do Método de Proteção de Caminho Global (MPCG), tendo como parâmetros a falha nos enlace L3 e L5 da topologia 1 e nos enlaces L4 e L6 da topologia 2, para tráfego exponencial e CBR. Pode-se observar através dos gráficos das Figuras 27 e 28, gerados a partir da Tabela 1, que o número de pacotes perdidos é aproximadamente 35% maior quando a falha ocorre nos enlace L5 da topologia 1 e enlace L6 da topologia 2, tanto para tráfego exponencial como para tráfego CBR. Pode-se então inferir que quanto mais distante do LER de ingresso (LER1) for a falha, maior será a perda de pacotes. Isso pode ser explicado pelo fato de que enquanto o FIS não chegar ao LER1, o tráfego de pacotes não é re-roteado.

Em relação ao atraso, identificaram-se resultados próximos para todos os parâmetros testados, uma vez que, independentemente do local da falha, o caminho de recuperação é de mesmo tamanho.

Tabela 1 - Resultados da simulação da Proteção (MPCG)

Topologia	Tráfego	Enlace da falha	Média de pacotes	Média de pacotes	Intervalo de confiança		Atraso médio	Intervalo de confiança	
1	Exp.	L3	2846,2	9	9,00	9,00	39,38	39,29	39,46
		L5	2831,4	16,2	15,64	16,76	39,53	39,51	39,56
	CBR	L3	1436	5			62,81		
		L5	1421	10			63,34		
2	Exp.	L4	2966,8	12,6	12,17	13,03	43,95	43,91	44,00
		L6	2919,6	20	20,00	20,00	44,5	44,44	44,55
	CBR	L4	1555	8			69,49		
		L6	1512	12			71,27		

Nota: Elaboração própria.

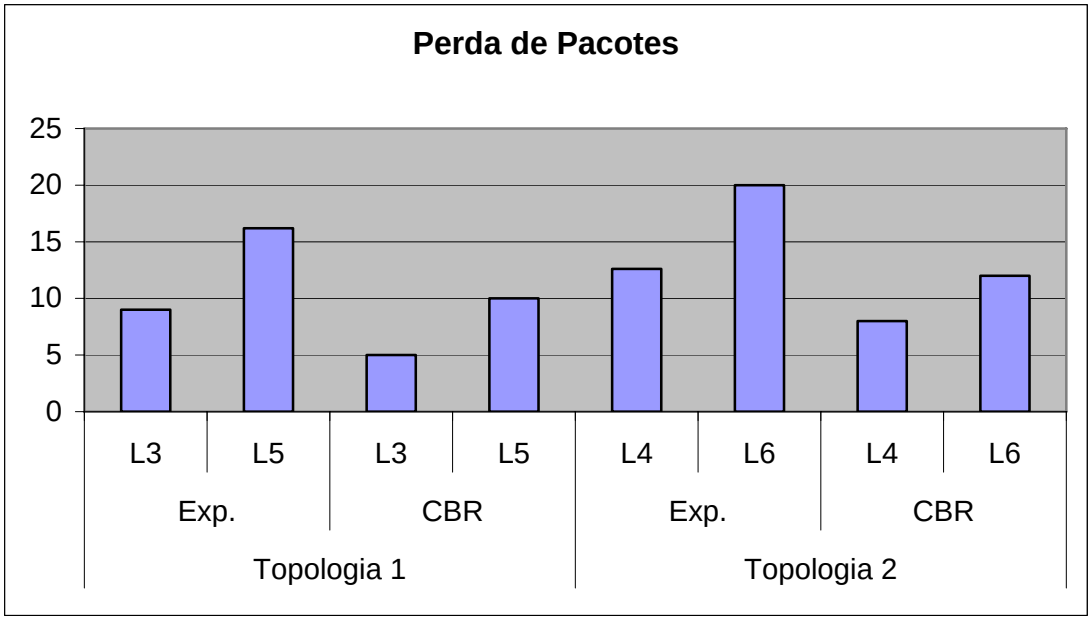


Figura 27 – Gráfico da perda de pacotes (MPCG)

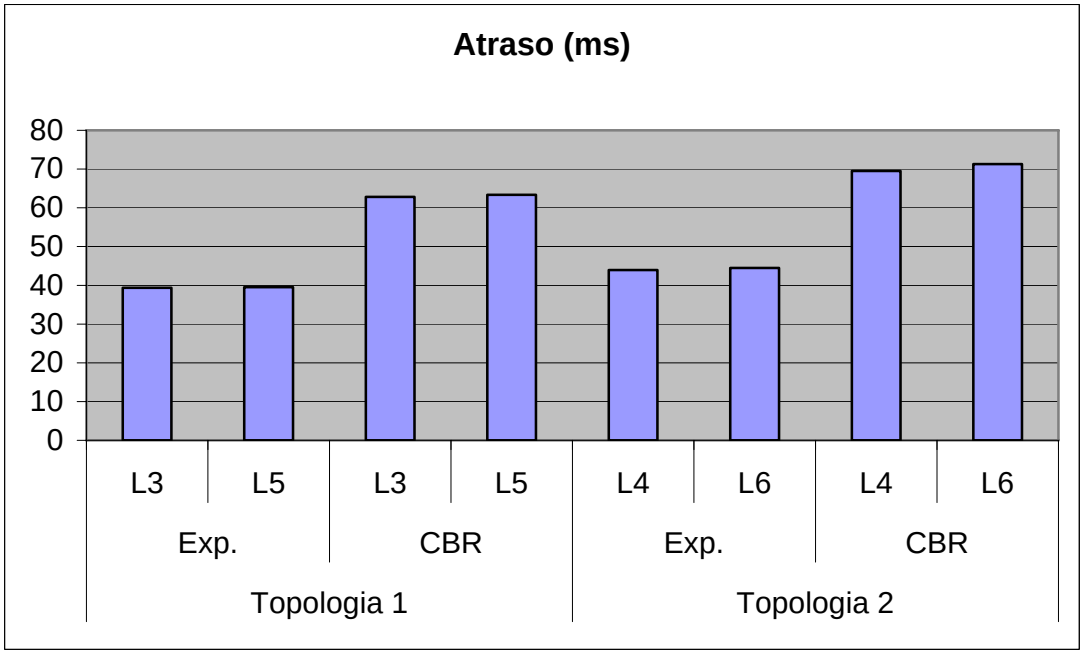


Figura 28 – Gráfico do atraso (MPCG)

4.4.3 Teste do Método de Proteção de Caminho Reverso (MPCR)

4.4.3.1 Topologia 1, falha no enlace L3

Nesta simulação, conforme a Figura 29, tão logo a falha é detectada, o LSR2, no início do enlace com falha, re-roteia o tráfego para o LSP de recuperação na

direção oposta, de volta para o nó de ingresso. Os LSP's de trabalho e recuperação são respectivamente: 1_2_3_4_5_6 e 1_7_8_9_10_6.

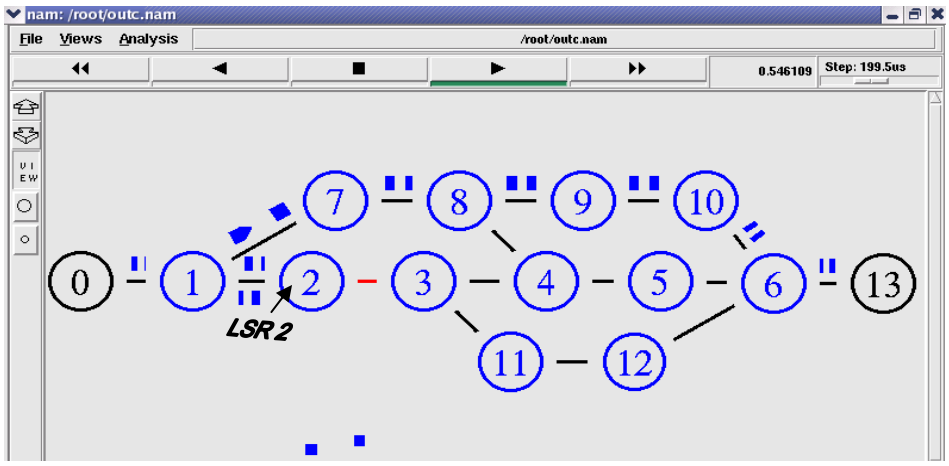


Figura 29 - Proteção (MPCR), falha no enlace L3

4.4.3.2 Topologia 1, falha no enlace L5

Nesta simulação, como mostra a Figura 30 o LSR4, no início do enlace com falha, é quem re-roteia o tráfego para o LSP de recuperação.

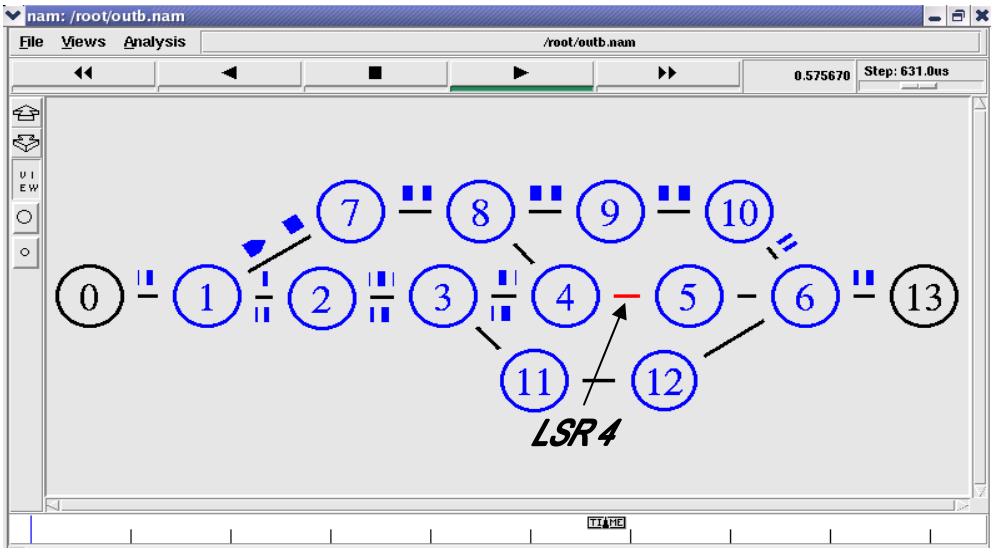


Figura 30 - Proteção (MPCR), falha no enlace L5

4.4.3.3 Topologia 2, falha no enlace L4

Conforme a Figura 31 tão logo a falha é detectada, o LSR2 re-roteia o tráfego para o LSP de recuperação na direção oposta, de volta para o nó de ingresso. Os

LSP's de trabalho e recuperação são respectivamente: 1_3_5_7_9_10_13 e 1_2_4_6_8_11_12_13.

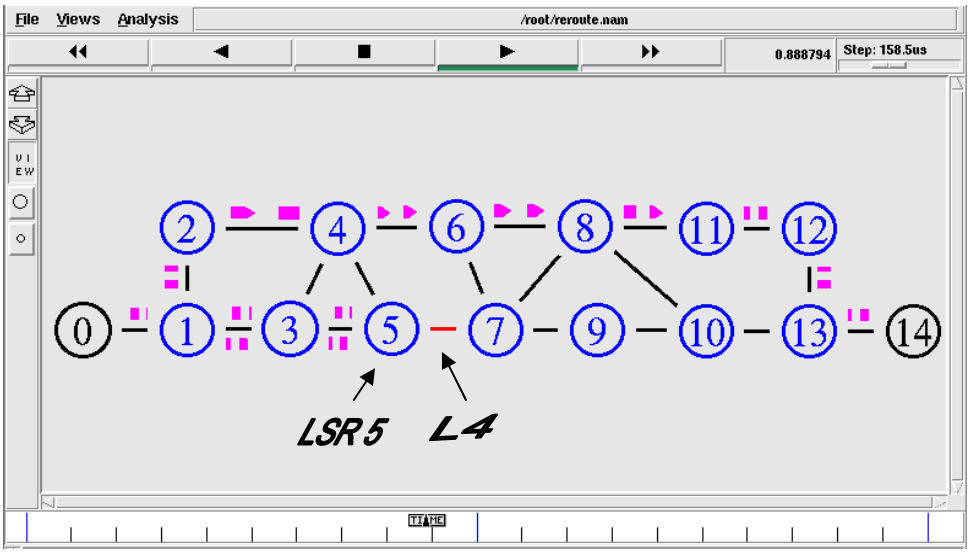


Figura 31 - Proteção (MPCR), falha no enlace L4

4.3.3.4 Topologia 2, falha no enlace L6:

Nesta simulação, o LSR9, no início do enlace com falha, é quem re-roteia o tráfego para o LSP de recuperação, conforme a Figura 32.

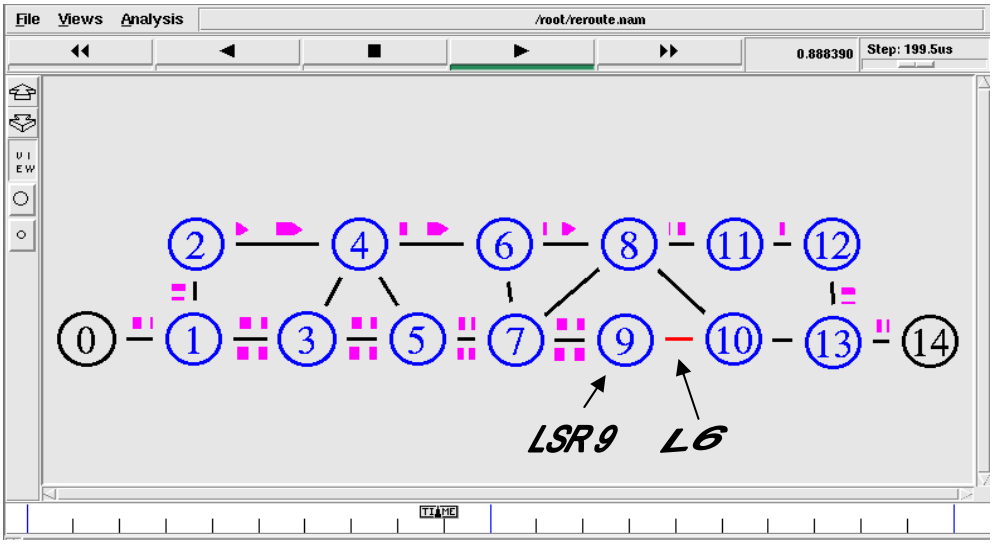


Figura 32 - Proteção (MPCR), falha no enlace L6

A Tabela 2 mostra os resultados da simulação do Método de Proteção de Caminho Reverso (MPCR). Pode-se observar através dos gráficos das Figuras 33 e 34 gerados a partir da Tabela 2, que o número de pacotes perdidos nos dois locais de falha da topologia 1 e topologia 2, para as simulações com tráfego exponencial e CBR, é o mesmo. Isso deve-se ao fato de que nestas simulações, tão logo ocorre a falha, o tráfego é re-roteado no sentido contrário, perdendo-se apenas os pacotes que trafegam no enlace, no instante da falha.

Em relação ao atraso, fica evidente que, quando a falha ocorre no enlace L5 da topologia 1 e no enlace L6 da topologia 2, ele é maior. Esse aumento explica-se em conseqüência do aumento do caminho a ser percorrido pelo tráfego re-roteado até o LER1.

Tabela 2 - Resultados da simulação da Proteção (MPCR)

Tipologia	Tráfego	Enlace da falha	Média de pacotes enviados	Média de pacotes perdidos	Intervalo de confiança		Atraso médio (ms)	Intervalo de confiança	
	Exp.	L3	2850,4	5,2	4,64	5,76	40,72	40,67	40,78
1		L5	2833	6	5,00	5,00	43,85	43,76	43,95
	CBR	L3	1444	3			69,34		
		L5	1425	3			84,23		
	Exp.	L4	2972,8	5,2	4,64	5,76	46,63	46,55	46,72
2		L6	2931,6	5	5,00	5,00	50,10	49,97	50,24
	CBR	L4	1563	3			82,02		
		L6	1516	3			97,77		

Nota: Elaboração própria.

Figura 33 – Gráfico da perda de pacotes (MPCR)

Figura 34 – Gráfico do atraso (MPCR)

4.4.4 Teste do Método de Proteção de Múltiplos Segmentos (MPMS)

4.4.4.1 Topologia 1, falha no enlace L3

Nesta simulação, o LSP de trabalho foi dividido em dois segmentos de trabalho S1 (1_2_3_4) e S2 (3_4_5_6), protegidos respectivamente pelos segmentos de recuperação (1_7_8_4) e (3_11_12_6). Conforme a Figura 35, tão logo ocorre a falha, o LSR2 inverte o tráfego re-roteando-o para o segmento de recuperação que se inicia no LER1.

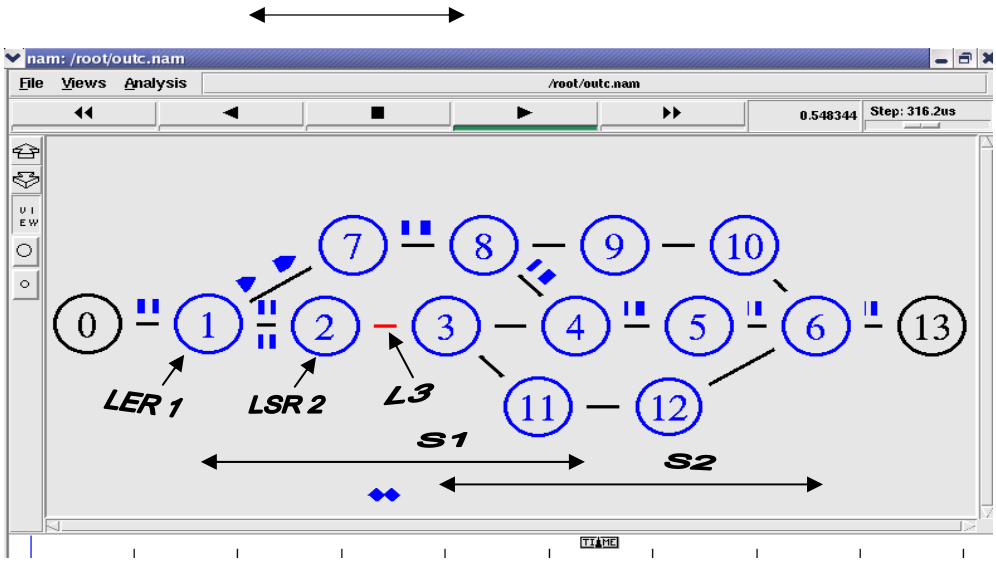


Figura 35 - Proteção (MPMS), falha no enlace L3

4.4.4.2 Topologia 1, falha no enlace L5

Conforme a Figura 36, nesta situação, tão logo ocorre a falha, o LSR4 inverte o tráfego re-roteando-o para o segmento de recuperação que se inicia no LSR3 e não mais no LER1, como no Método de Proteção de Caminho Reverso.

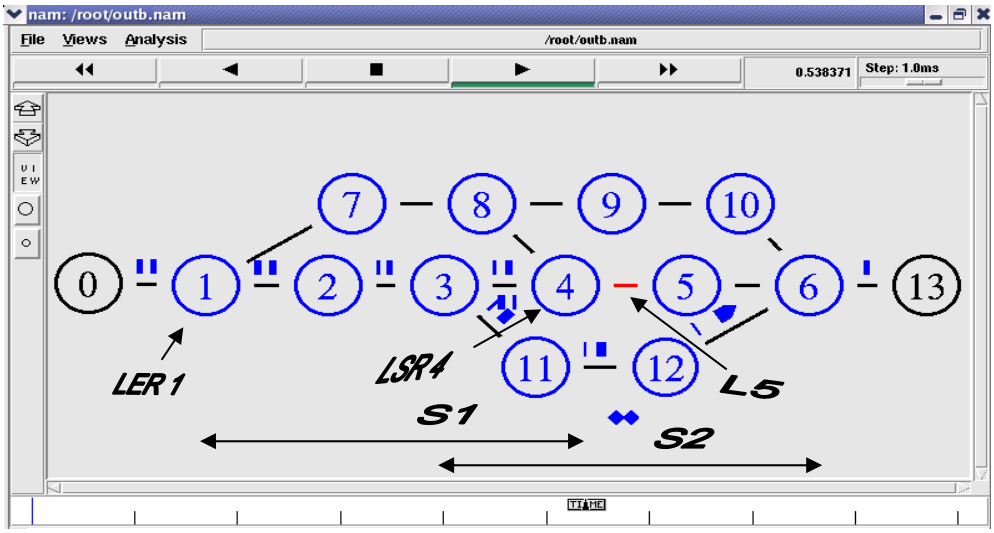


Figura 36 - Proteção MPMS (Topologia 1, falha no enlace L5)

4.4.4.3 Topologia 2, falha no enlace L4

Nesta simulação, o LSP de trabalho dividiu-se em três segmentos de trabalho S1 (1_3_5), S2 (3_5_7) e S3 (7_9_10_13), protegidos respectivamente pelos segmentos de recuperação (1_2_4_6_7), (3_4_6_7) e (7_8_11_12_13). Conforme a Figura 37, tão logo ocorre a falha, o LSR5 inverte o tráfego re-roteando-o para o segmento de recuperação que se inicia no LSR3.

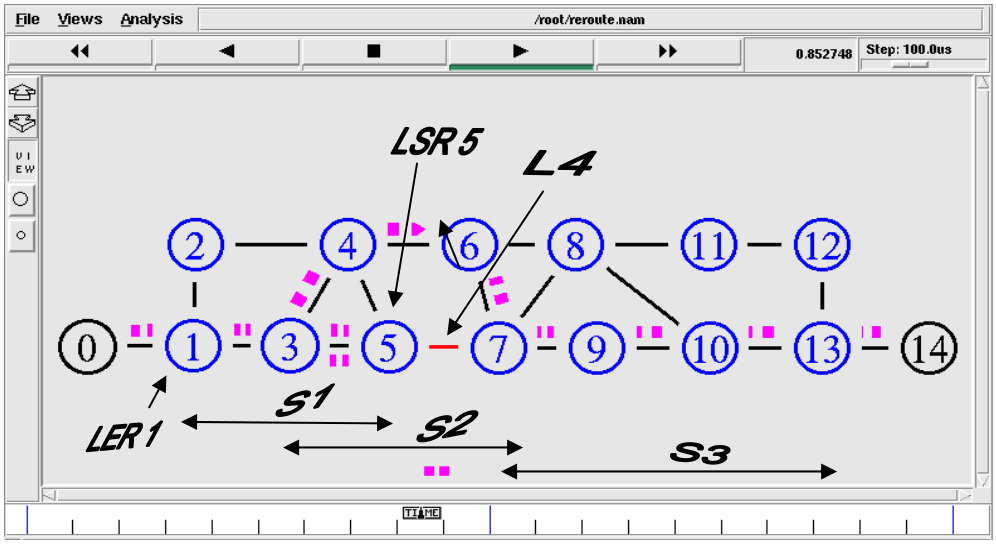


Figura 37 - Proteção MPMS (Topologia 2, falha no enlace L4)

4.4.4.4 Topologia 2, falha no enlace L6

A Figura 38 mostra o Método de Proteção de Múltiplos Segmentos (MPMS), simulando a falha no enlace L6. Nesta situação, o LSR9 inverte o tráfego re-roteando-o para o segmento de recuperação que se inicia no LSR7.

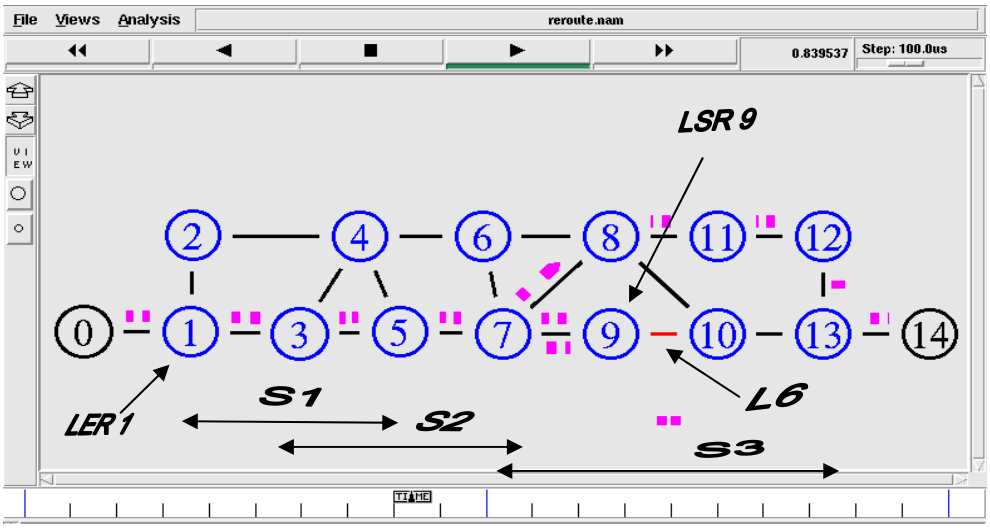


Figura 38 - Proteção MPMS (Topologia 2, falha no enlace L6)

A Tabela 3 mostra os resultados da simulação do método de proteção de Múltiplos Segmentos (MPMS). Pode-se observar através dos gráficos das Figuras 39 e 40, gerados a partir da Tabela 3, que o número de pacotes perdidos em todas as situações de tráfego, local da falha e topologia, é o mesmo. Isso pode ser justificado

pelas mesmas razões apresentadas no Método MPCR, ou seja, devido a inversão do tráfego assim que ocorre a falha. Em relação ao atraso, os resultados encontrados para os dois tipos de tráfego na topologia 1, foram semelhantes, uma vez que, os caminhos de recuperação são do mesmo tamanho.

Quanto a topologia 2, verificou-se um aumento do atraso no enlace L6 para ambos os tipos de tráfego, o que pode ser explicado pelo aumento do caminho de recuperação.

Conclui-se então, que na aplicação do Método de Proteção de Múltiplos Segmentos (MPMS), o atraso é diretamente proporcional ao tamanho do caminho de recuperação.

Tabela 3 - Resultados da simulação da Proteção (MPMS)

Topologia	Tráfego	Enlace da falha	Média de pacotes enviados	Média de pacotes perdidos	Atraso (S/ falha)	Atraso (C/falha)	Intervalo de confiança		Atraso médio (ms)	Intervalo de confiança	
1	Exp.	L3	2850,4	5,2	18,65	21,63	4,64	5,76	40,72	40,67	40,78
		L5	2830,4	5	19,27	22,35	5,00	5,00	41,01	40,97	41,05
	CBR	L3	1444	3	22,02	23,66			69,36		
		L5	1421	3	23,01	24,4			70,48		
2	Exp.	L4	2970,4	5,2	18,83	22,64	4,64	5,76	45,24	45,09	45,39
		L6	2918,6	4,6	20,14	23,33	3,49	5,71	45,97	45,89	46,05
	CBR	L4	1561	3	22,09	23,97			75,81		
		L6	1510	3	24,56	25,18			78,31		

Fonte: Elaboração própria.

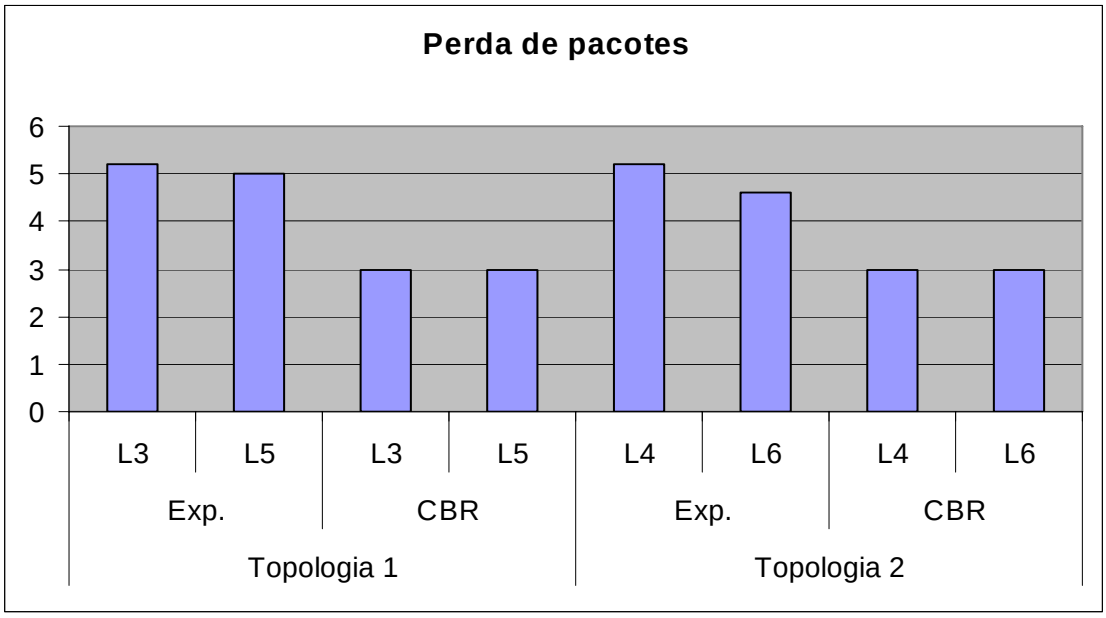


Figura 39 – Gráfico da perda de pacotes (MPMS)

Figura 40 – Gráfico do atraso (MPMS)

4.4.4 Análise comparativa entre os métodos

Conforme os gráficos das Figuras 41 e 42, verificou-se que em relação às perdas de pacotes, a Proteção MPCG demonstrou maiores perdas do que os outros métodos, independente da topologia aplicada. Assim como, a Proteção MPCR, foi a que apresentou o maior índice de atraso, independente da topologia e do tráfego utilizado.

A Proteção MPMS apresentou menos perdas de pacotes, quando comparado com a Proteção MPCG, bem como menores atrasos quando comparado com a Proteção MPCR.

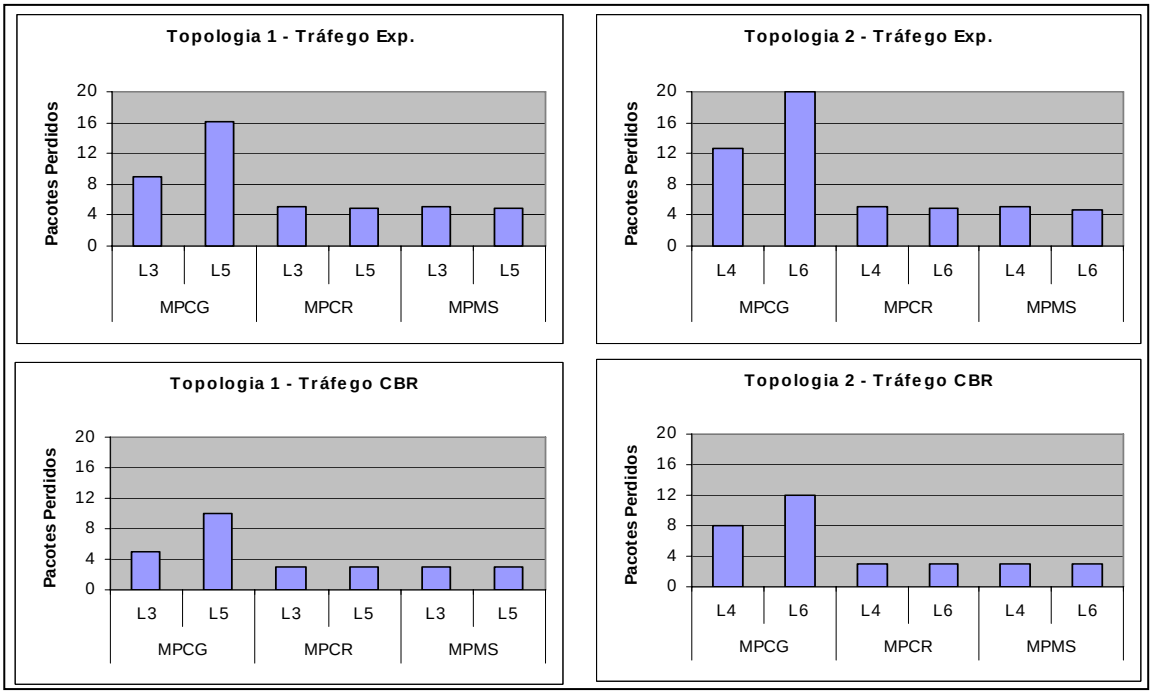


Figura 41 – Gráficos comparativos dos métodos de proteção (perda de pacotes)

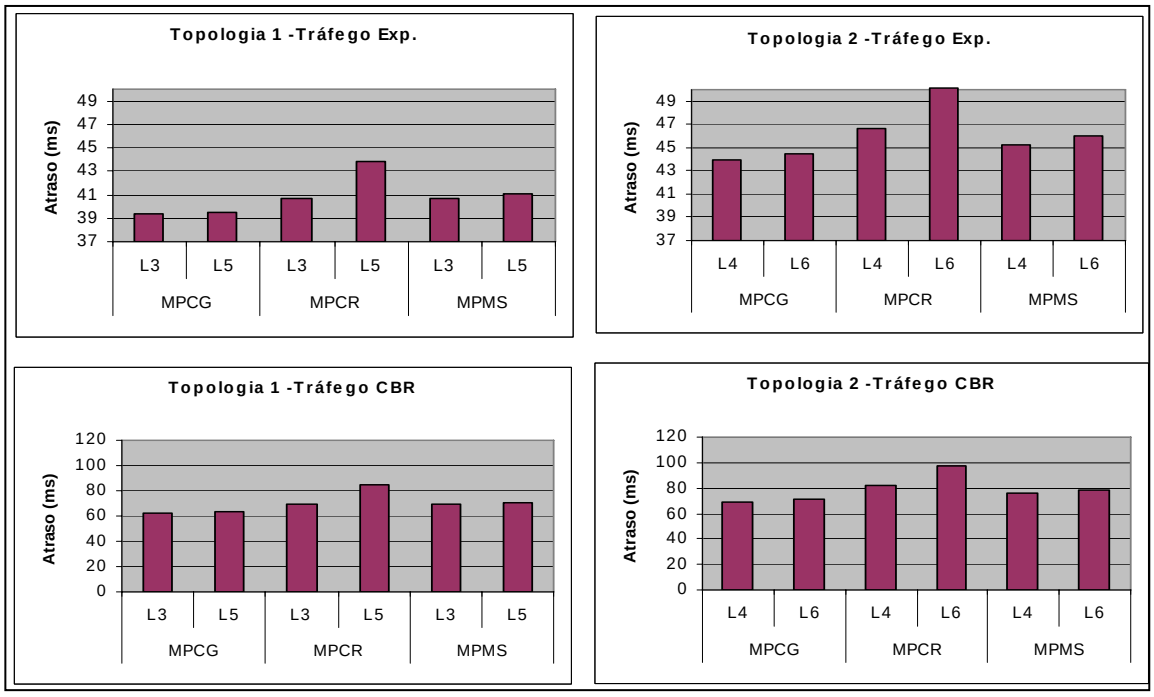


Figura 42 – Gráficos comparativos dos métodos de proteção (atraso)

5 CONCLUSÕES

Com o surgimento de novas aplicações em tempo real, as organizações estão cada vez mais dependentes das redes de computadores. Para atender as exigências de serviços dessas aplicações, é importante o aprimoramento e o desenvolvimento de novas técnicas de recuperação de falhas na rede.

Observa-se que, a partir da capacidade de conferir orientação à conexão em redes IP, facilitando a implementação de técnicas de engenharia de tráfego e re-roteamento rápido, a tecnologia de comutação de rótulos multiprotocolo - *Multiprotocol Label Switching (MPLS)* desponta como promissora para o suporte aos métodos de recuperação de falhas.

Não é possível ter mecanismos de recuperação completamente sem perdas, porém é possível ter mecanismos que minimizem a perda de pacotes ao máximo. Esse é o objetivo principal dos métodos de proteção.

O pré-estabelecimento dos recursos de recuperação é a principal característica dos métodos de proteção. Se não fossem pré-estabelecidos teriam que ser configurados após a detecção da falha, e nesse caso, a convergência não seria rápida o suficiente para as aplicações em tempo real.

Para avaliar e comparar os parâmetros de desempenho do Método de Proteção de Múltiplos Segmentos (MPMS) com os Métodos de Proteção de Caminho Global (MPCG) e Proteção de Caminho Reverso (MPCR), realizou-se experimentos no simulador de redes - *Network Simulator (NS).*

A estratégia adotada foi a medição do número de pacotes perdidos e do atraso. Sendo assim, criou-se duas topologias e simulou-se uma falha em dois enlaces distintos de cada uma, utilizando primeiro tráfego exponencial e em seguida tráfego de taxa constante de bits - *Constant Bit Rate (CBR)*.

Para a obtenção dos resultados, efetuou-se 72 replicações de 10 segundos, alterando os fatores e faixas de valores dos parâmetros de avaliação, além da semente para geração de números aleatórios. Trabalhou-se também, com um intervalo de confiança de 95% para garantir mais confiabilidade nos resultados obtidos.

Em relação ao percentual de perdas de pacotes, a Proteção MPMS obteve desempenho igual ao da Proteção MPCR, porém comparado-se a Proteção MPCG, obteve um ganho de desempenho de 40% quando a falha ocorreu nos enlaces mais próximos do nó de origem, e um ganho de 75% quando a falha ocorreu nos enlaces mais distantes.

Em relação ao atraso, a Proteção MPMS obteve uma perda de desempenho de 11,8% comparado a Proteção MPCG e um ganho de 23,8% comparado ao método de proteção MPCR.

Comprovando as afirmações anteriores, verificou-se que em relação à perda de pacotes, como nas proteções MPMS e MPCR o tráfego é re-roteado no sentido inverso pelo nó mais próximo da falha, perdem-se apenas os pacotes que estão trafegando no enlace no instante da falha. Enquanto que, na Proteção MPCG o tráfego é re-roteado apenas pelo nó de origem. Portanto quanto mais distante do nó de origem ocorrer a falha, mais pacotes serão perdidos, pois mais tempo levará para o sinal de indicação de falha *Fault Indication Signal* (FIS) sinalizar a falha.

Quanto ao atraso é em decorrência do tamanho do caminho a ser percorrido pelo tráfego re-roteado.

5.1 CONTRIBUIÇÕES

Como principais contribuições deste trabalho, destacam-se:

1) Implementação do Método de Proteção de Múltiplos Segmentos (MPMS), no NS.

2) Estudo comparativo do desempenho dos métodos de proteção MPMS, MPCG e MPCR.

3) Referencial teórico e prático para implementação, avaliação e comparação de novas situações de re-roteamento rápido em redes de computadores no NS.

5.2 TRABALHOS FUTUROS

A partir deste trabalho, sugere-se:

1) O aprofundamento do estudo do algoritmo desenvolvido para seleção dos segmentos de trabalho e recuperação, utilizados no Método de Proteção de Múltiplos Segmentos (MPMS) proposto por Dahai Xu, a fim de implementá-lo em computadores com sistema operacional LINUX, com função de roteadores e no núcleo do NS.

2) O estudo da tecnologia MPLS generalizado (GMPLS) e implementação da proteção de falhas em rede óptica, no NS.

REFERÊNCIAS

AWDUCHE, D. et al. **Requirements for traffic engineering over MPLS. RFC 2702.** 1999. Disponível em: <http://www.ietf.org/rfc/rfc2702.txt>. Acesso em: 15 jan. 2004.

_____. **LDP Specification RFC3036 internet engineering task force**. 2001. Disponível em: <ftp://ftp.rfc-editor.org/in-notes/rfc3036.txt>. Acesso em: 20 fev. 2004.

_____.**RSVP-TE: extensions to RSVP for LSP tunnels**. **RFC 3209**. 2001. Disponível em: <http://www.ietf.org/rfc/rfc3209.txt>. Acesso em: 15 jan. 2004.

_____. **Overview and principles of internet traffic engineering**. **RFC 3272.** 2002. Disponível em: <http://www.ietf.org/rfc/rfc3272.txt>. Acesso em: 15 jan. 2004.

AWDUCHE, D.; BIJAN, J. Internet traffic engineering using Multi-Protocol Label Switching (MPLS). **Computer Networks,** v.40, n. 1, p.111-129, 2002. Disponível em: <http://www.awduche.com/papers/papers2002/TE_COMPNTWK_Sep02.pdf>. Acesso em: 3 jun. 2005.

CALLE E. et al. Protection performance components in MPLS networks. In: INTERNATIONAL SYMPOSIUM ON PERFORMANCE EVALUATION OF COMPUTER AND TELECOMMUNICATION SYSTEMS (SPECTS), 2003, Montreal (Canadá), 2003, **Anais eletrônicos...** 2003, Montreal (Canadá), 2003. Disponível em: <http://bcds.udg.es/?publications.html>. Acesso em: 6 dez. 2003.

CHUNG, J.; CLAYPOOL, M. **NS by example.** Disponível em: <http://nile.wpi.edu/NS>. Acesso em: 10 jan. 2004.

COUTINHO, M. **Network simulator, guia básico para iniciantes**. 2003. Disponível em: <http://www.cci.unama.br/margalho/networksimulator/>. Acesso em: 25 jan.2004.

DAHAI X. et al. Novel algorithms for shared segment protection. **IEEE Journal on Selected Areas in Communications,** v. 21, n. 8, p. 1320-1331, oct. 2003. Disponível em: <http://www.cse.buffalo.edu/~MPMSxu/files/jsac03.pdf>. Acesso em: 18 jan. 2004.

DANNY Y. **Traffic engineering prioritized IP packets over multi-protocol label switching networks, project submitted in partial fulfillment of the requirements for the degree of master of engineering**. 2002. Disponível em: <http://www.ensc.sfu.ca/people/faculty/ljilja/data/projects_cnl_simon.html>. Acesso em: 20 nov. 2003.

DAVIE, B. et al. **MPLS using LDP and ATM VC switching. RFC 3035. Internet engineering task force**. 2001. Disponível em: <ftp://ftp.rfc-editor.org/in-notes/rfc3035.txt>. Acesso em: 15 maio. 2004.

GAEIL, A.; W. Chun, Design and implementation of MPLS network simulator (MNS). In: **IEEE Internet**. Disponível em: <http://flower.ce.cnu.ac.kr/~fog1/mns/>, Acesso em: 2 dez. 2003.

_____. Simulator for MPLS path restoration and performance evaluation. In: **IEEE Internet. Conf. on Networks (ICON2000)**, Singapore. Disponível em: <http://flower.ce.cnu.ac.kr/~fog1/mns/>. Acesso em: 2 dez. 2003.

GANDHI A.; R. B. **Dynamic issues in MPLS service restoration**. Tech. Rec. Tr 01-10. New Hampshire: Univ. of New Hampshire. Dept. of Computer Science, 2001. Disponível em: <http://www.cs.unh.edu/cnrg/papers/pdcs02.pdf>. Acesso em: 11 nov. 2003.

JAMOUSSI B. (Ed.) **Constraint-based LSP setup using LDP. RFC 3212.** 2002. Disponível em: <http://www.faqs.org/rfcs/rfc3212.html>. Acesso em: 20 dez. 2003.

HASKIN, D.; KRISHNAN, R. **A method for setting an alternative label switched paths to handle fast reroute:** work in progress. [draft-MPCR-mpls-fast-reroute-05.txt], 2000. Disponível em: <http://quimby.gnus.org/internet-drafts/draft-MPCR-mpls-fast-reroute-02.txt>. Acesso em: 11 nov. 2003.

KUROSE, J. F.; KEITH W. R. **Rede de computadores e a internet:** uma nova abordagem. Tradução. Arlete Simille Marques. São Paulo: Addison Wesley, 2003.

MAGALHÃES, M.; CARDOSO E. Introdução à comutação IP por rótulo através de MPLS. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES – SBRC, 19., 2001, Florianópolis, **Anais eletrônicos...** 2001, Florianópolis, 2001. Disponível em: <http://www.dca. fee.unicamp.br/~eleri/ia365/refs.html>, Acesso em: 18 dez. 2003.

MALEK, J. **Trace files analyser trace graph 2.02**. 2004. Disponível em: <http://www. geocites.com/tracegraph/>. Acesso em: 10 out. 2004.

OSBORNE, E.; AJAY, S. **Engenharia de tráfego com MPLS:** projeto, configuração e gerenciamento do MPLS para otimização de desempenho de rede. Tradução de Daniel Vieira. Rio de Janeiro: Campus, 2002.

PAN, P. et al. **Fast reroute extensions to RSVP-TE for LSP tunnels , work in progress**. [draft-ietf-mpls-rsvp-lsp-fastreroute-00.txt]. 2002. Disponível em: <http://www.ietf.org/proceedings/02jul/I-D/draft-ietf-mpls-rsvp-lsp-fastreroute-00.txt>. Acesso em: 10 jan. 2004.

PETERSON, L.; BRUCE, S. **Redes de computadores:** uma abordagem de sistemas. Tradução de Daniel Vieira. Rio de Janeiro: Campus, 2004.

PORTNOI, M.; RAFAEL, G. SOBRNOME **Network simulator:** visão geral da ferramenta de simulação de redes. Disponível em: <http://www.angelfire.com/electronic2/lock smith/articles/networksimulator-sepa.pdf>. Acesso em: 03 dez. 2003.

_____. SOBRNOME **CR-LDP:** aspectos e funcionamento. 2005. Disponível em: <http://www.geocities.com/ResearchTriangle/4480/academic/academic-files/cr-ldp_ as pectos_funcionamento.pdf>. Acesso em: 28 set. 2005.

ROSEN, E.; VISWANATHAN, A.; CALLON, R. **Multiprotocol label switching architecture**. January, 2001. RFC 3031. Disponível em: <http://www.ietf.org/rfc/rfc3130. txt>. Acesso em: 25 jun. 2004.

SHARMA, V. **Framework for MPLS-based recovery**. **RFC 3469.** 2003. Disponível em: <http://www.ietf.org/rfc/rfc3469.txt>. Acesso em: 22 jan. 2004.

TAUMATURGO, O. **Alocação dinâmica de banda e re-roteamento dinâmico de LSPs em Redes MPLS**. 2003. 104 f. Dissertação (Mestrado Integrado Profissionalizante em Computação) – Centro Federal de Educação Tecnológica do Ceará – CEFET, 2003.

VINT/UCB/LBNL. **Network Simulator, NS notes and documentation**. 2004. Disponível em: <http://www-mash.cs.berkeley.edu/ns>. Acesso em: 3 dez. 2005.

WEBER, T. **Tolerância a falhas:** conceitos e exemplos. Local: Universidade Federal do Rio Grande do Sul (UFRGS). Programa de Pós-Graduação em Computação - Instituto de Informática, 2004 Disponível em: <http://www.inf.ufrgs.br/~taisy/disciplinas/textos/ConceitosDependabilidade.PDF>. Acesso em: 3 set. 2004.

YOON, S. An efficient recovery mechanism for MPLS-based Protection LSP. In: **Joint 4th IEEE Internet. Conf. on ATM (ICATM 2001)**. Seoul: Korea, 2001, Disponível em: <http://www.angelfire.com/electronic2/locksmith/articles/networksimula tor-sepa.pdf>. Acesso em: 3 dez. 2003.

APÊNDICE A – Script de Simulação para a Topologia 1

```
# IDENTIFICAÇÃO
# Universidade Salvador - Unifacs
# Marcos Guimarães Fonseca
# Orientador Prof. Dr. José Augusto Suruagy

# Script de simulação dos métodos de proteção no NS para a topologia 1
#############################################################################

# Select methods of protection

proc usage { } {
      puts stderr {usage: ns test-reroute.tcl reroute-option
The reroute-options are as follows:
    - drop             : drop all traffic around link failure.
    - MPCR           : the scheme proposed by MPCR.
    - MPCG            : the scheme proposed by MPCG.
    - MPMS            : the scheme proposed by MPMS.
          }
}

# Create simulator object

set ns [new Simulator]

# Open files to write trace-data for NAM and Xgraph

# Finish procedure which closes the trace file and opens Xgraph and NAM

proc finish {} {
      global ns nf f
      $ns flush-trace
      close $nf
      close $f
      #close $fs

      puts "running nam..."
      exec nam outc.nam &

      exit 0
}
# Set dynamic distance-vector routing protocol

$ns rtproto DV

set f [open outc.tr w]
$ns trace-all $f
set nf [open outc.nam w]
$ns namtrace-all $nf

# Define nodes & MPLS LSRs

set n0     [$ns node]
set LSR1   [$ns MPLSnode]
set LSR2   [$ns MPLSnode]
set LSR3   [$ns MPLSnode]
set LSR4   [$ns MPLSnode]
set LSR5   [$ns MPLSnode]
```

```
set LSR6   [$ns MPLSnode]
set LSR7   [$ns MPLSnode]
set LSR8   [$ns MPLSnode]
set LSR9   [$ns MPLSnode]
set LSR10  [$ns MPLSnode]
set LSR11  [$ns MPLSnode]
set LSR12  [$ns MPLSnode]
set n13    [$ns node]

# Define links, bandwidth 1 Mb, delay 5ms, queue managementDropTail

$ns duplex-link $n0    $LSR1  1Mb  5ms DropTail
$ns duplex-link $LSR1  $LSR2  1Mb  5ms DropTail
$ns duplex-link $LSR2  $LSR3  1Mb  5ms DropTail
$ns duplex-link $LSR3  $LSR4  1Mb  5ms DropTail
$ns duplex-link $LSR4  $LSR5  1Mb  5ms DropTail
$ns duplex-link $LSR5  $LSR6  1Mb  5ms DropTail
$ns duplex-link $LSR1  $LSR7  1Mb  5ms DropTail
$ns duplex-link $LSR7  $LSR8  1Mb  5ms DropTail
$ns duplex-link $LSR8  $LSR4  1Mb  5ms DropTail
$ns duplex-link $LSR8  $LSR9  1Mb  5ms DropTail
$ns duplex-link $LSR9  $LSR10 1Mb  5ms DropTail
$ns duplex-link $LSR10 $LSR6  1Mb  5ms DropTail
$ns duplex-link $LSR3  $LSR11 1Mb  5ms DropTail
$ns duplex-link $LSR11 $LSR12 1Mb  5ms DropTail
$ns duplex-link $LSR12 $LSR6  1Mb  5ms DropTail
$ns duplex-link $LSR6  $n13   1Mb  5ms DropTail


# Control layout of the network

$ns duplex-link-op $n0    $LSR1  orient right
$ns duplex-link-op $LSR1  $LSR2  orient right
$ns duplex-link-op $LSR2  $LSR3  orient right
$ns duplex-link-op $LSR3  $LSR4  orient right
$ns duplex-link-op $LSR4  $LSR5  orient right
$ns duplex-link-op $LSR5  $LSR6  orient right
$ns duplex-link-op $LSR1  $LSR7  orient right-up
$ns duplex-link-op $LSR7  $LSR8  orient right
$ns duplex-link-op $LSR8  $LSR4  orient right-down
$ns duplex-link-op $LSR8  $LSR9  orient right
$ns duplex-link-op $LSR9  $LSR10 orient right
$ns duplex-link-op $LSR10 $LSR6  orient right-down
$ns duplex-link-op $LSR3  $LSR11 orient right-down
$ns duplex-link-op $LSR11 $LSR12 orient right
$ns duplex-link-op $LSR12 $LSR6  orient right-up
$ns duplex-link-op $LSR6  $n13   orient right

# configure ldp agents on all mpls nodes

$ns configure-ldp-on-all-mpls-nodes

#set ldp-message clolr

$ns ldp-request-color       blue
$ns ldp-mapping-color       red
$ns ldp-withdraw-color      magenta
$ns ldp-release-color       orange
$ns ldp-notification-color  green

# Define agent to send packets
```

```
proc attach-expoo-traffic { node sink size burst idle rate } {
      global ns defaultRNG

      $defaultRNG seed 6000

      set arrivalRNG   [new RNG]
      set arrival2RNG   [new RNG]
      set sizeRNG      [new RNG]

      set arrival_ [new RandomVariable/Exponential]
      $arrival_    set  avg_          $burst
      $arrival_    use-rng            $arrivalRNG
      $arrival_    set  burst_time_   $arrival_

      set arrivalidle_  [new RandomVariable/Exponential]
      $arrivalidle_   set  avg_       $idle
      $arrivalidle_   use-rng         $arrival2RNG
      $arrivalidle_   set  idle_time_ $arrivalidle_

      set source [new Agent/CBR/UDP]
      $ns attach-agent $node $source

      set traffic [new Traffic/Expoo]
      $traffic set packet-size $size
      $traffic set burst-time $burst
      $traffic set idle-time $idle
      $traffic set rate $rate

      $source attach-traffic $traffic

      $ns connect $source $sink
      return $source
}

# Monitoring queue
#calculate the drops packets and delay link


# Methods of rerouting

proc set-reroute-init {option} {
      global ns LSR1 LSR6 LSR2
        switch $option {
      drop                 { # set reroute action to take when a link fails
                                $ns enable-reroute drop
                                $LSR1 enable-data-driven
                              }

     MPCR                { # set reroute action to take when a link fails
                                $ns enable-reroute drop
                                $LSR2 set-protection-lsp 0.1 0.01 1000
                              }
     MPCG                 { # set reroute action to take when a link fails
                                 $ns enable-reroute notify-prenegotiated
                                $LSR2 set-protection-lsp 0.1 0.01 1000
                              }

     MPMS                  { # set reroute action to take when a link fails
                                $ns enable-reroute drop
                                $LSR2 set-protection-lsp 0.1 0.01 1000
```

```
                                }
    default               { usage
                                  exit
                                }
        }
}

# Option protection

proc set-reroute-event {option} {
global ns LSR1 LSR2 LSR3 LSR4 LSR5 LSR6
   switch $option {
   MPCR       {
                # The setup of working LSP
                $ns at 0.0  "$LSR1  setup-erlsp  6   2_3_4_5_6  1000"

                # The setup of alternative LSP
                $ns at 0.1  "$LSR1  setup-erlsp  6   1_7_8_9_10_6  2000"
                $ns at 0.2  "$LSR6  setup-erlsp  6   5_4_3_2_1_L2000  2005"

                # bind a flow to LSP
                $ns at 0.3  "$LSR1 bind-flow-erlsp   13   100   1000"

                # binding working LSP to alternative LSP
                $ns at 0.3  "$LSR1 reroute-lsp-binding      1000    2000"
                $ns at 0.3  "$LSR2 reroute-lsp-binding      1000    2005"
                $ns at 0.3  "$LSR3 reroute-lsp-binding      1000    2005"
                $ns at 0.3  "$LSR4 reroute-lsp-binding      1000    2005"
                $ns at 0.3  "$LSR5 reroute-lsp-binding      1000    2005"

                   }
   MPCG       {
                # The setup of working LSP
                $ns at 0.0  "$LSR1  setup-erlsp  6   2_3_4_5_6   1000"

                # The setup of alternative LSP
                $ns at 0.1  "$LSR1  setup-erlsp  6   7_8_9_10_6  2000"

                # bind a flow to LSP
                $ns at 0.3  "$LSR1 bind-flow-erlsp   13   100   1000"

                # binding working LSP to alternative LSP
                $ns at 0.3  "$LSR1 reroute-lsp-binding      1000    2000"
                }
   MPMS      {
               # The setup of working LSP
                $ns at 0.0  "$LSR1  setup-erlsp  6   2_3_4_5_6  1000"

                # The setup of alternative LSP
                $ns at 0.1  "$LSR1  setup-erlsp  6   1_7_8_4_5_6  2000"
                $ns at 0.2  "$LSR6  setup-erlsp  6   5_4_3_2_1_L2000  2005"

                # bind a flow to LSP
                $ns at 0.3  "$LSR1 bind-flow-erlsp   13   100   1000"

                # binding working LSP to alternative LSP
                $ns at 0.3  "$LSR1 reroute-lsp-binding      1000    2005"
                $ns at 0.3  "$LSR2 reroute-lsp-binding      1000    2005"

                $ns at 0.3  "$LSR3 reroute-lsp-binding      1000    2005"
                $ns at 0.3  "$LSR4 reroute-lsp-binding      1000    2005"
```

```
                $ns at 0.3  "$LSR5 reroute-lsp-binding      1000    2005"

                 }
                 }
   }

# Create a traffic sink and attach it to the node node14

set sink0 [new Agent/LossMonitor]
$ns attach-agent $n13  $sink0
$sink0 clear

# Create a traffic source

set src0 [attach-expoo-traffic $n0  $sink0 200 1 0.02 500k]

$src0 set fid_ 100
$ns color 100  blue

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

#set udp1 [new Agent/UDP]
#$udp1 set fid_ 100
#$ns attach-agent $n0 $udp1

#set src0 [new Application/Traffic/CBR]
#$src0 set packetSize_ 2000
#$src0 set interval_ 0.015
#$src0 attach-agent $udp1

#$src0 set fid_ 100
#$ns color 100  red

#$ns connect $udp1 $sink0

#set null0 [new Agent/Null]
#$ns attach-agent $n13 $null0

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


proc notify-erlsp-setup {node lspid} {

        set ns [Simulator instance]

        if { $lspid==1001 } {
             $node secondary-lsp-binding 1000 1001
        }
}

proc notify-erlsp-fail {node status lspid tr} {
        global LSR1

        set ns [Simulator instance]

        if { [$node id] == 1 && $status=="BSNodeError" } {
             $node set-lib-error-for-lspid $lspid 1
        }
        if { [$node id] == 1 && $status=="NodeRepair" } {
             $node set-lib-error-for-lspid $lspid -1
        }
```

```
   }
# reroute mechanisms

if {$argc == 1} {
   set-reroute-init  $argv
   set-reroute-event $argv
} else {
   usage
   exit
}

# Source start

$ns at 0.3  "$src0 start"

# Define link failures and when the link have to be restored

$ns rtmodel-at 5.0 down $LSR2 $LSR3
$ns rtmodel-at 6.0 up   $LSR2 $LSR3

$ns at 8.0 "$src0 stop"

# Calls the procedure "finish"

$ns at 10.0 "finish"

#Run the simulator

$ns run
```

APÊNDICE B – Script de Simulação para a Topologia 2

 # Script de simulação dos métodos de proteção no NS para a topologia 2

```
proc usage { } {
      puts stderr {usage: ns test-reroute.tcl reroute-option
The reroute-options are as follows:
    - drop             : drop all traffic around link failure.
    - MPCR           : the scheme proposed by MPCR.
    - MPCG            : the scheme proposed by MPCG.
    - MPMS            : the scheme proposed by MPMS.
          }
}

# Create simulator object

set ns [new Simulator]

# Open files to write trace-data for NAM and Xgraph

# Finish procedure which closes the trace file and opens Xgraph and NAM

proc finish {} {
      global ns nf f
      $ns flush-trace
      close $nf
      close $f
      #close $fs

      puts "running nam..."
      exec nam outd.nam &

      exit 0
}

# Set dynamic distance-vector routing protocol

$ns rtproto DV

set f [open outd.tr w]
$ns trace-all $f
set nf [open outd.nam w]
$ns namtrace-all $nf

set n0     [$ns node]
set LSR1   [$ns MPLSnode]
set LSR2   [$ns MPLSnode]
set LSR3   [$ns MPLSnode]
set LSR4   [$ns MPLSnode]
set LSR5   [$ns MPLSnode]
set LSR6   [$ns MPLSnode]
set LSR7   [$ns MPLSnode]
set LSR8   [$ns MPLSnode]
set LSR9   [$ns MPLSnode]
set LSR10  [$ns MPLSnode]
set LSR11  [$ns MPLSnode]
set LSR12  [$ns MPLSnode]
```

```
set LSR13  [$ns MPLSnode]
set n14    [$ns node]

# Configuração dos Links dos Nós

$ns duplex-link $n0    $LSR1  1Mb  5ms DropTail
$ns duplex-link $LSR1  $LSR3  1Mb  5ms DropTail
$ns duplex-link $LSR3  $LSR5  1Mb  5ms DropTail
$ns duplex-link $LSR3  $LSR4  1Mb  5ms DropTail
$ns duplex-link $LSR5  $LSR7  1Mb  5ms DropTail
$ns duplex-link $LSR7  $LSR8  1Mb  5ms DropTail
$ns duplex-link $LSR7  $LSR9  1Mb  5ms DropTail
$ns duplex-link $LSR1  $LSR2  1Mb  5ms DropTail
$ns duplex-link $LSR2  $LSR4  1Mb  5ms DropTail
$ns duplex-link $LSR4  $LSR5  1Mb  5ms DropTail
$ns duplex-link $LSR4  $LSR6  1Mb  5ms DropTail
$ns duplex-link $LSR6  $LSR7  1Mb  5ms DropTail
$ns duplex-link $LSR6  $LSR8  1Mb  5ms DropTail
$ns duplex-link $LSR8  $LSR10 1Mb  5ms DropTail
$ns duplex-link $LSR8  $LSR11 1Mb  5ms DropTail
$ns duplex-link $LSR11 $LSR12 1Mb  5ms DropTail
$ns duplex-link $LSR12 $LSR13 1Mb  5ms DropTail
$ns duplex-link $LSR9  $LSR10 1Mb  5ms DropTail
$ns duplex-link $LSR10 $LSR13 1Mb  5ms DropTail
$ns duplex-link $LSR13 $n14   1Mb  5ms DropTail

# Configuração da topologia da rede

$ns duplex-link-op $n0    $LSR1  orient right
$ns duplex-link-op $LSR1  $LSR3  orient right
$ns duplex-link-op $LSR3  $LSR5  orient right
$ns duplex-link-op $LSR3  $LSR4  orient up
$ns duplex-link-op $LSR5  $LSR7  orient right
$ns duplex-link-op $LSR7  $LSR8  orient right-up
$ns duplex-link-op $LSR7  $LSR9  orient right
$ns duplex-link-op $LSR1  $LSR2  orient up
$ns duplex-link-op $LSR2  $LSR4  orient right
$ns duplex-link-op $LSR4  $LSR5  orient right-down
$ns duplex-link-op $LSR4  $LSR6  orient right
$ns duplex-link-op $LSR6  $LSR7  orient down
$ns duplex-link-op $LSR6  $LSR8  orient right
$ns duplex-link-op $LSR8  $LSR10 orient right-down
$ns duplex-link-op $LSR8  $LSR11 orient right
$ns duplex-link-op $LSR11 $LSR12 orient right
$ns duplex-link-op $LSR12 $LSR13 orient right-down
$ns duplex-link-op $LSR9  $LSR10 orient right
$ns duplex-link-op $LSR10 $LSR13 orient right
$ns duplex-link-op $LSR13 $n14   orient right
# configure ldp agents on all mpls nodes

$ns configure-ldp-on-all-mpls-nodes

#set ldp-message clolr

$ns ldp-request-color       blue
$ns ldp-mapping-color       red
$ns ldp-withdraw-color      magenta
$ns ldp-release-color       orange
$ns ldp-notification-color  green

# Define agent to send packets
```

```
proc attach-expoo-traffic { node sink size burst idle rate } {
      global ns defaultRNG

      $defaultRNG seed 4000

      set arrivalRNG   [new RNG]
      set arrival2RNG   [new RNG]
      set sizeRNG      [new RNG]

      set arrival_ [new RandomVariable/Exponential]
      $arrival_    set  avg_          $burst
      $arrival_    use-rng            $arrivalRNG
      $arrival_    set  burst_time_   $arrival_

      set arrivalidle_  [new RandomVariable/Exponential]
      $arrivalidle_   set  avg_       $idle
      $arrivalidle_   use-rng         $arrival2RNG
      $arrivalidle_   set  idle_time_ $arrivalidle_

      set source [new Agent/CBR/UDP]
      $ns attach-agent $node $source

      set traffic [new Traffic/Expoo]
      $traffic set packet-size $size
      $traffic set burst-time $burst
      $traffic set idle-time $idle
      $traffic set rate $rate

      $source attach-traffic $traffic

      $ns connect $source $sink
      return $source
}

##########################################################################
# Methods of rerouting

proc set-reroute-init {option} {
      global ns LSR1 LSR7 LSR9 LSR13
      switch $option {
     drop                 { # set reroute action to take when a link fails
                              $ns enable-reroute drop
                              $LSR1 enable-data-driven
                            }

     MPCR                { # set reroute action to take when a link fails
                              $ns enable-reroute drop
                              $LSR9 set-protection-lsp 0.1 0.01 1000
                            }
     MPCG                 { # set reroute action to take when a link fails
                               $ns enable-reroute notify-prenegotiated
                          $LSR9 set-protection-lsp 0.1 0.01 1000
                            }

     MPMS                  { # set reroute action to take when a link fails
                              $ns enable-reroute drop
                              $LSR9 set-protection-lsp 0.1 0.01 1000
                            }
     default               { usage
                              exit
```

```
                                }
        }
}


# Opções de Proteção

proc set-reroute-event {option} {
      global ns LSR1 LSR3 LSR5 LSR7 LSR9 LSR10 LSR13
    switch $option {
MPCR   {
        # The setup of working LSP
        $ns at 0.0  "$LSR1  setup-erlsp  13   3_5_7_9_10_13  1000"

        # The setup of alternative LSP
        $ns at 0.1  "$LSR1  setup-erlsp  13   1_2_4_6_8_11_12_13  2000"
        $ns at 0.2  "$LSR13  setup-erlsp  13   13_10_9_7_5_3_1_L2000  2005"

        # bind a flow to LSP
        $ns at 0.3  "$LSR1 bind-flow-erlsp   14   100   1000"

        # binding working LSP to alternative LSP
        $ns at 0.3  "$LSR1 reroute-lsp-binding      1000    2000"
        $ns at 0.3  "$LSR3 reroute-lsp-binding      1000    2005"
        $ns at 0.3  "$LSR5 reroute-lsp-binding      1000    2005"
        $ns at 0.3  "$LSR7 reroute-lsp-binding      1000    2005"
        $ns at 0.3  "$LSR9 reroute-lsp-binding      1000    2005"
        $ns at 0.3  "$LSR10 reroute-lsp-binding      1000    2005"
        $ns at 0.3  "$LSR13 reroute-lsp-binding      1000    2005"

        }
MPCG  {
        # The setup of working LSP
        $ns at 0.0  "$LSR1  setup-erlsp  13   3_5_7_9_10_13   1000"

        # The setup of alternative LSP
        $ns at 0.1  "$LSR1  setup-erlsp  13   2_4_6_8_11_12_13  2000"

        # bind a flow to LSP
        $ns at 0.3  "$LSR1 bind-flow-erlsp   14   100   1000"

        # binding working LSP to alternative LSP
        $ns at 0.3  "$LSR1 reroute-lsp-binding      1000    2000"
        }
MPMS   {
        # The setup of working LSP
        $ns at 0.0  "$LSR1  setup-erlsp  13   3_5_7_9_10_13  1000"

        # The setup of alternative LSP
        $ns at 0.1  "$LSR1  setup-erlsp  13   7_8_11_12_13  2000"
        $ns at 0.2  "$LSR13  setup-erlsp  13   13_10_9_7_L2000  2005"

        # bind a flow to LSP
        $ns at 0.3  "$LSR1 bind-flow-erlsp   14   100   1000"

        # binding working LSP to alternative LSP
        #$ns at 0.3  "$LSR1 reroute-lsp-binding      1000    2000"
        #$ns at 0.3  "$LSR3 reroute-lsp-binding      1000    2005"
        #$ns at 0.3  "$LSR5 reroute-lsp-binding      1000    2005"
        $ns at 0.3  "$LSR7 reroute-lsp-binding      1000    2005"
        $ns at 0.3  "$LSR9 reroute-lsp-binding      1000    2005"
```

```
        $ns at 0.3  "$LSR10 reroute-lsp-binding      1000    2005"
       }
      }
}

#############################################################################
proc notify-erlsp-setup {node lspid} {

        set ns [Simulator instance]

        if { $lspid==1001 } {
             $node secondary-lsp-binding 1000 1001
        }
}

proc notify-erlsp-fail {node status lspid tr} {
        global LSR1

        set ns [Simulator instance]

        if { [$node id] == 1 && $status=="BSNodeError" } {
             $node set-lib-error-for-lspid $lspid 1
        }
        if { [$node id] == 1 && $status=="NodeRepair" } {
             $node set-lib-error-for-lspid $lspid -1
        }
   }

# reroute mechanisms

if {$argc == 1} {
   set-reroute-init  $argv
   set-reroute-event $argv
} else {
   usage
   exit
}


#############################################################################
# Create a traffic sink and attach it to the node node14

set sink0 [new Agent/LossMonitor]
$ns attach-agent $n14  $sink0
$sink0 clear

# Create a traffic source

set src0 [attach-expoo-traffic $n0  $sink0 200 1 0.02 500k]

$src0 set fid_ 100
$ns color 100  blue

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

#set udp1 [new Agent/UDP]
#$udp1 set fid_ 100
#$ns attach-agent $n0 $udp1

#set src0 [new Application/Traffic/CBR]
#$src0 set packetSize_ 2000
```

```
#$src0 set interval_ 0.015
#$src0 attach-agent $udp1

#$src0 set fid_ 100
#$ns color 100  red

#$ns connect $udp1 $sink0

#set null0 [new Agent/Null]
#$ns attach-agent $n14 $null0

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



# Source start

$ns at 0.3  "$src0 start"

# Define link failures and when the link have to be restored

$ns rtmodel-at 5.0 down $LSR9 $LSR10
$ns rtmodel-at 6.0 up   $LSR9 $LSR10

$ns at 8.0 "$src0 stop"

# Calls the procedure "finish"

$ns at 10.0 "finish"

#Run the simulator

$ns run
```