



**UNIVERSIDADE SALVADOR – UNIFACS  
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS E  
COMPUTAÇÃO  
MESTRADO EM SISTEMAS E COMPUTAÇÃO**

**WALTER DA COSTA PINTO NETO**

**PROPOSTA E AVALIAÇÃO: ALGORITMOS PARA A  
GERENCIA DE BANDA E SELEÇÃO DE CAMINHOS EM REDES  
MPLS (DS-TE)**

Salvador  
2008

**WALTER DA COSTA PINTO NETO**

**PROPOSTA E AVALIAÇÃO: ALGORITMOS PARA A  
GERENCIA DE BANDA E SELEÇÃO DE CAMINHOS EM REDES  
MPLS (DS-TE)**

Dissertação apresentada ao Curso de Mestrado Acadêmico em Sistemas e Computação, Universidade Salvador - UNIFACS, como requisito parcial para obtenção do grau de Mestre.

Orientador: Prof. Joberto Sérgio Barbosa Martins

Salvador  
2008

## FICHA CATALOGRÁFICA

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

Pinto Neto, Walter da Costa

Proposta e avaliação: algoritmos para gerência de banda e seleção de caminhos em redes MPLS (DS-TE)/Walter da Costa Pinto Neto. - 2008.

148 f.

Dissertação (Mestrado) - Universidade Salvador – UNIFACS. Mestrado em Sistemas de Computação, 2008.

Orientador: Prof. Joberto Sérgio Barbosa Martins

1. Algoritmos. 2. Redes 3. Gerência de banda. I. Martins, Joberto Sérgio Barbosa, orient. II. Título.

CDD: 004.6068

# TERMO DE APROVAÇÃO

WALTER DA COSTA PINTO NETO

## PROPOSTA E AVALIAÇÃO: ALGORITMOS PARA A GERENCIA DE BANDA E SELEÇÃO DE CAMINHOS EM REDES MPLS (DS-TE)

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre em Sistemas e Computação, Universidade Salvador - UNIFACS, pela seguinte banca examinadora:

JOBERTO SÉRGIO BARBOSA MARTINS – orientador  
Doutor pela *Universite de Paris VI (Pierre et Marie Curie), UP VI, França.*  
UNIVERSIDADE SALVADOR UNIFACS

JOSÉ AUGUSTO SURUAGY MONTEIRO  
Doutor pela *University of California Los Angeles, U.C.L.A., Estados Unidos*  
UNIVERSIDADE SALVADOR – UNIFACS

JOSÉ NEUMAN DE SOUZA  
Doutor pela *Universite de Paris VI (Pierre et Marie Curie), UP VI, França.*  
UNIVERSIDADE FEDERAL DO CEARÁ - UFC

Salvador, 19 de Dezembro de 2008

## **AGRADECIMENTOS**

Aos meus pais, Nilton e Graça, por todo apoio fornecido durante mais uma etapa de minha vida.

Aos meus professores e colegas de turma.

Aos meus companheiros de pesquisa, Herbert Monteiro (Garrocha), Rafael Costa, Cayo Pablo, Marcelo Giglioti e Dimitri Anjos, pelas experiências trocadas e pelos momentos de descontração no NUPERC (Núcleo de Pesquisa em Redes e Computação)

Às ex-secretárias do NUPERC, Jane e Izabel, pela paciência e presteza.

À Universidade Salvador, por ter me oferecido uma estrutura de trabalho.

A CAPES, pelo suporte financeiro.

Ao meu Orientador, Prof. Dr. Joberto Martins, pela atenção e incentivo em todos os momentos. Agradeço também pelo convite e oportunidade de trabalhar no Projeto CHESF, que me motivou ainda mais a ingressar no curso de mestrado e foi de grande importância para minha caminhada.

## RESUMO

Este trabalho tem como objetivo propor e avaliar algoritmos para a gerência de banda compartilhada entre tipos de classes e a seleção de caminhos em redes DS-TE (*DiffServ Aware MPLS Traffic Engineering*) configurada com o modelo de restrição de banda RDM (*Russian Doll Model*). A dissertação introduz inicialmente as tecnologias que suportam a Qualidade de Serviço (QoS) e a Engenharia de Tráfego (*Traffic Engineering - TE*) em redes IP multi-serviço, juntamente com a arquitetura DS-TE e seus componentes. Os algoritmos ADAPT-RDM e MinpreptDSTE são apresentados e discutidos com uma avaliação via simulação dos seus resultados para diferentes topologias e cenários de rede. A simulação apresentada faz uso de um simulador desenvolvido no escopo da dissertação, com o propósito específico de avaliar o desempenho dos algoritmos propostos. Através dos resultados obtidos observou-se que a utilização do algoritmo ADAPT-RDM é capaz de preservar as configurações do modelo RDM nos enlaces da rede, independentemente do critério de seleção de caminho de LSP utilizado e do nível de tráfego nos enlaces da rede. A seleção de caminhos para os LSPs através do algoritmo MinpreptDSTE se mostrou mais eficiente em termos da redução total da necessidade de preempções de LSPs, quando comparado com a seleção de caminhos baseada na minimização da quantidade de saltos em rede usado pelo algoritmo Constrained Shortest Path First (CSPF).

**Palavras-chave:** *DiffServ Aware-MPLS Traffic Engineering* (DS-TE). *Russian Doll Model* (RDM). Modelo de Restrição de Banda. Engenharia de Tráfego. Qualidade de Serviço. Avaliação de Desempenho de Redes de Computadores. Simulação.

## ABSTRACT

This work aims at proposing and evaluating a bandwidth sharing management algorithm among class types and a path selection algorithm for DiffServ Aware-MPLS Traffic Engineering (DS-TE) networks configured with RDM (Russian Doll Model) bandwidth constraint model. Initially, the technologies that offer Quality of Service (QoS) and Traffic Engineering (TE) in multiservice IP networks, together with the DS-TE architecture are introduced. The ADAPT-RDM and MinpreptDSTE algorithms are presented and the results of a performance evaluation done by simulation are discussed for different topologies networks scenarios. The simulation was done by a specific simulator developed in the scope of this work, with the specific proposal of evaluate the performance of the proposed algorithms. The results show that the ADAPT-RDM algorithm preserves the RDM model configuration for networks links independently of LSPs path selection criteria and network traffic level. The LSP path selection resulted by use of the MinpreptDSTE algorithm reduced the total number of LSPs preemptions, when compared with the minimum hops criteria used by the Constrained Shortest Path First (CSPF) algorithm.

**Keywords:** DiffServ Aware MPLS Traffic Engineering (DS-TE). Russian Doll Model (RDM). Bandwidth Constraint Model. Traffic Engineering. Quality of Service. Performance Evaluation of Computer Networks. Simulation.

## LISTA DE FIGURAS

Figura 1 - Detalhamento do campo DSCP do cabeçalho DiffServ (IPv4).....	22
Figura 2 - Mecanismos para implantação DiffServ.....	25
Figura 3 - Domínio com tecnologia de Serviços Diferenciados - DiffServ .....	26
Figura 4 - O Problema da Espinha de Peixe.....	28
Figura 5 - Cabeçalho MPLS .....	31
Figura 6 - Atribuição de FEC no Roteador de Fronteira - LER .....	32
Figura 7 - Processo de Encaminhamento em um Domínio MPLS (Guido, 2004) .....	33
Figura 8 - Seleção do Menor Caminho pelo Protocolo de Roteamento .....	34
Figura 9 - Distribuição de Rótulos e Construção da Tabela de Encaminhamento .....	35
Figura 10 - Cálculo do caminho de uma LSP através do CSPF .....	39
Figura 11 – Troca de mensagens RESV e PATH no protocolo RSVP-TE .....	40
Figura 12 - Exemplo com a Evolução do Conteúdo do Objeto RRO .....	42
Figura 13 - Mapeamento do DiffServ no MPLS-TE (L-LSP) .....	45
Figura 14 - Escolha das TE-classes em uma rede DS-TE (Minei e Lucek, 2005) .....	52
Figura 15 - Alocação de banda em um enlace com o modelo MAM para três CTs.....	57
Figura 16 - Exemplo de estabelecimento de LSPs em uma rede DS-TE habilitada com o modelo MAM .....	59
Figura 17 - Modelo RDM e as restrições de banda dos CTs (BCs) .....	60
Figura 18 - Estabelecimento de LSP em uma rede DS-TE habilitada com o modelo RDM....	62
Figura 19 - Ilustração da necessidade da preempção para garantir as configurações do modelo RDM.....	63
Figura 20 - Seleção do Caminho X Ocorrência de Preempção .....	70
Figura 21 - Gerência Centralizada de um Domínio DS-TE .....	73
Figura 22 - LSPs agrupadas por CT na base de dados da Entidade Gerenciadora.....	74
Figura 23 - Pseudo-código do algoritmo ADAPT-RDM .....	78
Figura 24 - Funcionamento da integração da seleção com base no menor caminho e o ADAPT-RDM .....	80
Figura 25 - Pseudo-código para uma implementação do algoritmo CSPF integrada com o ADAPT-RDM .....	82
Figura 26 - Pseudo-código da proposta de algoritmo MinPreptDSTE.....	84
Figura 27 - Máquina de Eventos do Simulador.....	88
Figura 28 - Eventos, módulos e principais etapas da simulação .....	91

Figura 29 - Topologia Utilizada para Avaliação do Algoritmo ADAPT-RDM.....	96
Figura 30 - Evolução da Largura de Banda Ocupada por CT para o Enlaces A e B Respectivamente (com A=1 e B=0) .....	100
Figura 31 - Evolução da Largura de Banda Ocupada por BC para os Enlaces A e B Respectivamente (com A=1 e B=0) .....	102
Figura 32 - Evolução da Largura de Banda Ocupada por CT para o Enlaces A e B Respectivamente (com A=1 e B=1) .....	105
Figura 33 - Evolução da Largura de Banda Ocupada por BC para os Enlaces A e B Respectivamente (com A=1 e B=1) .....	106
Figura 34 - Topologia de rede utilizada para a avaliação do algoritmo MinPreptDSTE .....	108
Figura 35 - Comparação do número de preempções entre os dois algoritmos de seleção de caminhos.....	111
Figura 36 - Evolução do Número de Preempções Ocorridas nos LSPs associados ao CT1 ..	112
Figura 37 - Evolução do Número de Preempções Ocorridas nos LSPs associados ao CT0 ..	112

## LISTA DE QUADROS

Quadro 1 - Faixa de largura de banda típica para algumas aplicações em rede .....	20
Quadro 2 - Exemplo típico de SLA para aplicação de voz sobre IP .....	20
Quadro 3 - Exemplo de Mapeamento em classes de serviço DiffServ .....	24
Quadro 4 - Mapeamento do DiffServ no MPLS-TE (E-LSP).....	44
Quadro 5 - Mapeamento do DiffServ no MPLS-TE (L-LSP).....	45
Quadro 6 - Mapeamento de Classes DiffServ em CTs DS-TE .....	49
Quadro 7 - Aplicação da preempção em redes DS-TE.....	52
Quadro 8 - Restrições de Banda do Modelo MAM para 8 CTs configuradas em rede.....	58
Quadro 9 - Restrições de Banda do Modelo RDM para 8 CTs configuradas em rede.....	61
Quadro 10 - Comparativo entre os modelo MAM e RDM.....	64
Quadro 11 - Configuração das LSPs estabelecidas .....	67
Quadro 12 - Configuração dos BCs antes e depois da nova requisição .....	67
Quadro 13 - Configurações das LSPs Pré-estabelecidas .....	70
Quadro 14 - Configurações das BCs dos Enlaces do Menor Caminho .....	71
Quadro 15 - Configurações das BCs dos Enlaces do Maior Caminho.....	71
Quadro 16 - Variáveis de Controle da Entidade Gerenciadora .....	75
Quadro 17 - Estrutura da Matriz de Caminhos para um par Origem-Destino de LSP .....	75
Quadro 18 - Configurações dos BCs para o Cenário de Simulação .....	96
Quadro 19 - Resultados de Simulação – Algoritmo ADAPT-RDM (Fatores “w(LSP)” A=1 e B=0).....	99
Quadro 20 - Resultados de Simulação – Algoritmo ADAPT-RDM (Fatores “w(LSP)” A=1 e B=1).....	104
Quadro 21 - Configurações dos BCs para o Cenário de Simulação .....	108
Quadro 22 - Resultados de Simulação – Algoritmos CSPF e MinpreptDSTE (Fatores “w(LSP)” A=1 e B=0).....	110

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	13
1.1	MOTIVAÇÃO	13
1.2	CONTIBUIÇÕES	17
1.3	ORGANIZAÇÃO DA DISSERTAÇÃO	18
<b>2</b>	<b>QUALIDADE DE SERVIÇO – CONCEITOS E TECNOLOGIAS</b>	19
2.1	INTRODUÇÃO	19
2.2	QUALIDADE DE SERVIÇO	19
2.3	A ARQUITEURA DIFFSERV	21
<b>2.3.1</b>	<b>Classes de serviço Diffserv</b>	23
<b>2.3.2</b>	<b>Mecanismos para a implantação Diffserv</b>	24
2.4	A ENGENHARIA DE TRÁFEGO	27
2.5	O MPLS	30
<b>2.5.1</b>	<b>Plano de encaminhamento do MPLS</b>	31
<b>2.5.2</b>	<b>Plano de controle do MPLS</b>	34
2.6	PLANO DE CONTROLE DO MPLS PARA APLICAÇÃO DA ENGENHARIA DE TRÁFEGO	36
<b>2.6.1</b>	<b>Informação com o estado da rede</b>	36
<b>2.6.2</b>	<b>Cálculo de caminho baseado em restrições</b>	38
<b>2.6.3</b>	<b>Estabelecimento de caminho e controle de admissão – RSVP-TE</b>	39
2.7	SUPORTE DO MPLS-TE PARA MAPEAR O DIFFSERV	43
<b>3</b>	<b>ARQUITETURA DS-TE</b>	47
3.1	INTRODUÇÃO	47
3.2	TIPOS DE CALSES (CTs)	48
3.3	PRIORIDADE E PREEMPÇÃO EM REDES DS-TE	50
3.4	POLÍTICA DE PREEMPÇÃO PARA REDES DS-TE	53
3.5	DS-TE vs MPLS-TE	54
<b>3.5.1</b>	<b>Cálculo do caminho em uma rede DS-TE</b>	55
<b>3.5.2</b>	<b>Estabelecimento do caminho em uma rede DS-TE</b>	55

3.6	MODELOS DE RESTRIÇÃO DE BANDA NAS REDES DS-TE	56
3.6.1	<b>Modelo de restrição de banda MAM</b>	56
3.6.2	<b>Modelo de restrição de banda RDM</b>	60
<b>4</b>	<b>FORMALIZAÇÃO DOS PROBLEMAS ABORDADOS</b>	<b>65</b>
4.1	GERÊNCIA DE BANDA COMPARTILHADA ENTRE CTS EM REDES DS-TE	65
4.2	OCORRÊNCIA DESNECESSÁRIA DE PREEMPÇÕES	69
<b>5</b>	<b>APRESENTAÇÃO DAS SOLUÇÕES PROPOSTAS: ALGORITMOS ADAPT-RDM, CSPF E MINPREPTDSTE</b>	<b>72</b>
5.1	CARACTERÍSTICAS GERAS DAS SOLUÇÕES PROPOSTA	72
5.2	ALGORITMO PARA GERENCIAMENTO DE BANDA COMPARTILHADA ENTRE CTS PARA O MODELO RDM	75
5.3	ALGORITMO PARA SELEÇÃO DE CAMINHOS BASEADO NO MENOR NÚMERO DE SALTOS CIENTE DO RDM	79
5.4	ALGORITMO PARA SELEÇÃO DE CAMINHOS QUE MINIMIZA A OCORRÊNCIA DE PREEMPÇÃO CIENTE DO MODELO RDM	82
<b>6</b>	<b>VALIDAÇÃO DOS ALGORITMOS</b>	<b>86</b>
6.1	O SIMULADOR	86
6.2	SIMULAÇÃO – DESCRIÇÃO	88
6.3	EVENTOS E MÓDULOS DA SIMULAÇÃO	90
6.3.1	<b>Evento 1: Geração de um novo LSP para a rede</b>	91
6.3.2	<b>Evento 2: Tentativa de estabelecimento de LSP</b>	92
6.3.3	<b>Evento 3: Desativação de um LSP</b>	93
<b>7</b>	<b>RESULTADOS DA AVALIAÇÃO DOS ALGORITMOS (ADAPT-RDM, CSPF E MINPREPTDSTE)</b>	<b>95</b>
7.1	AVALIAÇÃO DO ALGORITMO ADAPT-RDM	95
7.2	AVALIAÇÃO DE ALGORITMOS DE SELEÇÃO DE CAMINHOS BASEADOS EM RESTRIÇÕES – CSPF E MINPREPTDSTE	107
<b>8</b>	<b>CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS</b>	<b>117</b>
	<b>REFERÊNCIAS</b>	<b>119</b>
	<b>ANEXO A – CÓDIGO DAS PRINCIPAIS FUNÇÕES UTILIZADAS PARA SIMULAÇÃO</b>	<b>124</b>



# 1 INTRODUÇÃO

Este capítulo tem por objetivo apresentar a motivação para a realização desse trabalho, as contribuições identificadas após a realização, bem como uma apresentação de como está feita a organização dos demais capítulos da dissertação.

## 1.1 MOTIVAÇÃO

Ultimamente existe uma tendência cada vez maior em se pesquisar, desenvolver e prover uma infra-estrutura única de rede multiserviço (serviços de dados, voz e vídeo). Uma rede multiserviço oferece a possibilidade da redução dos custos financeiros e operacionais para o operador da rede, pois possibilita a operação de uma única rede ao invés de redes separadas para cada tipo de serviço. Essa vantagem culmina por facilitar a gerência da rede como um todo. As redes baseadas no protocolo IP surgiram com a premissa de serem utilizadas por qualquer aplicação nos diversos tipos de meios físicos e tecnologias existentes na época. Como, inicialmente, as aplicações eram meramente restritas ao tráfego de dados, a tecnologia IP não tinha a pretensão, nem a expectativa, de fazer este processo de integração de tráfego de voz, vídeo e dados que as aplicações multimídia em rede requisitam atualmente. No entanto, devido à própria penetração de mercado e popularização do protocolo IP, vem ocorrendo cada vez mais uma utilização de novos serviços que requisitam recursos que se contrapõem à filosofia inicial do protocolo IP de apenas trafegar dados na forma melhor esforço (*best effort*). O serviço melhor esforço é caracterizado pelo encaminhamento de pacotes sem o estabelecimento de conexão entre as entidades, não garantindo a entrega dos pacotes em um determinado limite de tempo, nem que eles cheguem ao destino na ordem correta, e nem mesmo garante que eles cheguem ao destino. Os roteadores fazem o melhor que podem, se “esforçam” ao máximo, mas não podem garantir a entrega dos pacotes.

Determinadas aplicações em rede como, por exemplo, as que necessitam trafegar voz e vídeo em tempo real, podem não operar com a qualidade desejada quando atendidas dessa maneira visando à satisfação do usuário final. Existe, portanto, uma necessidade mínima de requisitos de desempenho associada ao tráfego dessas mídias contínuas que, tipicamente, toleram baixa perda de pacotes, são sensíveis ao atraso e possuem limites mínimos de largura de banda.

As aplicações multimídia em rede podem ser classificadas em várias dimensões baseadas em suas características de operação de acordo com o tipo de mídia envolvida (MIRAS, 2002). As

aplicações denominadas elásticas podem suportar altas variações de largura de banda e atraso, sem que sua qualidade seja degradada. Em outras palavras, essas aplicações não são afetadas em termos de qualidade pelas flutuações de vazão e atraso causadas pela rede. O principal requisito de serviço da rede por essas aplicações é uma entrega de pacotes confiável e em ordem. Já as aplicações denominadas inelásticas, também conhecidas como aplicações de tempo-real, têm sua qualidade diretamente relacionada com alguns parâmetros de desempenho de rede (atraso, variação de atraso, variações da largura de banda e a perdas de pacotes). Tipicamente as aplicações inelásticas estão associadas ao tráfego de mídias de voz e vídeo em tempo-real. Essas aplicações não suportam altas flutuações de desempenho da rede, e se o mesmo não estiver adequado, a qualidade da aplicação pode ser inaceitável em termos de satisfação para o usuário, ou até chegar ao caso extremo da aplicação não operar.

A implantação da Qualidade de Serviço (QoS) em redes de computadores é utilizada para atender satisfatoriamente o desempenho das aplicações, sendo que ela é garantida pelos equipamentos e componentes da rede (MARTINS *et al.*, 2003). Os parâmetros de desempenho das aplicações são também conhecidos como parâmetros de QoS, e geralmente intervalos de valores aceitáveis são definidos para estes parâmetros (Ex: atraso fim-a-fim entre 100 e 200 ms), de forma que dentro desse intervalo as aplicações envolvidas operam satisfatoriamente.

Portanto, redes mutiserviço requerem serviços com garantias de QoS fim-a-fim (MARTINS *et al.*, 2003; ZHANG *et al.*, 2007) e, nesse contexto, aplicações de técnicas adicionais à proposta inicial do IP foram feitas no intuito de tornar a rede IP capaz de suportar as demandas requisitadas pelos diferentes tipos de aplicações em rede.

Os Serviços Diferenciados (*DiffServ*) (BLAKE *et al.*, 1998) é uma arquitetura que foi recomendada com a intenção de que os diferentes requisitos de QoS de distintas aplicações em rede fossem atingidos. A proposta do *DiffServ* consiste em agregar os fluxos de tráfego com as mesmas características de QoS em classes de serviço através de uma classificação e marcação prévia de pacotes realizada nos roteadores de fronteira da rede. Os fluxos de tráfego associados às classes de serviço mais exigentes possuem tratamento diferenciado no processo de encaminhamento de pacotes ao longo da rede. Apesar de promover a seleção de quais fluxos de tráfego exigem maiores recursos da rede, a utilização apenas da alternativa *DiffServ* sozinha não é capaz de assegurar que os parâmetros de QoS serão cumpridos de forma fim-a-

fim na rede. O *DiffServ* assume que há recursos suficientes nos enlaces da rede para o encaminhamento dos pacotes de todas as aplicações, considerando apenas o encaminhamento salto-a-salto e desconsiderando os recursos disponíveis no caminho ao qual o fluxo de tráfegos seguirá fim-a-fim na rede. A possibilidade da existência de congestionamento na rede no instante do encaminhamento dos pacotes em qualquer um dos saltos do caminho entre o roteador de origem e o roteador de destino pode fazer com que a QoS esperada não seja atingida.

Para o melhor aproveitamento dos recursos e roteamento adequado dos fluxos de tráfego em rede fez-se necessário também a aplicação da técnica de Engenharia de Tráfego (ET). A Engenharia de Tráfego é proposta na tentativa de lidar com a complexidade resultante de requisitos e conflitos existentes quando os recursos de rede (largura de banda dos enlaces) são um fator limitante (AWDUCHE, 1999b; AWDUCHE *et al.*, 2002). A Engenharia de Tráfego torna possível o roteamento dos fluxos de tráfego levando em conta um mapeamento adequado das requisições de fluxos de tráfego nos recursos existentes da rede, visando, dentre outros objetivos, a minimização da ocorrência de congestionamento.

O *Multi Protocol Label Switchin* (MPLS) (ROSEN *et al.*, 2001) é uma arquitetura inicialmente criada para a otimização de desempenho no processamento dos roteadores utilizados a época da sua recomendação inicial. Essa otimização é conseguida através do encaminhamento de pacotes IP com base na utilização de um rótulo (*label*) que agrega informações de encaminhamento nos roteadores de borda da rede, reduzindo assim o tempo de consulta nas tabelas de rotas nos roteadores do núcleo da rede.

O MPLS tornou-se com o passar do tempo uma alternativa tecnológica interessante para a aplicação da Engenharia de Tráfego nas redes IP. A extensão da tecnologia MPLS para a aplicação da Engenharia de Tráfego é conhecida na literatura como Protocolo MPLS ciente da Engenharia de Tráfego (MPLS-TE) (AWDUCHE *et al.*, 1999a). O princípio básico é que através de caminhos previamente estabelecidos, denominados *Label Switched Paths* (LSPs), é possível reservar recursos para um determinado fluxo de tráfego em rede fazendo com que a QoS possa ser atingida fim-a-fim na rede. Dessa forma é possível minimizar a ocorrência de congestionamentos.

Na recomendação original da Engenharia de Tráfego com o MPLS-TE, a largura de banda dos enlaces da rede é gerenciada de maneira agregada. A partir de um roteador de origem, os pacotes assinados com um mesmo rótulo (associados a um mesmo LSP), e com direção ao mesmo roteador de destino, são roteados coletivamente de acordo com uma única restrição de banda considerando a banda do enlace como um todo (MINEI; LUCEK, 2005; ZHANG *et al.*, 2007). O fato da gerência da largura de banda não considerar a possibilidade de existência de distintas classes de serviço torna o MPLS-TE pouco eficiente em um ambiente multiclasse *DiffServ*.

Uma solução para a integração dos mecanismos de provisionamento QoS do *DiffServ* com a possibilidade da aplicação simultânea da técnica de Engenharia de Tráfego ao nível de classes de serviço, é a arquitetura *DiffServ aware MPLS Traffic Engineering* (DS-TE) proposta pelo *Internet Engineering Task Force* (IETF) em (LE FAUCHER; LAI, 2003). No DS-TE o cálculo da banda disponível nos enlaces da rede pode ser realizado considerando a banda de diferentes tipos de classes (CTs) configurados em rede (ao invés de considerar a banda do enlace como um todo). Essa característica do DS-TE permite o fornecimento de garantias mais rigorosas de QoS em paralelo com uma maior otimização do uso dos recursos da rede. Na introdução da recomendação do DS-TE (LE FAUCHER; LAI, 2003) são listados três tipos de redes que podem se beneficiar das características rigorosas de QoS que podem ser oferecidas pela arquitetura DS-TE:

- a) redes onde a largura de banda seja um recurso escasso;
- b) redes com quantidade significativa de aplicações com tráfego sensível ao atraso;
- c) redes onde a proporção relativa de tráfego entre diferentes tipos de classes (CTs) configurados em rede não seja uniforme.

Uma das exigências da arquitetura DS-TE para o cálculo da banda disponível por CT é a necessidade da adoção da alocação de banda por classe de serviço, que é realizada através de um modelo de restrição de banda (LE FAUCHER; LAI, 2003). O modelo de restrição de banda define os limites e regras adotadas para a utilização da largura de banda dos enlaces de uma rede DS-TE, para cada uma dos diferentes CTs configurados na rede. Os limites e regras para a utilização da banda nos enlaces de uma rede DS-TE são denominado de requisitos ou restrições de banda *Bandwidth Constraints* (BCs). Para prevenir a degradação de QoS ao nível de pacotes, o número de LSPs associados a um determinado CT deve estar em consonância com a restrição de banda (BC) (LAI, 2005). As BCs são utilizadas para

determinar o número de LSPs que podem estar simultaneamente estabelecidos para diferentes CTs em diferentes condições operacionais (diferentes cargas de tráfego na rede).

Dois aspectos de interesse das redes DS-TE são abordados neste trabalho: a escolha de caminhos para novas requisições de LSPs que chegam à rede e a possibilidade de compartilhamento de banda entre diferentes tipos de classes (CTs), quando especificamente o modelo de restrição de banda *Russian Dolls Model* (RDM) (LE FAUCHER, 2005a) é utilizado nos enlaces da rede.

A escolha do caminho para uma nova requisição de LSP deve levar em consideração a existência de recurso (banda) disponível no tipo de classe (CT) ao qual o novo LSP está associado, de acordo com restrições impostas pelo modelo de restrição de banda utilizado. A possibilidade de compartilhamento de banda entre CTs é de interesse para que uma maior eficiência na utilização dos recursos da rede seja atingida.

## 1.2 CONTRIBUIÇÕES

A seguir são identificadas as contribuições deste trabalho de mestrado:

- a) uma proposta e implementação de algoritmo (PINTO NETO; MARTINS, 2008a; PINTO NETO; MARTINS, 2008b) com uma estratégia de gerência de banda compartilhada entre CTs em conformidade com as configurações do modelo de restrição de banda RDM;
- b) uma implementação do algoritmo de cálculo do menor caminho baseado em restrições *Constrained Shortest Path First* (CSPF) para uma rede multiclasse, que utiliza a estratégia de gerência de banda compartilhada proposta e implementada para se adequar ao modelo de restrição de banda RDM;
- c) uma proposta e implementação de algoritmo (PINTO NETO *et al.*, 2008) para o cálculo de caminho que visa minimizar a ocorrência de preempções em uma rede DS-TE utilizando o modelo de restrição de banda RDM;
- d) a implementação em linguagem C de um simulador de propósito específico para avaliar o comportamento dos algoritmos propostos; e
- e) a avaliação de desempenho dos algoritmos propostos a partir dos resultados dos cenários de simulação propostos.

### 1.3 ORGANIZAÇÃO DA DISSERTAÇÃO

O restante desse trabalho de dissertação está dividido da seguinte forma:

- a) no capítulo 2 os conceitos de Qualidade de Serviço (QoS) e Engenharia de Tráfego *Traffic Engineering* (TE) são detalhados. A arquitetura e as características de funcionamento de duas alternativas tecnológicas estabelecidas para o fornecimento de QoS e TE, o DiffServ e o MPLS-TE, são apresentadas;
- b) no capítulo 3 são apresentados os objetivos, a arquitetura e os mecanismos de funcionamento do DS-TE;
- c) no capítulo 4 é feita uma revisão da literatura e a apresentação de problemas encontrados;
- d) no capítulo 5 são apresentadas propostas de algoritmos que procuram solucionar os problemas descritos no capítulo anterior;
- e) no capítulo 6 é apresentada a forma como a proposta de simulador de propósito específico foi implementado para avaliar as propostas de algoritmos;
- f) no capítulo 7 são apresentados os resultados de simulação e as análises de resultados de simulação referentes aos cenários idealizados para a avaliação de desempenho dos algoritmos propostos;
- g) no capítulo 8 são apresentadas as considerações finais sobre o trabalho e são indicados cenários possíveis para trabalhos futuros.

## 2 QUALIDADE DE SERVIÇO: CONCEITOS E TECNOLOGIAS

Neste capítulo serão apresentados os conceitos de Qualidade de Serviço e Engenharia de Tráfego. Um resumo das características das soluções Arquitetura de Serviços Diferenciados (DiffServ) e *Multiprotocol Label Swiching - Traffic Engineering* (MPLS-TE) importantes para o contexto histórico e entendimento da arquitetura *DiffServ Aware MPLS Traffic Engineering* (DS-TE) também serão apresentados.

### 2.1 INTRODUÇÃO

A estratégia de utilização do protocolo *Internet Protocol* (IP), juntamente com tecnologias adicionais, vem sendo utilizada como uma solução capaz de criar uma plataforma capaz de suportar uma rede multi-serviço em uma infra-estrutura única e eficiente.

A utilização de tecnologias adicionais para o suporte a redes multi-serviço deve-se ao fato de que a premissa inicial do IP era suportar o tráfego de dados da melhor maneira possível (*Best Effort*), sendo que o protocolo não foi projetado para o tráfego de mídias contínuas como voz e vídeo, principalmente quando a característica de tempo-real deve ser considerada.

Para possibilitar a integração de diversos tipos de tráfego (dados, voz e vídeo) e serviços cada vez mais exigentes das novas aplicações (Videoconferência, VoIP, etc.) que surgiram ultimamente em uma arquitetura de rede multi-serviço única, foram desenvolvidos mecanismos que introduziram os conceitos de suporte à Qualidade de Serviço (MARTINS, 1999) e Engenharia de Tráfego (AWDUCHE *et al.*, 2002) nas redes de *backbones* baseadas na tecnologia IP.

### 2.2 QUALIDADE DE SERVIÇO

A Qualidade de serviço (QoS) é definida pelo *International Telecommunication Union – Telecommunications* (ITU-T) na recomendação I.350 como: "o efeito coletivo de performance que determina o grau de satisfação do usuário deste serviço específico". Mais especificamente, em redes de computadores, a QoS é utilizada para definir o desempenho de uma rede relativa às necessidades das aplicações, como também o conjunto de tecnologias

que possibilitam às redes de computadores oferecerem garantias de desempenho. Segundo (MARTINS *et al.*, 2003), a QoS é um requisito da(s) aplicação(ões) para a qual exige-se que determinados parâmetros de QoS (largura de banda, atraso, *jitter* e perda de pacotes) estejam dentro de limites bem definidos (valor mínimo, valor máximo) para o desempenho satisfatório das mesmas em rede.

O quadro 1 a seguir apresenta a faixa de largura de banda típica requerida por algumas aplicações em rede:

<b>Aplicação</b>	<b>Faixa de Largura de Banda</b>
Aplicações Transacionais	1 kbps a 50 kbps
Quadro Branco ( <i>Whiteboard</i> )	10 kbps a 100 kbps
Voz	10 kbps a 120 kbps
Aplicações Web (WWW)	10 kbps a 500 kbps
Transferência de Arquivos (Grandes)	10 kbps a 1 Mbps
Vídeo ( <i>Streaming</i> )	100 kbps a 1 Mbps
Aplicação Conferência	500 kbps a 1 Mbps
Vídeo MPEG-2	1 Mbps a 10 Mbps
Aplicação Imagens Médicas	10 Mbps a 100 Mbps
Aplicação Realidade Virtual	80 Mbps a 150 Mbps

Quadro 1: Faixas típicas de largura de banda para algumas aplicações em rede (Martins 2003):

Os parâmetros de qualidade de serviço associados a cada tipo de aplicação são, normalmente, especificados previamente em um contrato ou solicitação de serviço denominado de *Service Level Agreement* (SLA) (TRIMINTZIOS *et al.*, 2001). Um SLA deve definir claramente quais requisitos devem ser garantidos para que a aplicação possa executar com qualidade. Um exemplo de SLA para uma aplicação de voz em tempo-real é mostrado no quadro 2:

<b>Parâmetro</b>	<b>Especificação</b>
Perda	$\leq 10 \%$
Atraso	$\leq 200$ mseg
<i>Jitter</i>	$\leq 40$ mseg

Quadro 2 - Exemplo típico de SLA para aplicação de voz sobre IP (Elaborado pelo autor, 2008)

Devido à característica de multiplexação estatística<sup>1</sup> presente nas redes IP, e como os pacotes IP entram em um *backbone* de rede de forma assíncrona, existe uma potencial disputa por recursos (banda) entre as aplicações em rede. Se dois pacotes de aplicações diferentes entram em um roteador ao mesmo tempo (a partir de duas interfaces de entrada diferentes) e são destinados à mesma interface de saída, isso significa uma disputa por recursos, sendo que um dos pacotes tem que esperar (tempo de enfileiramento ou tempo de espera em *buffer*) até que o outro seja transmitido. Em uma situação extrema, pacotes começarão a serem descartados, caso a taxa de chegada de pacotes que chegam pelas interfaces de entrada de determinado roteador seja maior do que a capacidade de transmissão das interfaces de saída. Alguns tipos de tráfego, como por exemplo, a transferência de dados em massa (Ex: FTP), lidam muito bem com o atraso em *buffer*, enquanto outras, como voz e vídeo em tempo-real (comunicação iterativa) não (MIRAS, 2002).

A partir das necessidades de serviço das aplicações em rede é possível configurar a melhor forma de se empregar alguma alternativa técnica visando o provisionamento de QoS para as aplicações, no caso tentando respeitar os requisitos especificados. O capítulo a seguir apresenta a arquitetura geral da alternativa técnica *DiffServ* (BLAKE *et al.*, 1998) especificada para suportar QoS em redes IP.

### 2.3 A ARQUITETURA DIFFSERV

A arquitetura *DiffServ* (Arquitetura de Serviços Diferenciados) fornece mecanismos adicionais à tecnologia IP (além do serviço melhor esforço convencional), tornando possível o tratamento diferenciado de pacotes a fim de lidar com as diferentes necessidades (demandas) das aplicações em rede.

O *DiffServ* (BLAKE *et al.*, 1998) é uma arquitetura de rede criada para atender às várias necessidades de QoS das aplicações, particularmente, as aplicações multimídia, sendo um mecanismo altamente flexível e de grande escalabilidade. Isso deve-se ao fato do *DiffServ* não trabalhar com o roteamento baseado em conexões individuais como outras tecnologias de núcleo de rede, como o *Asynchronous Transfer Mode* (ATM) e os *Serviços Integrados*

---

<sup>1</sup> A multiplexação estatística é uma técnica de multiplexação na qual a largura da banda é alocada conforme a necessidade. Nas redes que utilizam a multiplexação estatística, diferentes canais de comunicação compartilham os recursos físicos e a capacidade de transmissão, ocupando-os dinamicamente em função da demanda e da garantia mínima desejada.

(IntServ) (SHENKER *et al.*, 1997), e sim com agregados de fluxos de tráfego que são agrupados em um número limitado de classes de serviço. Dessa forma a quantidade de informações a serem armazenadas e processadas é reduzida, já que o número de estados armazenados pelos roteadores ao longo do caminho não cresce proporcionalmente ao número de conexões individuais de cada usuário.

No *DiffServ*, os pacotes são marcados, em um campo específico do pacote IP, de forma a ser possível a criação de distintas classes de serviço, a fim de possibilitar a identificação e o tratamento diferenciado (prioritário) para as aplicações que possuam requisitos de QoS diferenciados para a sua operação. Esta marcação é feita no cabeçalho do pacote IP nos roteadores de fronteira de acordo com a classe do tráfego, através de um campo de 6 bits denominado *Differentiated Service Code Point* (DSCP) (Figura 1). Os dois bits após o campo DSCP correspondem ao campo *Explicit Congestion Notification* (ECN), definido em (RAMAKRISHNAN *et al.*, 2001), que é utilizado para o controle de congestionamento. No IPv4 utiliza-se três bits do campo *Type of Service* (TOS) e o campo *Precedence* para a marcação do DSCP (Figura 1). No IPv6 utiliza-se o campo "*Traffic Class*". Estando os pacotes das aplicações identificados (marcados), diferentes níveis de serviço podem ser oferecidos para essas aplicações ao longo dos roteadores que compõem o núcleo da rede, a depender da prioridade do tráfego e critérios do projetista.

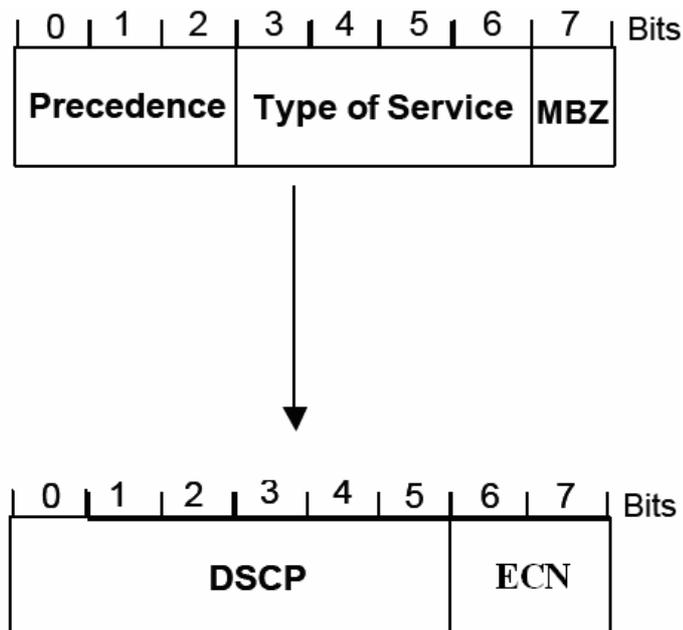


Figura 1 - Detalhamento do campo DSCP do cabeçalho DiffServ (IPv4) (Elaborado pelo autor, 2008)

### 2.3.1 Classes de serviço Diffserv

A partir das definições dos requisitos de QoS das aplicações pode ser feito o agrupamento das mesmas em classes de serviço DiffServ. Uma classe de serviço DiffServ está associado a um comportamento descrito por um *Per Hop Behavior* (PHB) no roteador. O PHB (TRIMINTZIOS *et al.*, 2001) define a maneira como o roteador irá tratar cada pacote IP rotulado que passa por ele. Os roteadores que implementam o DiffServ associam o fluxo de tráfego marcado com determinado valor DCSP ao do PHB correspondente. Em outras palavras, um PHB corresponde efetivamente ao tipo de serviço que será aplicado a determinado fluxo de tráfego previamente marcado através do rótulo DCSP. Outra maneira de perceber o conceito de comportamento (PHB) corresponde a considerá-lo como um serviço de encaminhamento de pacotes oferecido pelo DiffServ.

O fornecimento de um determinado serviço está diretamente relacionado aos mecanismos DiffServ apresentados na subseção a seguir. Considerando que o DSCP possui 6 bits, é possível um total de  $2^6$  tipos de serviço distintos. Entretanto na prática geralmente não são utilizados todos os serviços com as suas classes associadas possíveis. Do ponto de vista prático e considerando a padronização prevista nas RFCs do DiffServ, os seguintes serviços (PHBs) são definidos:

- a) *Expedited Forwarding* (EF) ou Encaminhamento Expresso (JACOBSON *et al.*, 1999): O EF oferece um serviço com baixa perda, mínimo atraso, mínimo *jitter* e banda garantida. É indicado para aplicações sensíveis ao atraso, com características de tempo real.
- b) *Assured Forwarding* (AF) ou Encaminhamento Assegurado (HEINANEN *et al.*, 1999): O AF é formado por quatro classes individuais de serviço. Estas possuem subclasses com três níveis de precedência em relação ao descarte de pacotes. É indicado para aplicações que precisam de um serviço confiável, que suporte perdas e que não sejam sensíveis ao atraso.
- c) Melhor Esforço *Best Effort* (BE): O melhor esforço é o serviço padrão da Internet, que não oferece nenhum tratamento diferenciado aos seus pacotes, e por consequência, não suporta restrições de banda, atraso ou *jitter*.

O encaminhamento expresso, EF-PHB, garante ao usuário uma banda, enquanto o encaminhamento assegurado, AF-PHB, também provê aos usuários uma banda, mas não garante a sua disponibilidade. Entretanto, com base em sua filosofia estatística de provisionamento pode obter resultados interessantes em relação ao aproveitamento da banda com atrasos e perdas mínimas.

O serviço de Encaminhamento Assegurado não se baseia em uma garantia estrita, mas em uma expectativa de serviço que será obtida por um determinado tráfego quando existem momentos de congestionamento. Um perfil associado a cada tráfego define o serviço esperado por um determinado fluxo de pacotes. Neste tipo de serviço, existe uma garantia que os pacotes marcados serão encaminhados com uma alta prioridade, contanto que o tráfego não exceda à taxa contratada, definida pelo perfil de serviço. No entanto, é permitido que um usuário exceda um perfil contratado, sabendo que o tráfego em excesso não será entregue com a mesma probabilidade do tráfego que está dentro do perfil.

Um exemplo de agrupamento em classes para um conjunto de aplicações em determinada rede é mostrado na quadro 3:

<b>Classe de Serviço</b>	<b>Aplicação</b>	<b>DSCP</b>
EF	Voz sobre IP (VoIP) Vídeo-Conferência (vídeo interativo)	"101110"
AF1	Vídeo MPEG Vídeo sob Demanda (VoD – <i>video on demand</i> ) ( <i>streaming</i> ) Sinalização	"001010"
AF2	Roteamento (gerência da roteadores) Gerência de Redes Web - HTTP	"001100"
AF3	FTP - <i>File Transfer Protocol</i> Aplicações TCP (outras)	"001110"
BE	Demais Aplicações de Dados	"010010"

Quadro 3 - Exemplo de Mapeamento em classes de serviço DiffServ  
(Elaborado pelo autor, 2008)

### 2.3.2 Mecanismos para a implantação Diffserv

O DiffServ possui um conjunto de mecanismos a serem implantados e configurados pelo gerente da rede, de forma a tornar possível o fornecimento dos diferentes tipos de serviços

(PHBs) descritos. A Figura 2 apresenta o conjunto de mecanismos existentes e utilizados no DiffServ (BLAKE *et al.*, 1998; MARTINS *et al.*, 2003) no roteador de borda da rede.

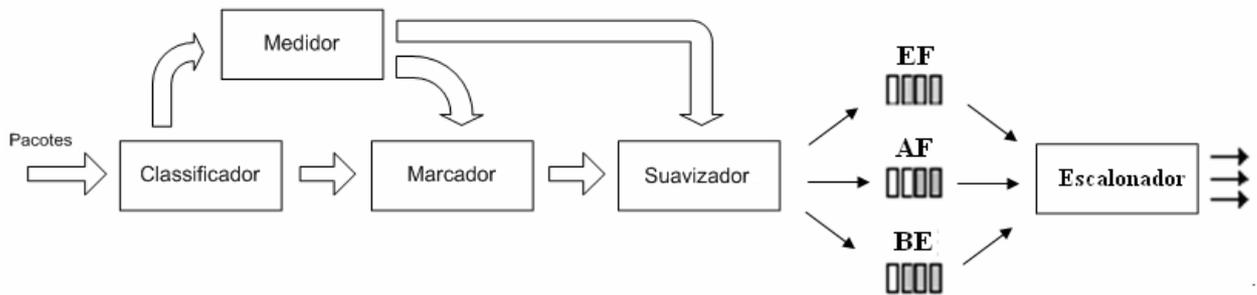


Figura 2 - Mecanismos para implantação DiffServ (Elaborado pelo autor, 2008)

São responsabilidades funcionais dos mecanismos ilustrados na Figura 2:

- a) **classificador:** O classificador é um mecanismo utilizado para classificar os pacotes que chegam ao roteador, de acordo com a classe de tráfego. Esse critério de classificação pode ser feito de acordo com alguns campos do cabeçalho IP como, por exemplo, número da porta e endereço de destino. A classificação prévia permite que posteriormente os pacotes tenham um tratamento diferenciado nas filas de saída dos roteadores;
- b) **marcador:** Ao ser classificado pelo mecanismo de classificação, o pacote é marcado em seu rótulo DSCP de acordo com a especificação definida pelo gerente da rede (classes);
- c) **medidor:** O medidor é utilizado para medir a vazão do tráfego dos pacotes associado a cada uma das classes de serviço que estão sendo tratadas pelo roteador;
- d) **suavizador (*Shapper*):** A suavização (ou regulação) de tráfego funciona como se desse “forma” (modelasse) a uma determinada classe de tráfego, para que os pacotes se mantenham em uma determinada vazão acordada. Isso evita que uma classe de serviço que contenha fluxos de tráfego mal comportados (fora das especificações do SLA) comprometa o fluxo de tráfego de outras classes de serviço. Geralmente um suavizador é implementado em classes de serviço onde haja a necessidade da regulação de tráfego. O suavizador mantém o tráfego regular, impedindo que aconteçam rajadas de pacotes no roteador. No caso mais extremo, quando não é possível a suavização devido a uma rajada de pacotes elevada (que dure um grande intervalo de tempo), os mesmos poderão ser descartados;

- e) **escalador:** O escalador é o elemento presente em equipamentos roteadores que define a ordem de atendimento das filas de saída do roteador, e, conseqüentemente, a largura de banda a ser atribuída/allocada a cada classe de serviço de acordo com as expectativas e parâmetros de QoS que foram especificadas. Em outras palavras, o escalador é o elemento que retira os pacotes enfileirados na interface de saída (buffers) dos roteadores para serem enviados através do enlace de saída. Efetivamente, o escalador de pacotes na fila de saída é o elemento funcional que provê o suporte para a implantação da qualidade de serviço (QoS) através dos seus parâmetros de banda, atraso, *jitter* e perdas de pacotes.

De acordo com a especificação do IETF para os serviços DiffServ (BLAKE *et al*, 1998), a implementação das classes de serviços, e a associação das aplicações alvo da rede nas mesmas é flexível, e depende do projetista da rede. Para atingir os parâmetros de QoS associados a cada uma das classes de serviço os mecanismos de QoS (classificador, marcador, medidor, suavizador e escalador) devem ser utilizados. Vale ressaltar que para atingir tais objetivos, nem sempre é necessário a utilização de todos os mecanismos de QoS descritos e, neste sentido, faz-se necessário um projeto específico para tal.

Pode-se exemplificar a operação dos roteadores habilitados com DiffServ, com base na Figura 3.

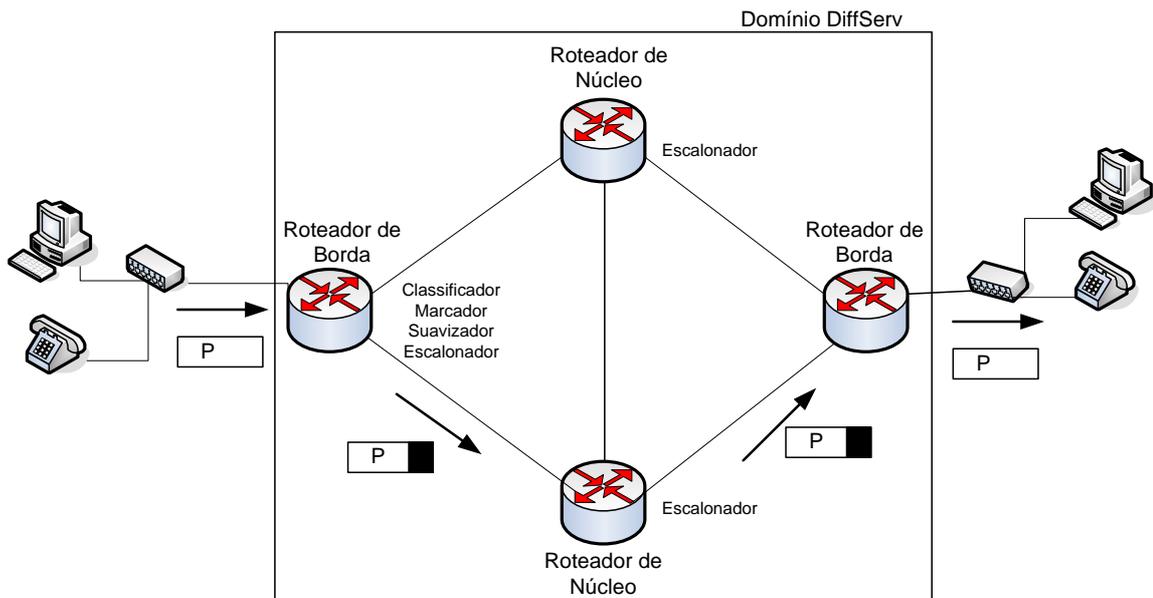


Figura 3 - Domínio com tecnologia de Serviços Diferenciados – DiffServ (Elaborado pelo autor, 2008)

Um domínio DiffServ consiste em um conjunto de roteadores habilitados a operar com a arquitetura DiffServ. Dentro do domínio DiffServ os roteadores podem ser divididos em:

- a) roteadores de borda (*edge router*): Cujos mecanismos DiffServ configurados têm, tipicamente, a função de classificar, marcar, medir e descartar pacotes de entrada não conformes. Uma vez condicionados os pacotes de entrada, os mesmos são escalonados da mesma forma que nos roteadores de núcleo;
- b) roteadores de núcleo (*core router*): Cujos mecanismos DiffServ configurados têm, tipicamente, a função de encaminhar os pacotes de acordo com a classificação e marcação feitas no roteador de borda, dando tratamento diferenciado aos serviços de acordo com uma determinada política de escalonamento de pacotes determinada pelo projetista/gerente da rede.

#### 2.4 A ENGENHARIA DE TRÁFEGO

A utilização do DiffServ não é uma condição suficiente para o bom funcionamento das aplicações em rede. A arquitetura DiffServ sozinha não é capaz de prover nenhuma garantia quanto ao congestionamento nem uso eficiente dos recursos da rede (MINEI; LUCEK, 2005). A garantia dos parâmetro de QoS das aplicações em rede apenas com a utilização da arquitetura DiffServ, só é válida caso sempre existam recursos suficientes para servir os requisitos especificados por cada uma das classes de serviço configuradas em cada um dos enlaces da rede.

A maioria dos algoritmos de roteamento convencionais utilizados nas redes fundamentam-se na escolha de rotas que minimizem métricas, como número de saltos ou o custo total entre a origem e o destino de um fluxo de dados. Essa abordagem não trata da minimização da ocorrência de congestionamento na rede. Em princípio, o problema do congestionamento poderia ser resolvido com o superdimensionamento dos enlaces da rede, só que essa estratégia não é a mais interessante em termos econômicos. Caso o tráfego associado a uma determinada classe de serviço siga um caminho já congestionado, o mesmo pode experimentar diferentes atrasos e *Jitter* em relação ao que foi especificado no SLA (TRIMINTZIOS *et al.*, 2001) ou até ser descartado em casos extremos.

A Figura 4 ilustra um problema típico (problema da espinha de peixe) onde o emprego dos protocolos convencionais de roteamento *Internet Gateway Protocol* (IGP) como o Open

*Shortest Path First* (OSPF) e o (*Intermediate System-to-Intermediate System* (IS-IS) (KUROSE; ROSS, 2006; TANENBAUM, 2003) podem ser ineficientes, por não considerarem técnicas de Engenharia de Tráfego:

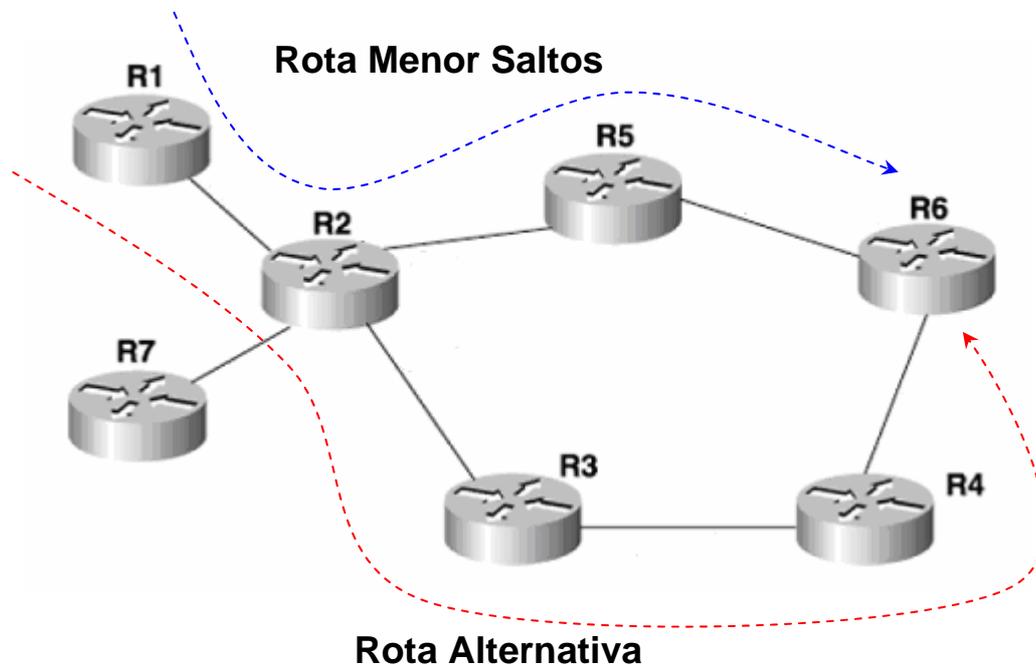


Figura 4 - O Problema da Espinha de Peixe (Elaborado pelo autor, 2008)

A rota selecionada com menor quantidade de saltos entre os roteadores R1-R6 pode vir a estar com os seus enlaces congestionados, com a consequente possibilidade da degradação da qualidade de serviço (perda de pacotes, *elevado atraso e jitter*) nas filas dos roteadores. A seleção de uma rota alternativa, mesmo que percorrendo um caminho mais longo, pode evitar o congestionamento e, nesse sentido, melhorar a QoS geral dos fluxos de dados entre R1-R6.

A principal motivação para a utilização da Engenharia de Tráfego vem das próprias limitações dos protocolos utilizados para o roteamento no mundo IP. A seguir são listadas algumas limitações dos algoritmos de roteamento convencionais que podem acarretar em problemas:

- a) uma baixa eficiência na utilização dos recursos da rede devido a caminhos alternativos serem desconsiderados. A rota a ser seguida será sempre a que possuir uma menor quantidade de saltos, mesmo que haja caminhos alternativos ociosos com disponibilidade de recursos. Do ponto de vista de QoS, nem sempre o caminho mais curto é o caminho que apresenta o melhor conjunto de recursos necessários a uma aplicação. Como a largura de banda existente nos enlaces não é considerada no roteamento convencional, não é possível saber se haverá recursos suficientes para

atender à demanda de tráfego no caminho escolhido. Recursos insuficientes geram a ocorrência de congestionamento implicando em degradação da QoS (perda de pacotes, aumento do atraso e *jitter*). O tráfego associado a cada uma das classes de serviço pode estar ineficientemente mapeado nos recursos existentes (partes da rede subutilizadas enquanto outras superutilizadas), havendo assim uma conseqüente degradação dos serviços, até mesmo quando existe a disponibilidade de recursos (Awduche *et al*, 1999a). A possibilidade de congestionamento é ainda maior em redes onde ocorre uma frequente mudança de topologia e demandas de tráfego, por exemplo, redes que lidam com o tráfego de voz e vídeo em tempo-real (MEDHI; PIÓRO, 2004) (ROUGHAN *et al*, 2003) (MIRAS, 2002);

- b) não garantir que os caminhos seguidos por fluxos de dados tenham certas características (por exemplo, que não utilizem enlaces com alta latência), garantir que os recursos de transmissão estarão disponíveis ao longo de um caminho específico, e a determinação de qual tráfego terá prioridade em momentos de indisponibilidade dos enlaces
- c) cenários de recuperação em caso de falhas nos enlaces da rede não são planejados para assegurar a continuidade dos serviços e não há garantias de que a capacidade para a restauração de serviços esteja disponível

O oferecimento de serviços que sejam confiáveis e utilizem eficientemente os recursos disponíveis nas *redes* IP sugerem a adoção da técnica de Engenharia de Tráfego (AWDUCHE *et al.*, 1999a). A Engenharia de Tráfego está ligada à otimização do desempenho global da rede e à utilização eficiente de seus recursos. Através da Engenharia de Tráfego é possível um mapeamento adequado dos fluxos de tráfego na estrutura física (topologia) da rede (MINEI; LUCEK, 2005).

Dentre os benefícios da Engenharia de Tráfego, é possível citar:

- a) a redução dos pontos de congestionamento na rede;
- b) o re-roteamento rápido de fluxo de dados em caso de falhas nos enlaces da rede;
- c) a redução dos custos, devido à utilização mais eficiente da largura de banda disponível nos enlaces;
- d) a melhoria geral da QoS, proporcionada pela redução do atraso e *Jitter*, e a redução das perdas.

A Engenharia de Tráfego pode estar associada a um controle centralizado ou distribuído (AWDUCHE, 1999b). No controle centralizado, existe uma unidade gerenciadora central que tem a visão total das configurações da rede responsável pela determinação das rotas e plano de roteamento. É esta unidade gerenciadora central que é a responsável pela coleta de informações do estado da rede, cálculo de rotas e que determinará para todos os roteadores as configurações de roteamento. No controle distribuído, cada roteador de uma forma independente seleciona as rotas dos fluxos que passam por ele baseado na sua visão do estado da rede, obtido por alguma medição que é feita, tipicamente baseada nas informações dos protocolos de roteamento do tipo estado de enlace.

## 2.5 O MPLS

Uma alternativa tecnológica para promover a Engenharia de Tráfego nas redes IP que vem sendo adotada ultimamente é o protocolo *Multiprotocol Label Switching* (MPLS) (AWDUCHE *et al.*, 1999a). Essa tecnologia é denominada multiprotocolo (MINEI; LUCEK, 2005), pois pode ser empregada de forma independente dos protocolos de camada de rede e enlace. Em princípio, qualquer tecnologia de camada de enlace pode ser empregada (Ethernet, Redes Ópticas, PPP, ATM, *Frame Relay*), sendo que para a camada de rede, o protocolo IP é a tecnologia difundida no mercado.

A proposta inicial do protocolo MPLS (ROSEN *et al.*, 2001) não era promover a Engenharia de Tráfego, e sim a otimização de desempenho no processamento dos roteadores utilizados à época da sua recomendação inicial. Através da utilização de rótulos (*labels*) para agregar informações de encaminhamento nos roteadores de borda da rede, reduziu-se o tamanho das tabelas de rotas nos roteadores do núcleo da rede. Uma quantidade grande de buscas em uma tabela de rotas IP (*Table Lookup*) podia ser reduzida a uma busca indexada na tabela de rótulos MPLS, de forma a definir para qual interface de saída o pacote seria encaminhado.

Com o MPLS tornou-se possível a adição do encaminhamento orientado à conexão ao mundo IP convencional, estabelecendo caminhos comutados por rótulos denominados *Label Switched Paths* (LSPs) entre dois nós fim-a-fim em uma rede. Os LSPs são semelhantes aos circuitos virtuais das tecnologias de nível 2, com uma menor complexidade e maior compatibilidade com o mundo IP. A possibilidade de um pré-estabelecimento de um caminho fim-a-fim (rota explícita) entre dois nós antes do envio de pacotes tornou o MPLS uma

ferramenta para a aplicação da Engenharia de Tráfego nas redes IP. A extensão da tecnologia MPLS utilizada como aplicação para a Engenharia de Tráfego é conhecido na literatura como MPLS-TE (MINEI; LUCEK, 2005).

### 2.5.1 Plano de encaminhamento do MPLS

O plano de encaminhamento do MPLS é o responsável pelo envio do pacote à interface de saída correspondente de cada roteador do Domínio MPLS. Estruturalmente, o cabeçalho MPLS, conhecido também como *Shim Header*, é colocado entre os cabeçalhos das camadas de enlace e rede. O encaminhamento dos pacotes em uma rede MPLS é feita com base em um identificador denominado rótulo MPLS. O cabeçalho MPLS é composto por quatro campos básicos, como pode ser visto na Figura 5:

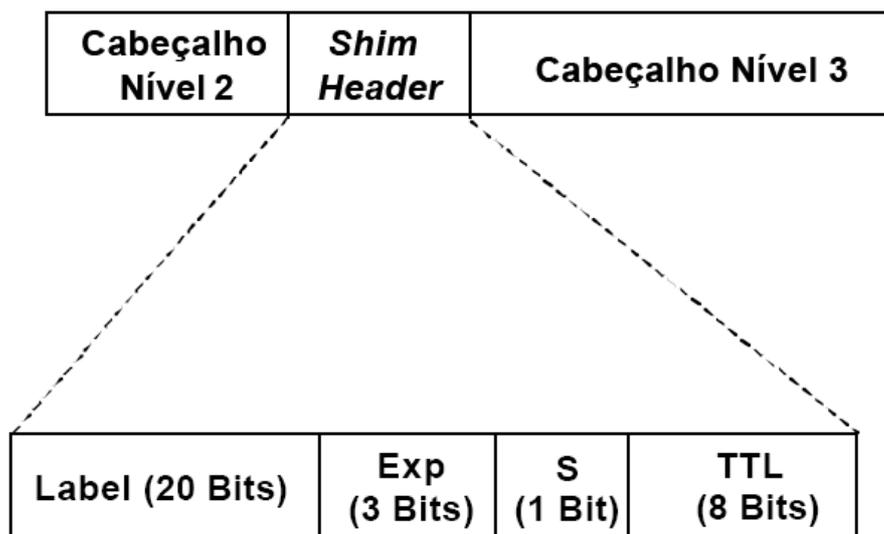


Figura 5 - Cabeçalho MPLS (Elaborado pelo autor, 2008)

- a) o campo “rótulo” (*label*) é um identificador de 20 bits que serve como um índice para a tabela de encaminhamento de pacotes. A base de dados de rótulos em cada roteador MPLS é denominada de base de informação de rótulo (LIB – *Label Information Database*) (OSBORNE; SIMBA, 2003). Através da análise do rótulo do pacote é que o roteador que suporta o MPLS vai saber para qual interface de saída o mesmo deve ser encaminhado.
- b) o campo “Exp” é um identificador de 3 bits utilizados para a marcação de pacotes com a finalidade da aplicação da Engenharia de Tráfego conjuntamente com a funcionalidade de diferenciação de tráfego (aplicação de técnicas de QoS).

- c) o campo “S” é um identificador de 1 bit, responsável por indicar o final da “pilha” de rótulos. Um pacote IP pode ter vários rótulos armazenados em uma “pilha” de rótulos. Em cada roteador da rede que o pacote passar, somente o rótulo mais externo é considerado para efeitos de encaminhamento do pacote.
- d) o campo “TTL” é um indicador de 8 bits, sendo normalmente uma cópia direta do parâmetro TTL do cabeçalho do protocolo IP. Este campo é decrementado a cada “salto” (*hop*) para evitar laços de roteamento.

No MPLS não há a inspeção do cabeçalho nos pacotes IP em todos os nós de roteamento (análise salto-a-salto), e sim somente nos roteadores de fronteira do “domínio MPLS”<sup>2</sup> (quando o Pacote IP entra na rede MPLS). Os roteadores de borda *Label Switch Edge Routers* (LERs) inserem rótulos aos pacotes que entram no “domínio MPLS”, de acordo com as denominadas classes de equivalência de encaminhamento *Forwarding Equivalent Classes* (FECs). Os pacotes são agrupados e encaminhados através de uma rede MPLS baseados na associação com a FEC, recebendo o mesmo tratamento ao longo da rede.

De acordo com (MINEI; LUCEK, 2005) existem diversas possibilidades para a definição de uma FEC. Ela pode incluir todos os pacotes que tenham o mesmo destino na rede, pode incluir todos os pacotes oriundos de uma mesma aplicação (porta), ou estar associada a alguma configuração de classe de serviço correspondentes a uma dada funcionalidade de QoS. A FEC é portanto representada numericamente pelo rótulo. Uma vez associados a um FEC, os pacotes seguirão pela mesma rota na rede (Figura 6). Esta rota é um caminho lógico denominado de caminho comutado por rótulo ou *Label Switch Path* (LSP).

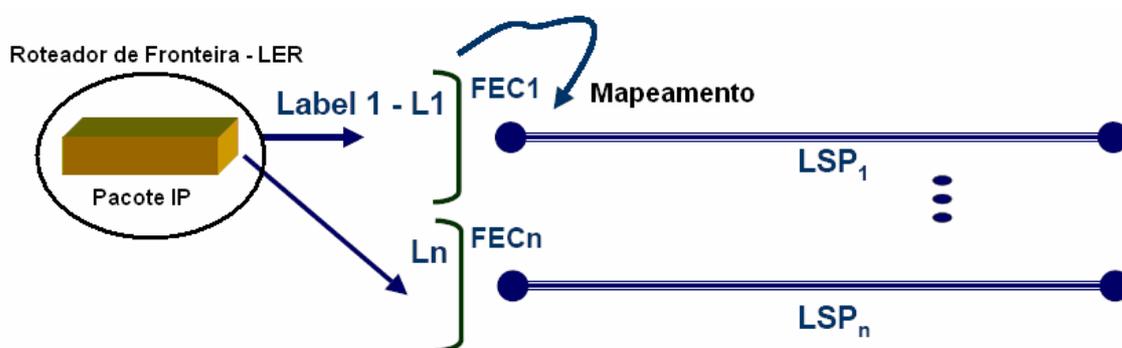


Figura 6 - Atribuição de FEC no Roteador de Fronteira – LER (Elaborado pelo autor, 2008)

<sup>2</sup> Um domínio MPLS é um conjunto de roteadores habilitados para operar com a tecnologia MPLS que fazem parte de uma mesma rede.

À medida que os pacotes vão atravessando os roteadores da rede, os rótulos vão sendo comutados (*switched*). Os pacotes são encaminhados para as interfaces de saída até que cheguem ao roteador de borda da rede, onde os rótulos são retirados e o pacote IP volta à sua semântica normal. Cada roteador MPLS possui uma tabela de encaminhamento com a correspondência entre as FECs, os rótulos, e as interfaces de saída dos roteadores. Um exemplo ilustrativo do processo de encaminhamento em uma rede MPLS é mostrado na Figura 7:

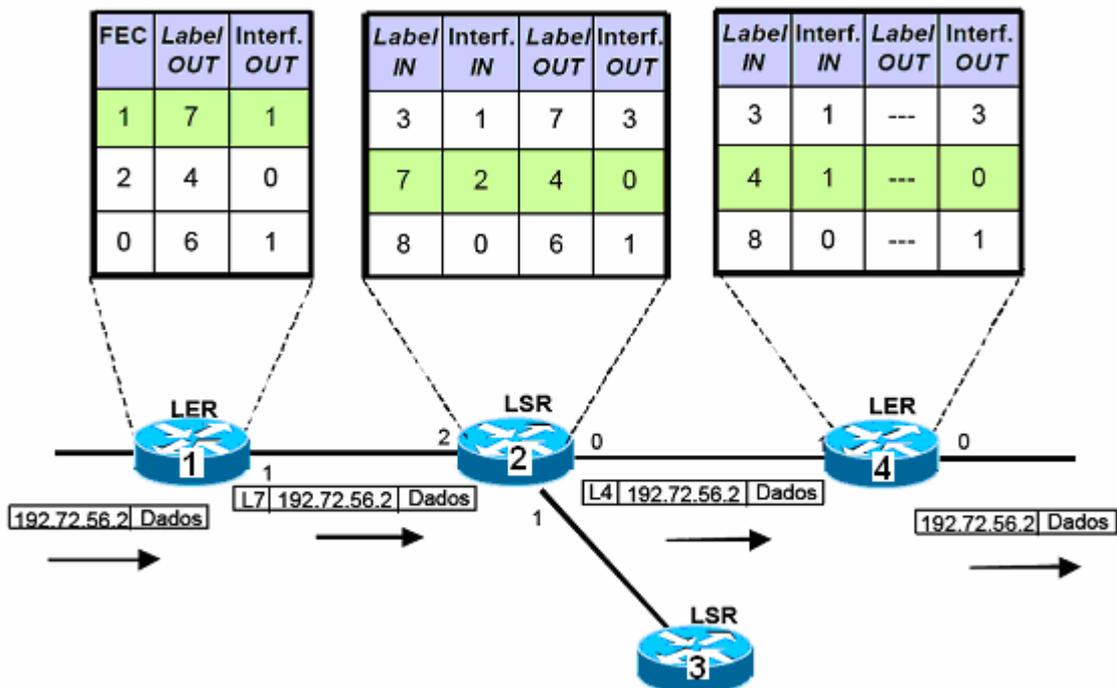


Figura 7 - Processo de Encaminhamento em um Domínio MPLS (Guido, 2004)

No exemplo da Figura 7, o pacote com destino endereço IP de destino 192.72.56.2 entra no domínio MPLS. Esse endereço de destino está associado à FEC1, e no roteador 1 (um) (de borda) de entrada na rede é mapeado com o rótulo 7 (sete) e enviado à interface de saída 1 (um). Os roteadores seguintes encaminham o pacote para uma interface de saída a partir do rótulo de entrada no qual o mesmo está associado (não mais fazem associação com a FEC), e realizam a troca do rótulo de acordo com a correspondência da tabela de encaminhamento presente em cada roteador. Esse processo de troca de rótulos é também conhecido na literatura como *label swap*, no qual um roteador designa um novo rótulo de saída para o pacote IP. Dessa forma, uma maior escalabilidade na utilização dos rótulos é atingida, já que um mesmo rótulo não precisa ser utilizado por todos os roteadores pertencentes ao caminho. O roteador 2 (dois) verifica que o rótulo de entrada do pacote é o rótulo 7 (sete), trocando o rótulo do

pacote para o rótulo 4 (quatro) e encaminha-o para a interface de saída 0 (zero). O roteador 3 (três) verifica que o rótulo de entrada do pacote é o rótulo 4 (quatro) e encaminha-o para a interface de saída 0 (zero). Como o roteador 3 (três) é o último roteador do domínio MPLS, não há mais necessidade de introdução de um rótulo MPLS. O rótulo então é retirado e o encaminhamento segue na forma IP convencional.

### 2.5.2 Plano de controle do MPLS

O Plano de Controle do MPLS é o responsável pela construção e manutenção da tabela de encaminhamento MPLS. Em outras palavras, o plano de controle do MPLS é responsável pela criação dos LSPs ao longo de um Domínio MPLS.

Quando o foco do MPLS era apenas a otimização do desempenho dos roteadores, a construção e manutenção da tabela de encaminhamento era executada no plano de controle basicamente pela atuação em conjunto de um protocolo de roteamento como o *Open Shortest Path First* (OSPF) com um protocolo de distribuição de rótulos, como, por exemplo, o (*Label Distribution Protocol* (LDP) (MARTINS *et al.*, 2003). Através de um conjunto de procedimentos e trocas de mensagens, os LSRs constroem LSPs pela rede MPLS, mapeando as informações de cálculo do menor caminho realizada pelo protocolo de roteamento. A seguir são descritas as etapas para a atuação do plano de controle em um domínio MPLS:

- a) roteadores MPLS executam protocolos de roteamento (OSPF, BGP, ...) para calcularem o melhor caminho entre um nó de origem e um nó de destino (Figura 8);

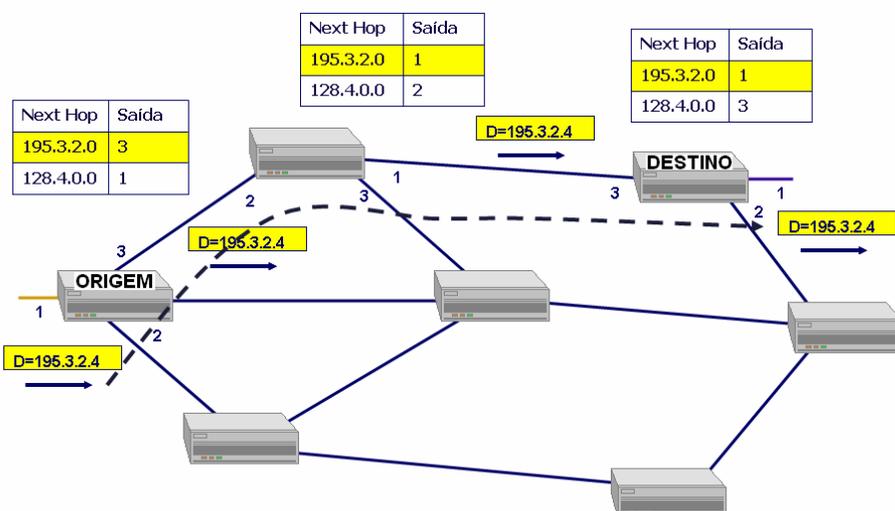


Figura 8 - Seleção do Menor Caminho pelo Protocolo de Roteamento (Martins et al., 2003)

- b) a partir do caminho calculado pelo protocolo de roteamento, o LSP é criado através da rede MPLS pelo protocolo de distribuição de rótulo (Ex: LDP). A partir da distribuição dos rótulos é criada uma base de dados de rótulos *Label Information Base* (LIB) em cada roteador MPLS pertencente ao caminho calculado pelo protocolo de roteamento (Figura 9);

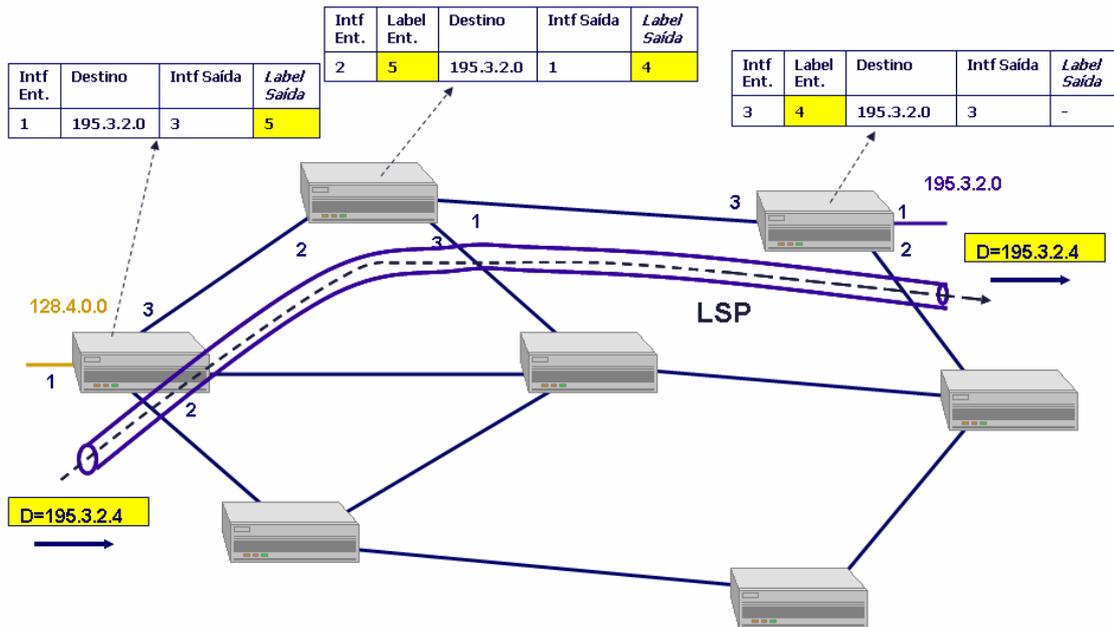


Figura 9 - Distribuição de Rótulos e Construção da Tabela de Encaminhamento (Martins et al., 2003)

Com a introdução da possibilidade de Engenharia de Tráfego com a tecnologia MPLS, os protocolos de distribuição de rótulos requerem extensões quando comparados ao protocolo LDP. Uma das extensões necessárias é a possibilidade de reserva de largura de banda para um LSP a ser estabelecido em um Domínio MPLS-TE. Dois protocolos de distribuição de rótulos/sinalização foram inicialmente utilizados, o *Constraint-based Routing Label Distribution Protocol* – Protocolo de Distribuição de Rótulos com Roteamento baseado em Restrições (CR-LDP) e o *Resource Reservation Protocol – Traffic Engineering* – Protocolo de Reserva de Recursos com Engenharia de Tráfego (RSVP-TE), sendo que o RSVP-TE passou a ser o protocolo de sinalização difundido para a aplicações de Engenharia de Tráfego com o MPLS (ANDERSSON; SWALLOW, 2003).

## 2.6 PLANO DE CONTROLE DO MPLS PARA APLICAÇÃO DA ENGENHARIA DE TRÁFEGO

Outro foco de interesse do MPLS além da otimização do encaminhamento nos roteadores é a possibilidade de, através das funcionalidades do protocolo, ser implementada a Engenharia de Tráfego. Dentre outros aspectos de interesse da Engenharia de Tráfego, uma melhor utilização da real capacidade de uma rede, através de um mapeamento adequado dos requisitos de largura de banda das aplicações nos recursos da rede, é possível de ser atingida. Evitando-se assim o congestionamento e o conseqüente baixo desempenho das aplicações em rede.

Entre outros aspectos, as atividades associadas ao plano de controle quando a Engenharia de Tráfego é aplicada com o MPLS são: o conhecimento do estado dos enlaces, cálculo do caminho baseado em restrições para as requisições de LSPs, e o estabelecimento ou sinalização do caminho com a possibilidade de reserva de recursos (largura de banda), detalhadas nas subseções seguintes.

### 2.6.1 Informação com o estado da rede

O objetivo da Engenharia de Tráfego executada com o MPLS é encontrar um caminho para um LSP de forma a atingir uma ou mais restrições previamente especificadas. Algumas das restrições passíveis de existirem para o cálculo do caminho de um LSP são listadas a seguir:

- a) largura de Banda (Ex: Um LSP com mínimo de 10Mbps de um nó origem “x” a um nó destino “y”);
- b) número de *hops* que determinado tipo de tráfego é capaz de transitar;
- c) prioridade do LSP;
- d) atributos administrativos, como a marcação de determinado enlace com uma cor restritiva de forma que sejam evitados enlaces com alta latência no caminho do LSP.

Calcular um caminho que satisfaça um conjunto de restrições requer saber a informação de quais enlaces na rede podem atender essas restrições. Estas informações devem estar disponíveis para todos os roteadores da rede no caso de um controle distribuído da Engenharia de Tráfego, ou as informações devem estar disponíveis na entidade gerenciadora em caso de uma Engenharia de Tráfego centralizada (AWDUCHE *et al.*, 1999a).

No caso do **controle distribuído** as informações dos enlaces relevantes para a Engenharia de Tráfego devem ser distribuídas pelos roteadores da rede, que são os responsáveis por calcular o caminho que seguirá cada LSP na rede. Este objetivo é atingido através das extensões dos protocolos de estado de enlace (LE FAUCHER, 2005b), tais como o *Intermediate System-to-Intermediate System* (IS-IS) ou o *Open Shortest Path First* (OSPF) (OSBORNE; SIMBA, 2003). Através destas extensões de protocolos é possível propagar para os roteadores da rede o estado (*up* ou *down*), atributos administrativos e a largura de banda dos enlaces. Desta forma cada nó da rede tem sempre o conhecimento das condições atuais dos enlaces da rede. As informações propagadas são armazenadas na denominada base de dados de Engenharia de Tráfego *Traffic Engineering Database* (TED) de cada um dos roteadores da rede, e são utilizadas para o cálculo/computação do caminho para os LSPs na rede.

A propagação das informações através dos protocolos de estado de enlace é feita periodicamente tipicamente com intervalo de 30 minutos (MINEI; LUCEK, 2005). Novos anúncios entre os roteadores da rede devem ser feitos quando houver alterações na largura de banda disponível dos enlaces.

Visando proteger a rede de inundações de anúncios dos protocolos de estado de enlace, novos anúncios não são enviados para toda alteração de banda nos enlaces da rede, e sim apenas para alterações que sejam consideradas relevantes (apenas para alterações de banda acima de determinado valor percentual).

No **controle centralizado** existe uma entidade gerenciadora central que realiza o plano de controle, e na qual estão concentradas as informações dos estados da topologia e enlaces da rede. De acordo com (SCOGLIO et al., 2004), uma entidade gerenciadora deve ser capaz de:

- a) receber uma requisição de reserva de largura de banda ou de estabelecimento de LSP do usuário;
- b) implementar algoritmos propostos para a boa performance do roteamento, estabelecimento do LSP e preempção;
- c) enviar comandos e configurar os roteadores da rede de forma que seja possível criar os LSPs e rotar o tráfego ao longo dos LSPs.

## 2.6.2 Cálculo de caminho baseado em restrições

A escolha de um caminho para o estabelecimento de um LSP dentro de um domínio MPLS deve ser feita baseada em um algoritmo de cálculo de caminho baseado em restrições ou algoritmo de roteamento baseado em restrições (MINEI; LUCEK, 2005; AWDUCHE *et al.*, 1999a).

Os algoritmos convencionais para o roteamento IP almejam encontrar rotas que minimizem ou otimizem uma certa métrica escalar (como, por exemplo, o número de saltos) (KUROSE; ROSS, 2006; TANENBAUM, 2003). No RIP, por exemplo, a métrica é o número de saltos (hops) ou nós. O RIP usa o algoritmo Bellman-Ford para computar, dentre vários caminhos, aquele que possui o menor número de nós ou hops. Para o OSPF, a métrica é estabelecida administrativamente, ou seja, o administrador ou gerente da rede estabelece, para cada enlace, a métrica desejada. O OSPF, então, recorre ao algoritmo Dijkstra a fim de calcular uma rota que minimize a métrica definida, sendo que a métrica total é a soma das métricas individuais de cada enlace, definidas pelo administrador da rede.

Um algoritmo de cálculo de caminho baseado em restrições busca encontrar uma rota que otimize uma certa métrica e, ao mesmo tempo, não viole um conjunto de restrições. A característica de encontrar uma rota que obedeça a um conjunto de restrições é o que distingue o cálculo do caminho das redes IP convencionais para o cálculo do caminho baseado em restrições.

Um algoritmo especificado para o cálculo de caminho baseado em restrições para redes que utilizam o MPLS é o *Constrained Shortest Path First* (CSPF) (MINEI; LUCEK, 2005). O CSPF apenas considera para o cálculo os caminhos que satisfaçam as restrições pré-definidas pelo gerente da rede. Para tal, todos os enlaces que não satisfaçam as condições exigidas não podem ser considerados no cálculo do caminho. A seguir é mostrado um exemplo do cálculo do caminho com o algoritmo CSPF, visando minimizar o número de saltos entre os nós origem-destino satisfazendo os requisitos mínimos de largura de banda dos LSPs a serem estabelecidos em uma determinada topologia de rede (Figura 10).

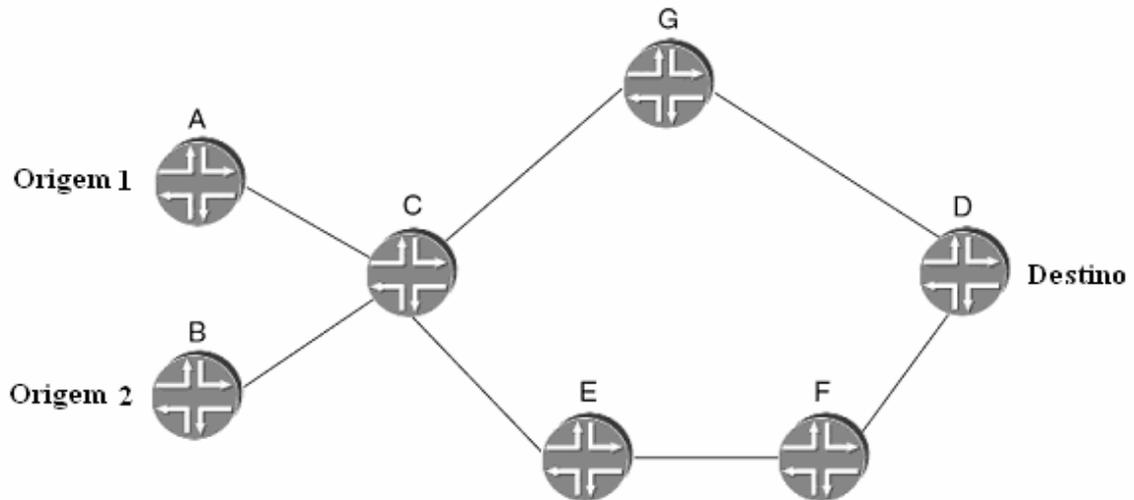


Figura 10 - Cálculo do caminho de um LSP através do CSPF (Elaborado pelo autor, 2008)

No exemplo mostrado (Figura 10) todos os enlaces da rede são de 150Mbps. Considerando que um primeiro LSP1 (120Mbps) solicita o estabelecimento com origem em A e destino em D. Nesse caso, como o objetivo é minimizar a quantidade de saltos, o caminho A-C-G-D é o escolhido pois todos os enlaces apresentam a largura de banda especificada pelo LSP1. Após o LSP1 ser estabelecido, um novo LSP2 (40Mbps) com origem em B e destino em D solicita o estabelecimento. Mesmo sendo o menor caminho A-C-G-D, nesse caso o CSPF selecionará o maior caminho B-C-E-F-D pois o menor caminho não satisfaz os requisitos de largura de banda do LSP2.

### 2.6.3 Estabelecimento de caminho e controle de admissão – RSVP-TE

Depois do cálculo do caminho ao qual seguirá um LSP pelo algoritmo de cálculo de caminho baseado em restrições, é necessário o estabelecimento ou sinalização do caminho propriamente dito. Em outras palavras, após o caminho do LSP ter sido calculado é necessário que o mesmo seja criada de fato. O processo de estabelecimento de um LSP em uma rede MPLS está associado ao fato da troca de rótulos entre os roteadores que pertencem ao caminho calculado e a consequente reserva de recurso (banda) necessária. Duas razões para a necessidade de sinalização do caminho são (MINEI; LUCEK, 2005):

- a) informar aos roteadores de salto a salto que um determinado LSP passará por cada um deles;
- b) reservar os recursos reservados (Largura de Banda) ao longo do caminho.

No MPLS a criação do LSP é realizada pelo protocolo RSVP-TE (AWDUCHE *et al*, 2001). O protocolo RSVP-TE é uma extensão do protocolo RSVP<sup>3</sup> para a utilização da reserva de recursos dentro do contexto da Engenharia da Tráfego com MPLS. O RSVP-TE possui três funções básicas:

- a) configuração e manutenção do caminho;
- b) encerramento de caminho;
- c) sinalização de erro.

A operação básica do RSVP-TE para a sinalização ou estabelecimento de um LSP em uma rede MPLS é feita através da troca das mensagens dos tipos PATH e RESV (AWDUCHE *et al*, 2001). A criação de um LSP sinalizado pelo RSVP-TE é iniciada a partir do envio de mensagens PATH de um roteador de ingresso na rede MPLS para um roteador de egresso na rede MPLS, como mostrado na Figura 11.

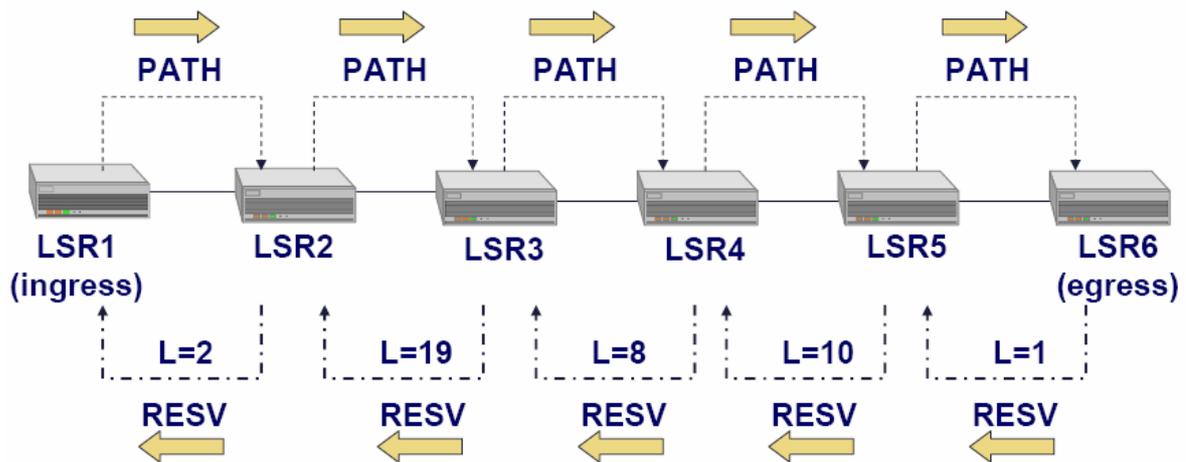


Figura 11 – Troca de mensagens RESV e PATH no protocolo RSVP-TE (MARTINS *et al*, 2003)

O conteúdo da mensagem PATH é composta pelos seguintes objetos:

- a) **Label Request Object**: Requisita um rótulo MPLS para o caminho associado ao LSP. Como consequência, os roteadores de egresso bem como os roteadores intermediários do caminho alocam um rótulo para o determinado LSP;

<sup>3</sup> O RSVP foi concebido antes do MPLS, e era utilizado para a realização de reserva de recursos para conexões de fluxos individuais para usuários em rede (por exemplo, uma sessão de vídeo telefonia entre um par de hosts particular). Essa utilização antiga do RSVP era feita para um tipo de rede denominada rede IntServ, modelo concorrente às redes DiffServ, que acabou por não vingarem principalmente devido à sua baixa escalabilidade.

- b) **Explicit Route Object (ERO):** O ERO contém os endereços dos nós através dos quais o LSP deve passar. Se requerido, o ERO pode conter todo o caminho que o LSP deve seguir desde o roteador de ingresso até o roteador de egresso da rede. A entrada do objeto ERO corresponde na verdade à saída do caminho que foi calculado pelo algoritmos de cálculo do caminho baseado em restrições;
- c) **Record Route Object (RRO):** O RRO requisita que o caminho a ser seguido pela mensagem PATH seja armazenado. Este será o caminho do LSP caso o mesmo seja criado com sucesso. Cada roteador através da mensagem PATH adiciona o seu endereço na lista dentro do objeto RRO. O objeto RRO também é utilizado para a detecção de laços, no caso de um roteador encontrar o seu próprio endereço já incluído no objeto RRO no momento em que for adicioná-lo;
- d) **Sender Tspec:** Item da mensagem PATH que contém a informação responsável por habilitar o roteador de ingresso a requisitar a reserva de largura de banda associada ao LSP. Essa informação é de fundamental importância em termos da manutenção dos parâmetros de QoS do tráfego associado ao LSP, já que ela que determina a largura mínima que deve ser reservada para um determinado conjunto de fluxos de tráfego.

Em resposta à mensagem PATH, o roteador de egresso ou destino envia uma mensagem RESV. A troca de rótulos visando o estabelecimento do LSP é feita roteador por roteador até a chegada no roteador de ingresso do LSP. O roteador de egresso envia uma mensagem RESV para o seu vizinho no caminho, sendo que esse procedimento se repete até que a mensagem chegue ao roteador de ingresso. Esta configuração garante que a mensagem RESV seguirá o mesmo caminho da mensagem PATH.

O conteúdo da mensagem RESV é composta pelos seguintes objetos:

- a) **Label Object:** Contém o rótulo que será utilizado para a comunicação entre dois roteadores habilitados com o MPLS. No exemplo da Figura 11, o rótulo 1 (um) é utilizado para a comunicação entre os roteadores 5 e 6. O rótulo 10 (dez) é utilizado para a comunicação entre os roteadores 4 e 5. O rótulo 8 (oito) é utilizado para a comunicação entre os roteadores 3 e 4, e assim sucessivamente. Ou seja, a distribuição do rótulo é feita a partir do último roteador (roteador de destino que é o final do LSP) para o roteador anterior adjacente, sendo que esse processo é repetido até que o rótulo seja atribuído ao último roteador (o roteador de origem do LSP). Essa forma de

distribuição de rótulos do RSVP-TE é conhecida na literatura como “*downstream-on-demand*” (AWDUCHE *et al*, 2001);

- b) **Record Route Object (RRO)**: O objeto RRO na mensagem RESV também tem a função de armazenar o caminho percorrido pela mensagem. Outra vez, assim como na mensagem PATH, a possibilidade de loops de roteamento pode ser detectada pelo objeto RRO. A evolução do conteúdo do objeto RRO é mostrada no exemplo de rede da Figura 12 a seguir.

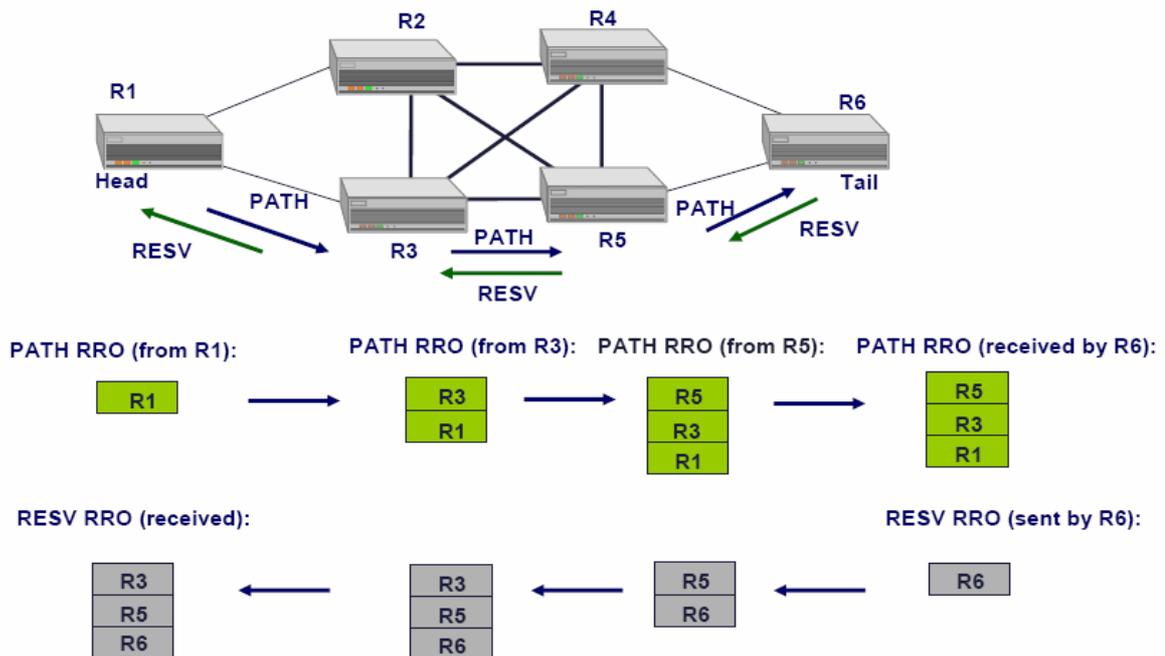


Figura 12 - Exemplo com a Evolução do Conteúdo do Objeto RRO (MARTINS *et al*, 2003)

O processo de confirmação através da troca de mensagens PATH e RESV pelo protocolo RSVP-TE, visando o estabelecimento ou não de um LSP que possua os requisitos de largura de banda desejado é conhecido como controle de admissão (OSBORNE; SIMBA, 2003). O controle de admissão é a garantia de que os recursos especificados vão ser preservados na criação do LSP. O controle de admissão é justificado pelos seguintes fatores:

- o cálculo do caminho a ser seguido pelo LSP pode não ter sido bem computado pelo algoritmo de cálculo de caminho baseado em restrições;
- mesmo que o cálculo do caminho pelo algoritmo baseado em restrições tenha sido bem sucedido, pode acontecer que os recursos disponíveis na rede tenham mudado entre o tempo em que a computação final do caminho foi executada e a sinalização para o estabelecimento do LSP;

- c) pode haver inconsistência na informação das *Traffic Engineering Databases* (TEDs), no caso da utilização de IGP's para a propagação das informações dos estados dos enlaces das redes para os roteadores. Esse último caso é aplicável apenas quando há a utilização do controle distribuído.

Dentro do conceito de controle de admissão, as mensagens PATH são responsáveis por solicitar o recurso no caminho especificado. As mensagens RESV são as que na verdade reservam os recursos. Em caso de haver recursos ao longo de todos os enlaces do caminho, o LSP estará estabelecido no momento em que a última mensagem RESV chegar ao roteador de origem do LSP. Caso não haja recursos em algum dos enlaces do caminho, é enviada uma mensagem de erro ("*PathErr*") para o roteador de origem do LSP informando que o mesmo não poderá ser estabelecido.

Em caso de utilização dos recursos de prioridade e preempção, a mensagem RESV é a responsável por desestabelecer um LSP (*Tear Down*) que tiverem uma baixa prioridade em relação a uma nova requisição de LSP.

## 2.7 SUPORTE DO MPLS-TE PARA MAPEAR O DIFFSERV

Em (LE FAUCHER *et al.*, 2002) foram especificadas alternativas de suporte do MPLS-TE para mapear a arquitetura DiffServ. O grande desafio para o suporte ao DiffServ em redes que utilizem o MPLS-TE deveu-se a necessidade dos LSRs realizarem as decisões de encaminhamento com base apenas no cabeçalho MPLS-TE. Portanto, os comportamentos (PHBs) previstos no DiffServ deveriam ser inferidos pelo cabeçalho MPLS-TE no roteador no qual o LSP é iniciado.

A primeira alternativa é conhecida como E-LSP (LE FAUCHER *et al.*, 2002), que significa o mapeamento das classes DiffServ em LSPs MPLS-TE com base no campo "EXP" do cabeçalho MPLS-TE. O E-LSP é apenas aplicável às redes que possuam menos que oito classes de serviço DiffServ definidas em rede (i.e., oito PHBs), devido ao fato do campo DSCP do cabeçalho *DiffServ* (6 bits) ser mapeado diretamente no campo EXP do cabeçalho MPLS-TE (3 bits). Como o campo EXP possui apenas 3 bits, o número de classes de serviço por LSP fica limitado a 8 (oito), devido às combinações binárias possíveis do campo EXP para mapear cada uma das classe de serviço DiffServ definida em rede. Com essa abordagem,

a definição da política de encaminhamento (PHBs – EFs, AFs e serviço melhor esforço) é atribuída a partir do próprio campo EXP inserido no pacote IP na entrada do domínio MPLS-TE. Ou seja, o mapeamento do MPLS-TE no *DiffServ* ocorre no roteador de fronteira LER. Nessa solução, o tráfego das classes de serviço *DiffServ* configuradas é enviado através de um único LSP. O quadro 4 apresenta um exemplo de mapeamento possível dos valores do campo “EXP” do cabeçalho MPLS-TE dentro de possíveis valores de código DSCP do *DiffServ* existentes. No exemplo do quadro 4 todas as oito possibilidades de mapeamento foram utilizadas.

EXP - Decimal	EXP - Binário	DCSP - Decimal	DSCP - Binário
0	000	0	000000
1	001	8	001000
2	010	16	010000
3	011	24	011000
4	100	32	100000
5	101	40	101000
6	110	48	110000
7	111	56	111000

Quadro 4- Mapeamento do *DiffServ* no MPLS-TE (E-LSP) (Elaborado pelo autor, 2008)

A segunda alternativa é conhecida como L-LSP (LE FAUCHER *et al.*, 2002), que significa o mapeamento das classes *DiffServ* em LSPs MPLS-TE com base no próprio rótulo do cabeçalho MPLS-TE. Com essa abordagem a associação da política de encaminhamento *DiffServ* (PHB – EFs, AFs e serviço melhor esforço) é realizada no momento do estabelecimento do próprio LSP. Portanto, durante o encaminhamento MPLS-TE (em cada salto) o rótulo determina o comportamento de escalonamento a ser utilizado, além da função original de determinar para qual interface de saída do roteador o pacote deve seguir. Na alternativa L-LSP, os bits EXP podem ser utilizados para a implementação da precedência de descarte associada a determinado pacote. Portanto, em um L-LSP podem trafegar pacotes de um único PHB ou de alguns PHBs que contenham comportamento similar em termos de escalonamento (Por exemplo, PHBs AF1y, onde “y” se refere a diferentes precedências de descarte). O quadro 5 apresenta um exemplo de mapeamento possível para associação de PHBs com um LSP estabelecido (de acordo com o rótulo MPLS-TE) e para valores de rótulos indicando a precedência de descarte (de acordo com o campo EXP).

PHB	Rótulo	EXP - Decimal	EXP - Binário	LSP estabelecido associado/Nível de Descarte de pacotes
EF	12	-	-	LSP 1
AF1.1	34	1	001	LSP 2/Baixo
AF1.2	6	2	010	LSP 2/Médio
AF1.3	8	3	011	LSP 2/Alto
AF2.1	14	4	100	LSP 3/Baixo
AF2.2	3	5	101	LSP 3/Médio
AF2.3	9	6	110	LSP 3/Alto
BE	16	-	-	LSP 4

Quadro 5 - Mapeamento do DiffServ no MPLS-TE (L-LSP) (Elaborado pelo autor, 2008)

Uma ilustração da estratégia L-LSP é mostrada na Figura 13. Nela o tráfego EF é mapeado no LSP2 e o tráfego AF1Y é mapeado no LSP1.

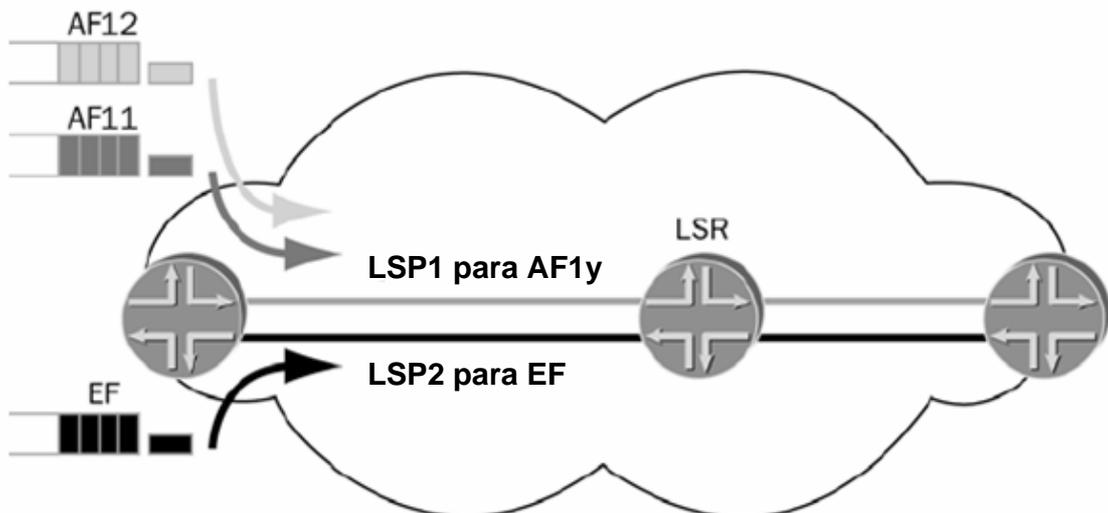


Figura 13 - Mapeamento do DiffServ no MPLS-TE (L-LSP) (Elaborado pelo autor, 2008)

Portanto, em algumas redes DiffServ onde a otimização dos recursos de transmissão também é desejada, os mecanismos DiffServ podem ser complementados pelos mecanismos da Engenharia de Tráfego com MPLS (MPLS-TE). Neste caso, o MPLS-TE opera a Engenharia de Tráfego considerando todas as classes de serviço como um todo (a restrição para o roteamento é a largura de banda total do enlace). Com esta opção, o *DiffServ* e o MPLS-TE, promovem os seus benefícios com seus respectivos mecanismos separadamente.

Algumas considerações importantes sobre o suporte do MPLS-TE ao DiffServ são feitas a seguir:

- a) a tecnologia DiffServ possui a capacidade de alocação de banda por classe ao nível do plano de encaminhamento de pacotes (salto-a-salto). Já o MPLS-TE é uma tecnologia que é capaz de realizar a alocação de banda ao nível do plano de controle (fim-a-fim);

- b) o fato do MPLS-TE fornecer suporte ao mapeamento do DiffServ não significa que a alocação de banda por classe de serviço feita pelo DiffServ será realizada de maneira fim-a-fim. A ocorrência de disputa por largura de banda entre LSPs em determinado enlace e a ordem na qual os LSPs são estabelecidos em rede, são dois exemplos de fatores que fazem com que não seja possível garantir fim-a-fim na rede a alocação de recursos para determinada classe de serviço. O plano de controle do MPLS-TE realiza o roteamento baseado em restrições considerando apenas a largura dos enlaces como um todo como restrição de banda no momento do cálculo do caminho para os LSPs na rede;
- c) o MPLS-TE não possui a capacidade de realizar o roteamento baseado em múltiplas restrições de largura de banda, i.e., não possui o suporte de estabelecer restrições de banda para diferentes classes;
- d) se a ordem de chegada de cada um dos LSPs associados a diferentes classes de serviço DiffServ for alterada (fato que ocorre na prática quando a chegada de tráfego em um roteador é aleatória), os caminhos “físicos” por que cada um dos LSPs seguirá na rede também poderão ser alterados. Esse fator pode fazer com que um LSP associado a uma classe de serviço associada ao tráfego de voz e/ou vídeo em tempo real seja preterido com relação a um LSP associado ao mero tráfego de dados. Isso ocorre pois não existe uma previsão de recurso a ser disponibilizado/utilizado na rede por classe de serviço, de forma a garantir fim-a-fim que os recursos sempre estarão disponíveis para as classes de serviço que contenham requisitos mais restritos de QoS.

O plano de controle com suporte ao roteamento baseado em restrições considerando restrições de largura de banda por classe de serviço, com a alocação de banda por classe de serviço fim-a-fim torna-se possível com a alternativa tecnológica *DiffServ Aware MPLS Traffic Engineering* (DS-TE) (LE FAUCHER; LAI, 2003) apresentada no capítulo 3 a seguir.

### 3 ARQUITETURA DS-TE

Neste capítulo são apresentados os requisitos para uma rede suportar os serviços diferenciados numa estratégia de Engenharia de Tráfego com MPLS a nível de classes, conhecido na literatura como *DiffServ Aware MPLS Traffic Engineering* (DS-TE) (LE FAUCHER; LAI, 2003).

#### 3.1 INTRODUÇÃO

O DiffServ é utilizado pelos provedores de serviços para o projeto de redes escaláveis que suportem múltiplas classes de serviço. Com ele é possível alocar diferentes larguras de banda para diferentes classes de serviço no nível do plano de encaminhamento. A atuação no plano de encaminhamento significa que o DiffServ atua de uma maneira isolada salto-a-salto em cada roteador no momento do encaminhamento dos pacotes. A atuação DiffServ é completamente independente do caminho ao qual o pacote vai seguir na rede. Para o DiffServ não importa se o pacote seguirá por um caminho congestionado, pois ele não tem associação e não se comunica com o plano de controle responsável pela seleção e estabelecimento do caminho. Por mais que haja uma diferenciação de tráfego no roteador por parte do DiffServ, se o tráfego seguir um caminho congestionado, pacotes podem ser perdidos ou experimentam atraso e *Jitter* possivelmente em desacordo com o que esteja especificado para uma operação com qualidade.

Para atingir a alocação de banda por classe de serviço fim-a-fim na rede, uma granularidade mais fina para a otimização dos recursos de transmissão, a promoção do aumento da performance e eficiência da rede, é desejável. Neste cenário é importante que a Engenharia de Tráfego seja executada ao nível de classes, ao invés de considerar todo o agregado de tráfego dos enlaces (LE FAUCHER; LAI, 2003; ZHANG *et al.*, 2007; MINEI; LUCEK, 2005). Esse conceito é a base para o desenvolvimento da arquitetura DS-TE (LE FAUCHER; LAI, 2003).

O DS-TE permite fornecer a característica de orientação à conexão do MPLS-TE na granularidade de classes de serviço. Com o DS-TE é possível que a alocação de banda por classe de serviço que é feita no DiffServ salto-a-salto, no momento do encaminhamento dos pacotes, seja estendida para atuar ao nível do plano de controle considerando os aspectos de

cálculo e seleção de caminhos fim-a-fim na rede. No momento do cálculo e estabelecimento do caminho de um LSP pelo plano de controle, é possível realizar a alocação de largura de banda fim-a-fim ao nível de classes, através do conhecimento da capacidade disponível de largura de banda do tipo de classe (CT – *Class Type*) em todos os enlaces da rede.

O DS-TE impõe diferentes restrições de banda *Bandwidth Constraints* (BCs) para diferentes tipos de classes (CTs) configurados em rede. O uso do controle de admissão ao nível de classes é possível, com o número máximo simultâneo de LSPs associados ao tráfego de uma mesma classe sendo limitado pelas regras (restrições de banda) impostas pelo DS-TE (LAI, 2005). Através do controle do número de LSPs admitidos a partir de diferentes tipos de classes (CTs), é possível fazer com que a quantidade de tráfego de diferentes agregados de fluxos esteja compatível com os objetivos de QoS a serem atingidos ao nível de pacotes.

### 3.2 TIPOS DE CLASSE (CTS)

O requisito fundamental do DS-TE é estar apto a garantir diferentes requisitos de largura de banda para diferentes conjuntos de "troncos de tráfego". Um "tronco de tráfego" em redes DS-TE é definido como um agregado de fluxos de tráfego pertencente a uma mesma classe e alocadas para um mesmo LSP (LE FAUCHER; LAI, 2003). Nesse sentido surge o conceito de Tipos de Classe (CT) para a aplicação das redes DS-TE.

Em (LE FAUCHER; LAI, 2003) o conceito de Tipo de Classe (CT) é definido da seguinte maneira:

O conjunto de troncos de tráfego que passam por um determinado enlace, que está governado por um conjunto específico de restrições de utilização da largura de banda nos enlaces da rede. Um Tipo de Classe (CT) é utilizado para os propósitos de alocação da banda, para o roteamento baseado em restrições e para o controle de admissão nas redes DS-TE. Um determinado Tronco de Tráfego pertence ao mesmo CT em todos os enlaces da rede DS-TE.

De acordo com o que está especificado em (LE FAUCHER; LAI, 2003), o mapeamento de classes de serviço DiffServ *Per Hop Behavior* (PHB) nos CTs do DS-TE é bastante flexível. Uma classe de serviço DiffServ pode ser mapeada diretamente em um CT, diferentes classes de serviço DiffServ podem ser mapeadas em diferentes CTs, múltiplas classes de serviço DiffServ podem ser mapeadas em um mesmo CT, e uma classe de serviço DiffServ pode ser

mapeada em múltiplos CTs. Dentro dessa possibilidade, o gerente da rede poderia dividir o tráfego associado à classe de serviço EF em dois diferentes "troncos de tráfego", assim cada um dos "troncos de tráfego" estaria subordinado a diferentes restrições de banda, com dois CTs associados ao tráfego da classe de serviço EF.

Para uma rede configurada com 4 classes de serviço DiffServ (EF, AF1, AF2 e Melhor Esforço), o administrador da rede pode, por exemplo, decidir configurar os CTs do DS-TE da maneira apresentada na quadro 6.

<b>Tipo de Classe DS-TE</b>	<b>Classe de Serviço DiffServ</b>	<b>Descrição da Aplicação</b>
CT2	EF	VoIP
CT1	AF1	Dados de alta prioridade
CT1	AF2	Dados de baixa prioridade
CT0	BE	Melhor Esforço ou Desconhecido

Quadro 6 - Mapeamento de Classes DiffServ em CTs DS-TE (Elaborado pelo autor, 2008)

- a) um CT para cada uma das 4 classes de serviço DiffServ configuradas;
- b) devido às classes AF1 e AF2 compartilharem uma necessidade similar de serviço (baixa perda), e compartilharem necessidades similares em termos de escalonamento, os mesmos requisitos de restrições de banda podem ser aplicados simultaneamente para as duas classes de serviço AF1 e AF2. Portanto, o administrador da rede poderia configurar três CTs: Um para os trocos de tráfego associados à classe de serviço EF, um para os LSPs associados às classes de serviço AF1 e AF2, e outro para os LSPs associados à classe de serviço melhor esforço.

Segundo recomendado em (LE FAUCHER; LAI, 2003), a solução DS-TE deve suportar até 8 (oito) CTs configurados em rede. Como a numeração dos CTs começa em 0 (zero) e a máxima quantidade de CTs configurados em rede é 8 (oito), a numeração que é dada aos CTs pode variar de 0 (zero) ao número máximo de CTs menos um. Tipicamente os CTs mais prioritárias em uma rede DS-TE estão associados a tráfegos iterativos de voz e vídeo (em tempo-real).

Normalmente não se faz necessária a utilização do número máximo de CTs recomendado em rede para a aplicação da tecnologia DS-TE. Sendo que uma implementação DS-TE deve possuir no mínimo dois CTs diferentes para que seja justificável a sua utilização.

### 3.3 PRIORIDADE E PREEMPÇÃO EM REDES DS-TE

O conceito de preempção em redes DS-TE está associado ao fato de um LSP já estabelecido ser encerrado, visando disponibilizar recursos para um outro LSP a ser estabelecido de acordo com determinado critério.

A preempção pode ser usada como uma ferramenta que ajuda a garantir que LSPs de alta prioridade possam ser roteados através de caminhos mais favoráveis e que atendam aos requisitos de QoS em uma rede. Apesar de não ser um atributo mandatório no mundo IP tradicional, a preempção se revela uma ferramenta importante nas redes onde a diferenciação de tráfego é necessária e o cálculo do caminho dos LSPs é feito dinamicamente. O estabelecimento dinâmico de LSPs é geralmente necessário em redes onde não é possível prever ou estimar as demandas de tráfego entre nós origem-destino na rede. Nessas condições, nem sempre os recursos estão disponíveis para o estabelecimento das requisições de LSPs prioritários.

As redes com a arquitetura DS-TE podem utilizar o suporte a preempção recomendado em (LE FAUCHER; LAI, 2003). Quando LSPs associados a diferentes CTs competem por recursos, a preempção é a alternativa para definir quem irá ter prioridade de manutenção ou estabelecimento, e conseqüentemente garantir os recursos previstos para cada um dos CTs na rede.

Na recomendação do DS-TE, o atributo recomendado a ser utilizado como base para a preempção são as prioridades de estabelecimento (*setup*) e de manutenção (*hold*). Se um novo LSP2 a ser estabelecido tiver uma prioridade de setup superior à prioridade de manutenção de um outro LSP1 já estabelecido, é possível a preempção do LSP1 por parte do LSP2. Os LSPs com maior prioridade de preempção são os que possuem menor valor numérico de prioridade em relação aos outros, sendo que os valores de prioridade podem variar de 0 (zero) a 7 (sete). Em (LAI, 2005) é recomendado que, na prática, um mesmo valor numérico seja utilizado para identificar ambas as prioridades de estabelecimento (*setup*) e de manutenção (*hold*).

A partir dos conceitos de prioridade e preempção, é definido o conceito de TE-classe para as redes DS-TE (LE FAUCHE; LAI, 2003):

- a) uma TE-classe consiste de um par (CT, prioridade de preempção), i.e., um CT associado a uma prioridade de preempção.

O valor da prioridade de preempção serve tanto para definir o estabelecimento e/ou a manutenção de um LSP na rede DS-TE. Isto significa que um LSP transportando um "tronco de tráfego" associado a um determinado CT pode utilizar um valor de prioridade de preempção de *setup* no momento do estabelecimento e um valor de prioridade de preempção de manutenção quando já estiver estabelecido, ou utilizar um mesmo valor numérico de prioridade de preempção para representar ambas as prioridades de *setup* e manutenção.

Pela definição de (LE FAUCHER; LAI, 2003):

- a) para um determinado CT, podem existir uma ou mais TE-classes, cada qual utilizando uma diferente prioridade de preempção;
- b) para uma determinada prioridade de preempção, podem existir uma ou mais TE-classes utilizando a mesma prioridade de preempção, cada uma utilizando um CT diferente.

Portanto, a aplicação da preempção em redes DS-TE é muito flexível, podendo variar a depender do objetivo de projeto do gerente da rede. Os casos possíveis são resumidos na quadro 7 (LE FAUCHER; LAI, 2003).

<b>Caso</b>	<b>Variações da aplicação da preempção em redes DS-TE</b>
<b>1</b>	A solução DS-TE pode permitir que todos os LSPs associados a um tipo de classe utilize a mesma prioridade de preempção. Em outras palavras, a solução DS-TE deve permitir que um tipo de classe seja utilizada com uma única TE-classe. Isso faz com que não seja possível a preempção dentro de um mesmo tipo de classe, e sim apenas é possível a preempção entre diferentes tipos de classes, quando assim for desejado pelo administrador da rede.
<b>2</b>	A solução DS-TE também pode permitir que seja possível dois LSPs associados ao mesmo tipo de classe utilizar diferentes prioridades de preempção, permitindo assim que um LSP de maior prioridade de <i>setup</i> (numericamente inferior) "preempte" o LSP com uma prioridade inferior de manutenção (numericamente superior) quando houver a disputa por recursos da rede. Nesse caso, o administrador da rede está permitindo a preempção dentro de um mesmo tipo de classe. Por exemplo, o administrador da rede pode definir um tipo de classe com três diferentes TE-classes, com prioridade de preempções 0(zero), 1(um) e 4(quatro).
<b>3</b>	A solução DS-TE pode permitir que LSPs associados a diferentes tipos de classe utilizem diferentes prioridades de preempção, permitindo assim a preempção entre tipos de classes distintos quando houver disputa por recursos.

<b>4</b>	A solução DS-TE pode permitir que dois ou mais LSPs associados a tipos de classes diferentes utilizem a mesma prioridade de preempção. Nesse caso, TE-classes utilizando diferentes tipos de classes teriam a mesma prioridade de preempção, tornando possível ao administrador da rede garantir que não haverá preempção de LSPs entre tipos de classes. Como exemplo, três tipos de classes (CT0, CT1 e CT2), todos utilizado a prioridade de preempção 0 (zero), onde não será possível a preempção de LSPs entre os tipos de classes.
----------	---

Quadro 7- Aplicação da preempção em redes DS-TE (Elaborado pelo autor, 2008)

Como existem, a partir da recomendação IETF, 8 (prioridades de preempção) e 8 (oito) tipos de classes, isso totalizaria teoricamente 64 (sessenta e quatro) tipos distintos de possibilidade de TE-classes. Entretanto, não é recomendada a utilização na prática de todas essas combinações teóricas possíveis. Em (LE FAUCHER; LAI, 2003; MINEI; LUCEK, 2005) é sugerido um máximo de 8 (oito) TE-classes configuradas em rede dentro das 64 combinações possíveis devido a questões de escalabilidade. Um exemplo ilustrativo de escolha de 8 (oito) das 64 (sessenta e quatro) possibilidades disponíveis para TE-Classes é mostrado na Figura 14 a seguir:

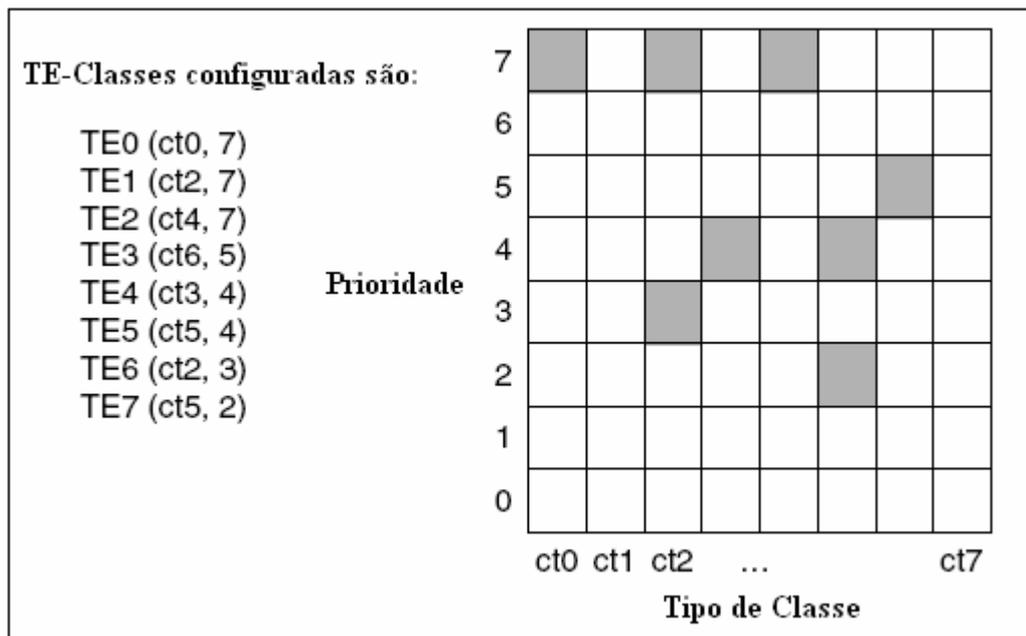


Figura 14 - Escolha das TE-classes em uma rede DS-TE (Minei e Lucek, 2005)

Um possível re-estabelecimento de um LSP que sofreu preempção vai depender se esta funcionalidade está ou não implementada pelo projetista da rede (OSBORNE; SIMBA, 2003). O re-estabelecimento envolve a computação de um novo caminho através de um algoritmo de roteamento baseado em restrições e um protocolo de sinalização de acordo com a capacidade de banda por CT disponível. Portanto, a preempção de um LSP e seu consequente re-estabelecimento sempre envolve uma quantidade adicional de *overhead* na rede.

### 3.4 POLÍTICA DE PREEMPÇÃO PARA REDES DS-TE

Adicionalmente à alternativa de preempção de LSPs recomendada em (LE FAUCHER; LAI, 2003), foi recomendada na RFC 4829 (DE OLIVEIRA *et al.*, 2007) uma política de preempção abrangente para redes DS-TE. De acordo com eles uma política de preempção consiste no agrupamento e definição de um conjunto de critérios de interesse que podem ser escolhidos visando tomar a decisão de qual ou quais LSPs sofrerão preempção para que um novo LSP mais prioritário seja estabelecido.

A política de preempção recomendada na RFC 4829 pode ter seus parâmetros ajustados com a finalidade de atribuir diferentes pesos a diferentes critérios de preempção escolhidos, visando melhor atender as necessidades do projetista em redes DS-TE. Esta recomendação foi o resultado de um conjunto de trabalhos publicados anteriormente pelos autores em (DE OLIVEIRA *et al.*, 2002), (DE OLIVEIRA *et al.*, 2004; SCOGLIO *et al.*, 2002). A política de preempção recomendada em (DE OLIVEIRA *et al.*, 2007) é mais flexível e abrangente do que o critério de preempção recomendado em (LE FAUCHER; LAI, 2003), pois considera as características de tráfego de cada LSP, e não se baseia apenas nas prioridades de *setup* e manutenção.

A política escolhe LSPs menos prioritários para serem “preemptados” de forma a acomodar uma nova requisição mais prioritária com base em valor de prioridade de preempção gerado a partir de um fator, denominado de “fator H”. Para cada LSP a ser estabelecido em rede, o fator H deve ser calculado. O fator H pode contemplar um conjunto de fatores de interesse, como mostrado na Equação (1) a seguir:

$$(1) \quad H(l) = \alpha \cdot Y(l) + \beta \cdot \frac{1}{b(l)} + \gamma \cdot (b(l) - r^2) + \theta \cdot b(l) \quad .$$

As variáveis da Equação (1) que definem o valor do Fator H são detalhadas a seguir:

- a) “Y(l)” está associado à necessidade da preempção de LSPs com baixa prioridade.  $Y(l) = 8 - p$ , onde “p” corresponde à prioridade do LSP.
- b) “1/b(l)” está associado ao impacto de se re-rotear um LSP depois que o mesmo sofrer preempção. Baseia-se no fato de que quanto menor for a banda de uma LSP “preemptado” mais facilmente ele será re-roteada posteriormente, i.e., mais provavelmente ela encontrará recursos disponíveis no momento em que for realizado o re-roteamento;

- c) “ $b(l)-r^2$ ” está associado ao fato de se buscar LSPs para serem “preemptados” que tenham o valor de largura de banda “ $b(l)$ ” o mais próximo possível da largura de banda “ $r$ ” da nova requisição de LSP. Isso minimiza o número de LSPs a serem preemptados, diminuindo a necessidade de re-roteamento de um número grande de LSPs (minimiza o overhead de sinalização na rede);
- d) “ $b(l)$ ” está associado ao fato de minimizar a banda “preemptada”, onde “ $b(l)$ ” corresponde à largura de banda do LSP;
- e)  $\alpha$ ,  $\beta$ ,  $\gamma$  e  $\theta$  são os pesos associados a cada uma das variáveis da equação, a serem atribuídos a depender dos objetivos do projetista da rede.

O fato é que não necessariamente todas as variáveis listadas devem ser consideradas simultaneamente, até mesmo porque o efeito de uma variável pode vir a anular o efeito de uma outra (DE OLIVEIRA *et al.*, 2007). O lado positivo, é que a combinação de fatores para a escolha de LSPs a serem preemptados, fornece flexibilidade para o projetista da rede fazer suas escolhas a depender dos seus objetivos (DE OLIVEIRA *et al.*, 2004; SCOGLIO *et al.*, 2002).

Os LSP estabelecidos passíveis de preempção na rede serão aqueles que apresentarem um menor valor de fator  $H(l)$  do que a nova requisição de LSP (i.e., preempção em ordem crescente do Fator  $H$ ). Segundo a recomendação (DE OLIVEIRA *et al.*, 2007), a preempção de LSPs ocorrerá enquanto o valor da banda “ $r$ ” da nova requisição de LSP não estiver disponível em todos os enlaces do caminho escolhido para seu estabelecimento.

### 3.5 DS-TE VS MPLS-TE

O DS-TE conserva as mesmas características globais da estrutura do MPLS-TE, quais sejam, necessidade do conhecimento das informações dos enlaces da rede, computação de caminho para o LSP e sinalização de LSP.

Entretanto, o DS-TE introduz extensões (LE FAUCHER, 2005b) para suportar o conceito de alocação de banda e para tornar possível o roteamento baseado em restrições e estabelecimento de caminho considerando múltiplas classes de tráfego.

As extensões necessárias ao DS-TE frente ao MPLS-TE são descritas a seguir.

### 3.5.1 Cálculo do caminho em uma rede DS-TE

Em (LE FAUCHER, 2005b) é recomendada a utilização do roteamento baseado em restrições, como o protocolo *Constrained Short Path First* (CSPF) (MINEI; LUCEK, 2005), para o cálculo do caminho de um novo LSP entre um nó origem e um nó destino em uma rede DS-TE.

Diferentemente das redes MPLS-TE, onde o cálculo do CSPF é baseado na banda disponível por enlace, no DS-TE, o CSPF deve considerar a banda disponível em cada um dos Tipos de Classe (CTs) disponíveis na rede para a realização do cálculo do caminho de um novo LSP. Portanto, o protocolo CSPF deve levar em consideração a largura de banda definida para o CT específico da nova requisição de LSP quando a computação do caminho for realizada. Por exemplo, para uma requisição de um LSPn (30Mbps) associada ao CT1 ser calculada, todas as larguras de banda disponíveis por CT de cada um dos enlaces da rede devem ser conhecidas. As informações com as larguras de banda associadas a cada um dos CTs devem ser conhecidas por todos os roteadores em caso de um controle descentralizado (cálculo do caminho feito pelos próprios roteadores), ou estar disponível na base de dados na entidade central em caso de um controle centralizado (cálculo do caminho realizado em um computador gerente central no estilo *Bandwidth Broker* (NICHOLS *et al.*, 1999; SCOGLIO *et al.*, 2004; TABATABEE *et al.*, 2005). Será escolhido o melhor caminho dentre os que possuírem os requisitos exigidos pelo LSPn.

### 3.5.2 Estabelecimento do caminho em uma rede DS-TE

Após o cálculo (escolha) do caminho, o mesmo deverá ser estabelecido na rede através de um processo de controle de admissão em cada roteador pertencente ao caminho. É necessária a verificação da largura de banda atual ocupada no CT correspondente ao LSP a ser estabelecido, em cada enlace do caminho escolhido pelo algoritmo de seleção de caminho (MINEI; LUCEK, 2005). Essa verificação ocorre de fato através da troca de mensagens RESV e PATH do protocolo RSVP-TE entre os roteadores do caminho, similar ao que é feito no MPLS-TE, de forma a serem configurados os parâmetros do novo caminho nos roteadores. Em (LE FAUCHER, 2005b) são definidas algumas extensões necessárias que permitem ao protocolo RSVP-TE (AWDUCHE *et al.*, 2001) a possibilidade do estabelecimento de

caminhos considerando a reserva de largura de banda por CT, funcionalidade exigida para a implantação do DS-TE.

A informação do CT para um LSP é transportada em um novo tipo de objeto denominado *Class Type Object* inserido dentro da mensagem PATH do protocolo RSVP-TE. Especificando assim para os roteadores qual o CT no qual a reserva de banda foi requisitada.

### 3.6 MODELOS DE RESTRIÇÃO DE BANDA NAS REDES DS-TE

A solução DS-TE deve garantir diferentes requisitos de restrições de banda para cada um dos CTs configurados. Nesse sentido uma questão fundamental para a operação eficiente do DS-TE é a adoção de um modelo de restrição de banda, modelo *Bandwidth Constraints Model* (BC) para os CTs configurados na rede (LE FAUCHER; LAI, 2003; LE FAUCHER, 2005b; (LAI, 2005).

O modelo de restrição de banda define as regras para o cálculo e utilização da largura de banda em cada enlace para um ou mais tipos de classes (CTs) configurados na rede. Isso permite que, por exemplo, LSPs associados a CTs com tráfego em tempo real tenham uma garantia de banda mínima independente do nível de congestionamento. Outro fator importante é que a adoção de um modelo de restrição de banda evita que LSPs associados a CTs com características elásticas (mais flexíveis) em termos de requisitos de banda não sejam completamente preteridos quando ocorrer a disputa por recursos. Em outras palavras, o número máximo de LSPs admitidos para cada uma diferentes CTs configurados em rede é assegurado pelo modelo de restrição de banda.

Dois modelos de restrições de banda são propostos na literatura, o modelo *Maximum Allocation Model* (MAM) proposto na RFC 4125 (LE FAUCHER; LAI, 2005), e o modelo *Russian Dolls Model* (RDM) proposto pela RFC 4127 (LE FAUCHER, 2005a).

#### 3.6.1 Modelo de restrição de banda MAM

O modelo da máxima alocação de banda (MAM) (LE FAUCHER; LAI, 2005) é o mais intuitivo entre as duas propostas, onde é definida uma restrição de utilização com uma fração

da largura de banda do enlace para cada CT individualmente. Em outras palavras, a largura de banda do enlace é simplesmente dividida entre diferentes CTs configurados em rede, como é mostrado na Figura 15. A soma total da banda ocupada pelos LSPs associados a um determinado CT não pode exceder o valor máximo da sua restrição de banda (BC). A soma total de todos os valores de restrições de banda (BCs) equivale à largura de banda total do enlace.

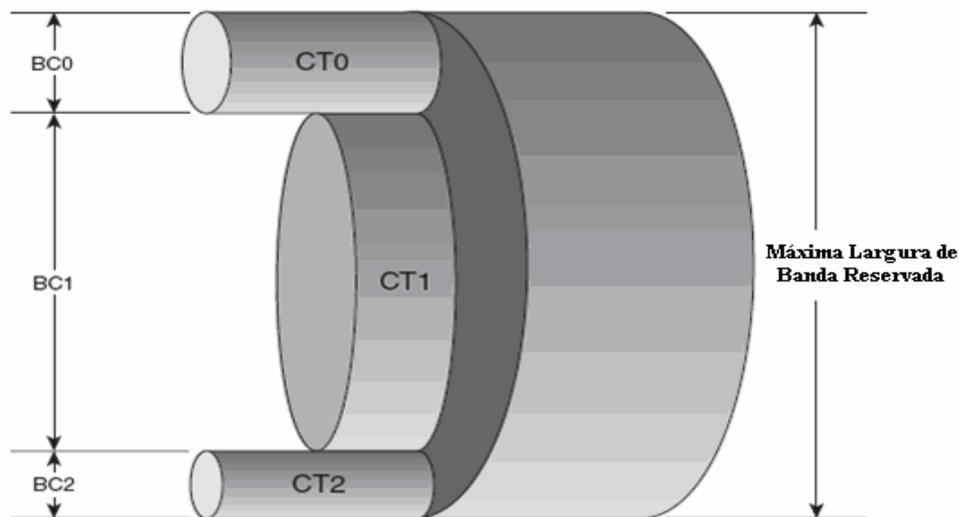


Figura 15 - Alocação de banda em um enlace com o modelo MAM para três CTs (Elaborado pelo autor, 2008)

De acordo com a configuração estabelecida, as restrições de banda do modelo MAM (Figura 15) seguem as seguintes regras:

- a) todos os LSPs associados ao CT2 não podem utilizar uma largura de banda maior do que BC2;
- b) todos os LSPs associados ao CT1 não podem utilizar uma largura de banda maior do que BC1;
- c) todos os LSPs associados ao CT0 não podem utilizar uma largura de banda maior do que BC0.

A recomendação do DS-TE (LE FAUCHER; LAI, 2003) admite um máximo de 8 (oito) BCs configurados em rede ( $BC_i; i=0, \dots, 7$ ), o que corresponde ao número máximo possível de CTs estabelecidos em rede. Uma configuração do modelo MAM considerando o número máximo de 8 BCs configuradas em rede, é apresentada no quadro 8.

<b>Restrição de Banda</b>	<b>Máxima Banda Alocada</b>
BC7	CT7
BC6	CT6
BC5	CT5
BC4	CT4
BC3	CT3
BC2	CT2
BC1	CT1
BC0	CT0

Quadro 8 - Restrições de Banda do Modelo MAM para 8 CTs configurados em rede  
(Elaborado pelo autor, 2008)

A grande vantagem do modelo MAM é que ele isola completamente o tráfego dos diferentes CTs configurados em rede. Conseqüentemente, o uso de prioridades não influencia na preferência do estabelecimento de LSPs carregando tráfego de diferentes CTs.

Na Figura 16 é mostrado um exemplo com uma seqüência de estabelecimentos de LSPs em consonância com o modelo MAM. Neste exemplo, todos os enlaces da rede possuem capacidade de 10 Mbps e estão particionados com 9 Mbps para o CT0 (Tráfego de Dados) e 1 Mbps para o CT1 (Tráfego de Voz).

Dois LSPs de dados (associados ao CT0) requisitam o estabelecimento: LSP1 com 9 Mbps e LSP2 com 1 Mbps. O LSP1 é estabelecido primeiramente no caminho mais curto A-B-C. Como consequência, o LSP2 é forçado a ser estabelecido pelo outro caminho A-D-E-C, devido ao fato do 1 Mbps ainda disponível no caminho A-B-C estar reservado para o tráfego associado ao CT1.

Quando o estabelecimento de um LSP associado ao CT1 (Tráfego de voz) for requisitado, os recursos previstos para o CT1 estarão garantidos ao longo do menor caminho A-B-C, não existindo a possibilidade de disputa por recursos com os LSPs associados ao CT0. A preempção de LSPs entre diferentes CTs não é possível no modelo MAM. Apenas a preempção entre LSPs pertencentes a um mesmo CT é permitida no modelo MAM (LE FAUCHER; LAI, 2005).

Como não existe o compartilhamento de recursos entre CTs, a largura de banda para novas requisições de LSPs disponível para o modelo MAM é calculada por CT da seguinte maneira: a largura de banda disponível para um "CTn" genérico para uma dada prioridade "p", é

calculada subtraindo a totalidade da largura de banda alocada para o "CTn" da largura de banda alocada para todos os LSPs associados ao "CTn" que possuem prioridade menor ou igual a "p" (LE FAUCHER; LAI, 2005). A maior prioridade dentro de um CT está associado aos LSPs com menor valor numérico.

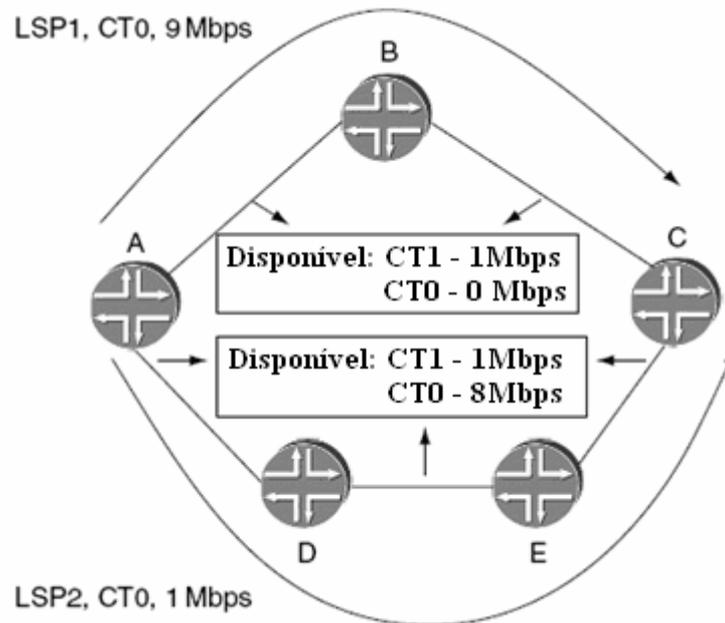


Figura 16 - Exemplo de estabelecimento de LSPs em uma rede DS-TE habilitada com o modelo MAM (Elaborado pelo autor, 2008)

O problema do MAM é que devido às suas características ele não é o mais eficiente em termos de utilização da largura de banda disponível por enlace, já que não permite o compartilhamento de banda não-utilizada (residual) entre CTs distintos. Apesar de promover total isolamento de tráfego entre os CTs configurados em rede, esta menor eficiência na utilização dos recursos é um aspecto ineficiente do ponto de vista da Engenharia de Tráfego. A falta de compartilhamento de banda pode forçar LSPs a seguir caminhos mais longos.

Considerando o exemplo da Figura 16, na ausência de LSPs associados ao CT1 (Tráfego de Voz), existe banda disponível em todos os enlaces do menor caminho que nunca poderá ser utilizada para o estabelecimento de LSPs associados ao CT0 (Tráfego de dados). Um segundo LSP de dados seria forçado a seguir um caminho não ótimo (maior em número de saltos), mesmo que haja banda disponível no menor caminho.

A eficiência na utilização da banda do modelo MAM foi comparada em (LAI, 2005) com o outro modelo de restrição de banda, o *Russian Dolls Model* (RDM) (LE FAUCHER, 2005a), sendo que uma maior eficiência na utilização da banda dos enlaces é atingida no modelo RDM. As características e propriedades do modelo RDM são mostradas em detalhes na subseção a seguir.

### 3.6.2 Modelo de restrição de banda RDM

O princípio básico de funcionamento do modelo RDM (LE FAUCHER, 2005a) é permitir o compartilhamento de banda entre classes de serviço. A banda não utilizada alocada para CTs hierarquicamente mais altos (com maior valor numérico) pode ser utilizada por CTs hierarquicamente mais baixos (com menor valor numérico).

A Figura 17 ilustra um exemplo do modelo RDM configurado com três CTs, onde os valores dos limites de restrição da largura de banda são configurados para um ou mais CTs.

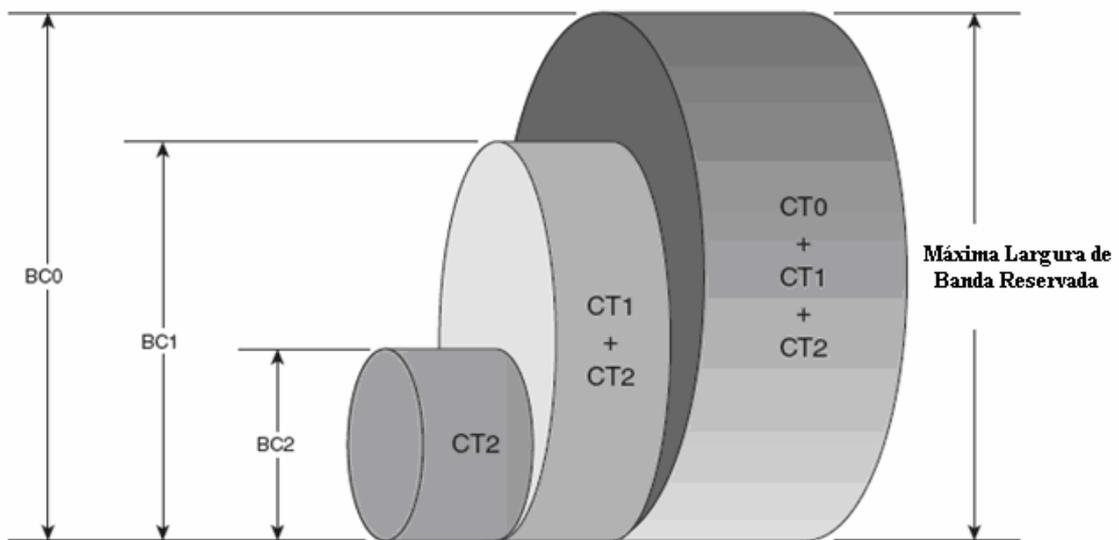


Figura 17 - Modelo RDM e as restrições de banda dos CTs (BCs) (Elaborado pelo autor, 2008)

De acordo com a configuração estabelecida, as restrições de banda do modelo RDM (Figura 17) estariam de acordo com as seguintes regras:

- todos os LSPs associados ao CT2 não podem utilizar uma largura de banda maior do que BC2;
- todos os LSPs associados aos CTs 1 e 2 não podem utilizar uma largura de banda maior do que BC1;

- c) todos os LSPs associados aos CTs 0, 1 e 2 não podem utilizar uma largura de banda maior do que BC0.

Assumindo que as restrições de banda dos enlaces estejam configuradas da seguinte maneira: BC2=50%, BC1=80% e BC0=100%. Neste cenário, caso todos os CTs estivessem utilizando suas máximas configurações de banda permitidas pelo modelo RDM, significaria dizer que: LSPs associados ao CT2 poderiam utilizar 50% da capacidade dos enlaces, LSPs associados ao CT1 não poderiam utilizar mais do que 30% (80%-50%) da capacidade dos enlaces, e LSPs associados ao CT0 não poderiam utilizar mais do que 20% (100%-80%) da capacidade dos enlaces. Entretanto, é muito difícil que simultaneamente todos os CTs configurados em rede estejam operando com suas máximas configurações de utilização de largura de banda. O efeito resultante com a adoção do modelo RDM é que uma maior eficiência na utilização da largura de banda dos enlaces é atingida, devido à possibilidade de compartilhamento de banda entre CTs prevista no modelo RDM.

O modelo de restrição de banda RDM melhora o desempenho em relação ao modelo MAM de acordo com a avaliação feita em (LAI, 2005), devido à uma maior eficiência na utilização da largura de banda dos enlaces. Na avaliação feita o modelo RDM foi considerado mais atrativo em termos de eficiência em termos de utilização dos recursos da rede.

A recomendação do DS-TE indica um máximo de 8 (oito) BCs configurados em rede ( $BC_i$ ;  $i=0, \dots, 7$ ), o que corresponde ao número máximo possível de CTs estabelecidos em rede (LE FAUCHER; LAI, 2003). Uma configuração do modelo RDM considerando o número máximo de 8 BCs configuradas em rede, é exemplificado no quadro 9.

<b>Restrição de Banda</b>	<b>Máxima Banda Alocada</b>
BC7	CT7
BC6	CT7+CT6
BC5	CT7+CT6+CT5
BC4	CT7+CT6+CT5+CT4
BC3	CT7+CT6+CT5+CT4+CT3
BC2	CT7+CT6+CT5+CT4+CT3+CT2
BC1	CT7+CT6+CT5+CT4+CT3+CT2+CT1
BC0	CT7+CT6+CT5+CT4+CT3+CT2+CT1+CT0

Quadro 9 - Restrições de Banda do Modelo RDM para 8 CTs configuradas em rede (Elaborado pelo autor, 2008)

A seguir é mostrado um exemplo (Figuras 18 e 19) com uma sequência de estabelecimentos de LSPs em consonância com o modelo RDM configurado para dois BCs. No exemplo todos os enlaces da rede possuem capacidade de 10 Mbps e estão particionados com BC1=1 Mbps para o CT1 (Tráfego de Voz) e BC0=10 Mbps para o CT0 (Tráfego de Dados) e CT1 (Tráfego de Voz).

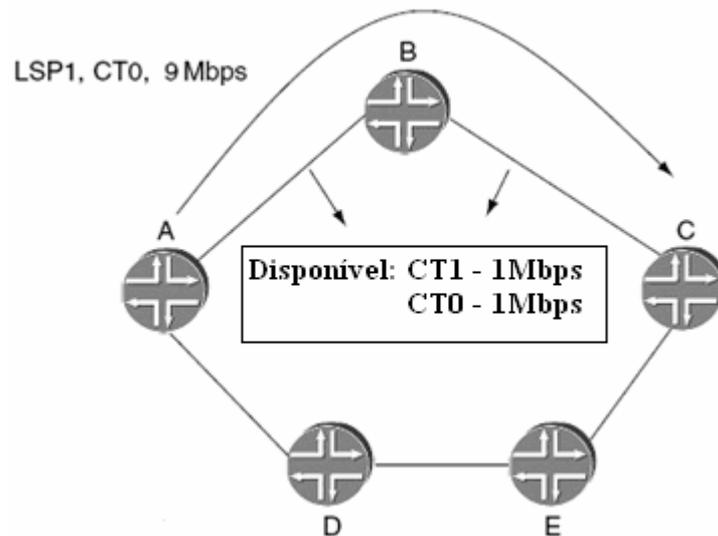


Figura 18 - Estabelecimento de LSP em uma rede DS-TE habilitada com o modelo RDM (Elaborado pelo autor, 2008)

Na Figura 18 1Mbps é alocado para BC1 e 10Mbps é alocado para BC0. Isto significa que cada enlace pode transportar entre 0 e 1 Mbps de tráfego de voz e utilizar o resto para o transporte de dados. O LSP1 associado ao CT0 está inicialmente estabelecida em rede através do menor caminho A-B-C com o valor de 9Mbps. Entretanto no RDM, diferentemente do modelo MAM, o 1Mbps restante nos enlaces do caminho A-B-C podem ser utilizados tanto por novos LSPs associados ao CT0 quanto por novos LSPs associados ao CT1. Portanto, a capacidade de banda disponível para ambos os tipos de classes (CT0 e CT1) é reportada como sendo 1Mbps, como mostrado na Figura 18. Um segundo LSP associado ao CT0 (Tráfego de dados) com um requisito de 1 Mbps poderia ser também estabelecido no menor caminho A-B-C, alocando a capacidade de banda não utilizada (residual) dos enlaces. Isso possibilita otimizar o custo da infra-estrutura de rede, diminuindo as necessidades de superdimensionamento para o atendimento de tráfego de tempo-real como é o tráfego de voz.

No modelo RDM, diferentemente do modelo MAM, não é possível garantir o isolamento de tráfego entre CTs. A preempção deve ser utilizada para garantir que cada CT possua sua

capacidade mínima de banda garantida independentemente do nível de congestionamento dos enlaces (influência do tráfego de outros CTs) de acordo com as configurações especificadas. A necessidade da facilidade do mecanismo da preempção é ilustrada na Figura 19 (continuação do exemplo da Figura 18 anterior).

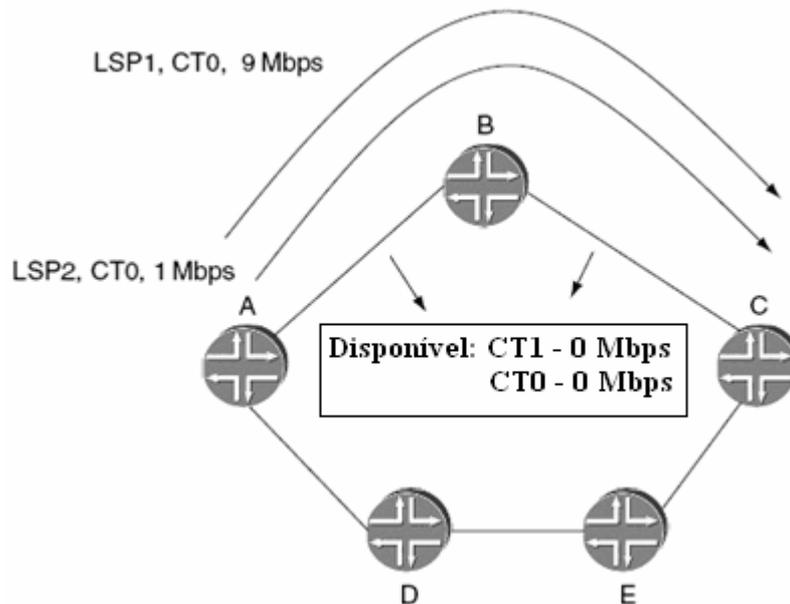


Figura 19 - Ilustração da necessidade da preempção para garantir as configurações do modelo RDM (Elaborado pelo autor, 2008)

Após o estabelecimento de um LSP2 associado ao CT0 (Tráfego de dados) passa a não existir mais capacidade de banda disponível nos enlaces do menor caminho A-B-C. Se for necessário o estabelecimento de um LSP3 associado ao CT1 (Tráfego de Voz) no menor caminho A-B-C, o mesmo não é possível sem que a utilização do mecanismo da preempção seja utilizada para garantia da banda mínima associada ao tipo de classe CT1 em cada enlace. Um LSP3 associado ao CT1 (Tráfego de voz) pode apenas fazer com que um LSP associado ao CT0 (Tráfego de dados) sofra preempção, apenas se os requisitos de banda do LSP3 não exceder a restrição máxima de banda que foi determinada,  $BC1=1\text{Mbps}$ .

Em resumo, na rede DS-TE habilitada com RDM nos exemplos das Figura 18 e 19, é possível que os LSPs associados ao CT0 aloquem banda não utilizada em BC1, devido à possibilidade de compartilhamento de banda prevista na especificação do modelo RDM (LE FAUCHER, 2005a). Entretanto, caso haja disputa de recursos entre CTs em determinado enlace, a preempção de LSPs deve ser implementada visando garantir as configurações e a hierarquia impostas pelo modelo RDM (LE FAUCHER, 2005a).

Na RFC 4127 são apresentadas as características de operação desejadas para uma implementação do modelo RDM, cujos pontos principais são descritos a seguir:

- a) o modelo RDM prevê uma utilização eficiente da largura de banda disponível no enlace com a proteção simultânea contra a degradação dos requisitos de QoS entre CTs.
- b) o recurso de preempção de banda tem que ser utilizado para garantir o isolamento e preservação de largura de banda mínima designada para cada CT independentemente do nível de disputa de recursos existentes entre CTs distintos.
- c) para a utilização da capacidade de compartilhamento de banda entre CTs prevista no RDM, é necessário planejamento e configuração extra: LSPs associados a diferentes tipos de classes devem estar marcados com diferentes prioridades para garantir o compartilhamento de banda nos enlaces.

O quadro 10 a seguir mostra um quadro comparativo com as características dos modelos de restrição de banda MAM e RDM.

<b>MAM</b>	<b>RDM</b>
Mapeia um BC para um CT; fácil de entender e de gerenciar.	Mapeia um BC para um ou mais CTs; mais difícil de gerenciar.
Alcança o isolamento entre CTs e a garantia de largura de banda para os CTs sem a necessidade da preempção.	Não possui isolamento entre CTs; requer e preempção para garantir a largura de banda para garantir a banda dos CTs que não seja o CT com maior valor numérico.
A eficiência na utilização da largura de banda e a proteção contra degradação de QoS são mutuamente exclusivas.	Fornece eficiência na utilização da largura de banda e proteção contra degradação de QoS simultaneamente.

Quadro10 - Comparativo entre os modelo MAM e RDM (Elaborado pelo autor, 2008)

## 4 FORMALIZAÇÃO DOS PROBLEMAS ABORDADOS

Neste capítulo é apresentada uma revisão bibliográfica e o estado da arte relativos ao contexto da utilização da arquitetura DS-TE em conjunto com o modelo de restrição de banda RDM. Dois problemas são objetivos da apresentação, a necessidade de gerência de banda compartilhada entre CTs e a escolha de caminhos para LSPs no contexto do modelo RDM.

### 4.1 GERÊNCIA DE BANDA COMPARTILHADA ENTRE CTS EM REDES DS-TE

Dentro das perspectivas de QoS e Engenharia de Tráfego, quando uma nova requisição de LSP chega a uma rede habilitada com DS-TE faz-se necessário procurar um caminho na rede (*path*) através de um algoritmo de seleção de caminho baseado em restrições. Um algoritmo de seleção de caminho baseado em restrições calcula o caminho (sequências de saltos a ser seguida) que o LSP vai seguir na rede desde o nó de origem até o nó de destino, visando atender a uma ou mais restrições exigidas.

Em (MINEI; LUCEK, 2005), é discutida a possibilidade de aplicação do *Constrained Shortest Path First* (CSPF) para o cálculo de caminhos de LSPs em redes DS-TE, tendo como restrição a banda mínima exigida pelo LSP e como métrica a ser minimizada o número de saltos entre o nó de origem e o nó de destino.

Existem trabalhos na literatura com a implementação do CSPF considerando apenas a banda do enlace como um todo como restrição (XIPENG *et al.*, 2000; SZETO *et al.*, 1997; OLIVEIR; MATEUS, 2006). Estes trabalhos não consideram a diferenciação por tipo de classe, o que não os torna aplicáveis para uma rede multiclasse como o DS-TE.

Outros trabalhos anteriores abordam os algoritmos para seleção de caminhos dentro do contexto multiclasse (MINGYING *et al.*, 2006; SZVIATOVSKY *et al.*, 2002). Estas propostas implementam o CSPF e avaliam o seu desempenho para operar com múltiplas classes, onde o cálculo da banda disponível é realizado por classe e não apenas por enlace. A partir da chegada de uma nova requisição de LSP, é selecionado o caminho que possua a menor quantidade de saltos, e que ao mesmo tempo tenha banda disponível na classe correspondente em todos os enlaces. Cada caminho é testado por ordem crescente de número

de saltos. E, se existir banda correspondente para ser utilizada na classe associada à nova requisição em todos os enlaces do caminho, o mesmo é selecionado. Se em nenhum dos caminhos existentes houver banda disponível na classe ao qual a nova requisição está associada, a mesma é bloqueada.

Os algoritmos (KE *et al.*, 2004; MINGYING *et al.*, 2006; XIAOMING *et al.*, 2006; SZVIATOVSKY *et al.*, 2002) implementam a possibilidade de preempção de LSPs para redes multiclasse com base nas prioridades de estabelecimento (“*setup*”) e manutenção (“*hold*”), oferecendo a possibilidade de priorização de LSPs em caso de disputa por recursos. Adicionalmente em (MINGYING *et al.*, 2006; XIAOMING *et al.*, 2006) também é implementada a mais nova recomendação de política de preempção de LSPs definida na recomendação RFC 4829 (DE OLIVEIRA *et al.*, 2007), detalhada na seção 3.4. A implementação da política de preempção é importante em redes multiclasse, que implementem níveis distintos de prioridade, e é aplicada quando o algoritmo de cálculo de caminho não encontra um caminho disponível para requisições mais prioritárias de LSP com as características mínimas de largura de banda exigidas em todos os seus enlaces.

Como resultado da pesquisa bibliográfica realizada, constatou-se que as propostas existentes não contemplam uma relação entre o algoritmo de cálculo de caminho com a utilização conjunta do modelo de restrição de banda RDM recomendado. Com efeito, as soluções contemplam apenas a seleção de caminhos considerando o aspecto multiclasse presente em redes DS-TE (KE *et al.*, 2004; MINGYING *et al.*, 2006; XIAOMING *et al.*, 2006; SZVIATOVSKY *et al.*, 2002), não considerando a existência de modelos de restrições de banda (BCs) e a possibilidade de compartilhamento de recursos entre tipos de classes (CTs) distintos presente no modelo RDM.

O modelo de restrição de banda RDM define as regras de utilização de banda (BCs) nos enlaces e possibilita o compartilhamento de banda entre CTs que estiverem definidos na rede. A utilização do modelo de restrição de banda RDM é um diferencial importante na arquitetura DS-TE, possibilitando uma maior eficiência na utilização dos recursos da rede.

Portanto, para a utilização do modelo RDM e a operação adequada em redes que utilizem a arquitetura DS-TE, é preciso que o algoritmo de seleção de caminhos esteja não só ciente da possibilidade de múltiplas classes, como também é fundamental que esteja ciente da forma

como a largura de banda está configurada e utilizada para cada um dos CTs em rede, através do conhecimento das configurações modelo de restrição de banda RDM no momento do cálculo do caminho.

A seguir é mostrado um exemplo numérico que ilustra a necessidade do conhecimento das configurações do modelo RDM na seleção de caminhos de LSPs em uma rede DS-TE. O quadro 11 mostra a largura de banda e o CT correspondente para um conjunto de oito LSPs estabelecidos em dado momento para um determinado enlace “A” hipotético de 300 Mbps. O enlace “A” está configurado com três CTs (CT0, CT1 e CT2). Os valores de restrições de banda (BCs) com estes oito LSPs estabelecidos são os seguintes: BC2 (CT2)=40% (120 Mbps), BC1 (CT1 + CT2)=70% (210 Mbps) e BC0 (CT0 + CT1 + CT2)=100% (300 Mbps).

<b>LSP</b>	0	1	2	3	4	5	6	7
<b>CT</b>	2	1	1	1	0	0	0	0
<b>Bandwidth</b>	80	40	50	30	30	30	20	20

Quadro 11 - Configuração dos LSPs estabelecidos<sup>4</sup> (Elaborado pelo autor, 2008)

Supondo que uma nova requisição de estabelecimento de LSP (LSP8 com largura de banda = 30Mbps) chegue associado ao CT2 (CT de mais alta hierarquia no exemplo de modelo RDM configurado no enlace “A”). Essa nova requisição de LSP passaria pelo enlace “A” devido ao fato desse enlace pertencer ao melhor caminho calculado pelo algoritmo de cálculo de caminho baseado em restrições utilizado. O quadro 12 a seguir ilustra qual seria a configuração resultante dos BCs antes e depois do possível estabelecimento da nova requisição.

<b>BC</b>	<b>Antes do “LSP8”</b>	<b>Máximo valor aceitável pelo modelo RDM</b>	<b>Depois do “LSP8”</b>
<b>2</b>	26,67%	40,00%	<b>36,67%</b>
<b>1</b>	66,67%	70,00%	<b>76,67%</b>
<b>0</b>	100,00%	100,00%	<b>110,00%</b>

Quadro 12 - Configuração dos BCs antes e depois da nova requisição (Elaborado pelo autor, 2008)

Como pode ser visto no quadro 12 acima (destacado em verde), o LSP8 poderia ser estabelecido pois existe capacidade disponível para o CT2 (capacidade disponível para o CT2 = 40 Mbps → 120 Mbps relativo à máxima configuração de banda RDM menos os 80 Mbps

<sup>4</sup> No exemplo, a utilização de BC0 foi configurada para ser igual à máxima capacidade do enlace (100%), no intuito de forçar a necessidade de preempção de banda entre CTs após novas requisições de estabelecimento de LSPs (ou seja, forçar a necessidade da utilização de uma estratégia de gerência de banda compartilhada).

já alocados para o LSP0). Entretanto (destacado em vermelho), BC1 e BC0 teriam seus valores máximos especificados ultrapassados com o estabelecimento da nova requisição. Isso aconteceria na prática devido aos LSPs associados aos CTs 0 e 1 estarem utilizando momentaneamente a largura de banda alocada para o CT2, devido à possibilidade de compartilhamento de banda prevista no modelo RDM.

A preempção de LSPs é um recurso adotado em caso da ocorrência da disputa de recursos em redes. Para respeitar as configurações de restrições de banda (BCs) do RDM é necessário que ela ocorra corretamente, nos CTs que façam parte da restrição de banda (BC) que esteja sendo desrespeitada. Caso não houvesse preempção nos CTs hierarquicamente inferiores, a nova requisição não poderia ser estabelecida (seria bloqueada), já que o valor máximo da largura de banda do enlace seria ultrapassado.

Portanto, é necessário que no momento do cálculo do caminho, as informações referentes às restrições de banda (BCs) do modelo RDM estejam disponíveis para cada enlace da rede DS-TE. Também faz-se necessário o conhecimento do tipo de classe à qual cada LSP estabelecido está associado. A partir dessas informações, seria possível realizar-se uma gerência da largura de banda compartilhada nos tipos de classes adequados (CT1 e CT0), no intuito de garantir o estabelecimento da nova requisição de maior prioridade (maior hierarquia em termos de CT), e conseqüentemente a preservação das configurações do modelo RDM.

A avaliação da possibilidade de uma estratégia de gerência da banda compartilha entre CTs que possibilite a interface com a seleção de caminhos para novos LSPs em uma rede DS-TE, deve, como medida de eficiência, permitir a utilização simultânea das propriedades de compartilhamento e proteção de tráfego do modelo RDM, permitindo uma melhor utilização da largura de banda dos enlaces, independentemente do nível de disputa por recursos nos mesmos. As seguintes etapas seriam necessárias:

- a) escolher um caminho com o menor número de saltos ou com outro tipo de métrica a ser minimizada,
- b) que possua a banda exigida disponível no CT correspondente à nova requisição (LSP) em todos os seus enlaces,
- c) que não infrinja as regras de máxima utilização (BCs) de banda dentro da configuração do modelo RDM adotada,

- d) que respeite a largura de banda mínima alocada para cada CT, e
- e) que a propriedade de compartilhamento de banda entre CTs prevista no modelo RDM possa ser utilizada.

#### 4.2 OCORRÊNCIA DESNECESSÁRIA DE PREEMPÇÕES

Outro problema importante que deve ser considerado nas redes DS-TE é a relação entre o caminho definido e o número de preempções resultante dessa escolha. Com efeito, a escolha de caminhos apenas baseado na menor quantidade de saltos e banda disponível por classe das implementações encontradas na literatura (MINGYING *et al.*, 2006; XIAOMING *et al.*, 2006; SZVIATOVSKY *et al.*, 2002), pode vir a ter baixa eficiência quando se trata de uma rede DS-TE habilitada com o modelo RDM.

A utilização do CSPF tendo como restrição a largura de banda do LSP e como métrica minimizar a quantidade de saltos em uma rede habilitada com o RDM, pode vir a causar preempções desnecessárias na rede visando acomodar uma nova requisição de LSP de determinado CT, considerando as propriedades de compartilhamento de banda entre CTs presente no modelo RDM.

A mera escolha com base no menor caminho em termos de saltos que satisfaça as restrições de banda do LSP não leva em consideração a possibilidade de CTs hierarquicamente inferiores estarem utilizando momentaneamente a banda do CT correspondente à nova requisição. O caso mais crítico ocorre quando os enlaces do caminho escolhido possuírem uma grande quantidade de banda compartilhada entre o CT do novo LSP e os CTs hierarquicamente inferiores à mesma.

Em outras palavras, a nova requisição de LSP associado a um determinado CT pode vir a ter o menor caminho escolhido, só que para tal pode vir a estar sacrificando de forma desnecessária o tráfego de outros LSPs já estabelecidos associados a CTs hierarquicamente menos prioritárias de acordo com a configuração do RDM adotada. A avaliação da possibilidade da escolha de um caminho alternativo ao menor caminho, deve, como medida de eficiência, minimizar o impacto da ocorrência de preempções em rede.

A Figura 20 a seguir mostra um exemplo em uma topologia de rede hipotética no qual a seleção de caminho baseada na menor quantidade de saltos pode vir a ser ineficiente em uma rede DS-TE configurada com o modelo RDM:

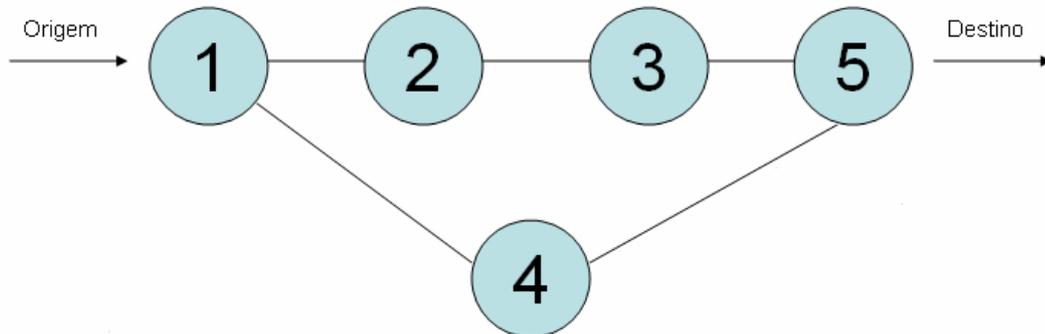


Figura 20 - Seleção do Caminho X Ocorrência de Preempção (Elaborado pelo autor, 2008)

Na topologia de rede da Figura 20, os enlaces estão configurados com o modelo RDM da seguinte maneira: BC0=100%, BC1=70% e BC2=50%. Todos os enlaces da rede possuem 100 Mbps. Uma ocorrência desnecessária de preempção ocorre na seguinte situação:

- a) os LSPs de 1 a 9 estão estabelecidos em rede por dois caminhos distintos, de acordo com o quadro 13 a seguir;

	CT	Mbps	Caminho
LSP0	0	10	1->4->5
LSP1	0	20	1->4->5
LSP2	1	20	1->4->5
LSP3	1	20	1->4->5
LSP4	2	20	1->4->5
LSP5	0	10	1->2->3->5
LSP6	0	20	1->2->3->5
LSP7	1	10	1->2->3->5
LSP8	1	20	1->2->3->5
LSP9	2	20	1->2->3->5

Quadro 13 - Configurações dos LSPs Pré-estabelecidos (Elaborado pelo autor, 2008)

- b) um novo LSP10=20Mbps associado ao CT2 solicita o estabelecimento em rede;
- c) se o menor caminho (1-4-5) for selecionado necessariamente ocorreria preempção de LSPs para os limites BC0 e BC1 não serem ultrapassados nos enlaces, como mostra o quadro14 a seguir. Preempções de LSPs associados aos CTs 0 e 1 teriam que necessariamente ocorrer, para que as regras do modelo RDM fossem respeitadas;

<b>Enlaces do Caminho 1-&gt;4-&gt;5</b>			
	<b>Antes do LSP10</b>	<b>Máximo</b>	<b>Depois do LSP10</b>
<b>BC0</b>	90%	100%	110%
<b>BC1</b>	60%	70%	80%
<b>BC2</b>	20%	50%	40%

Quadro 14 - Configurações das BCs dos Enlaces do Menor Caminho (Elaborado pelo autor, 2008)

- d) caso o caminho alternativo fosse o escolhido (1-2-3-5) nenhuma preempção seria necessária para o estabelecimento da nova requisição (LSP10) associada ao CT2, como mostra a quadro 15 a seguir.

<b>Enlaces do Caminho 1-&gt;2-&gt;3-&gt;5</b>			
	<b>Antes do LSP10</b>	<b>Máximo</b>	<b>Depois do LSP10</b>
<b>BC0</b>	80%	100%	100%
<b>BC1</b>	50%	70%	70%
<b>BC2</b>	20%	50%	40%

Quadro15 - Configurações das BCs dos Enlaces do Maior Caminho (Elaborado pelo autor, 2008)

No exemplo mostrado, o caminho mais longo (alternativo ao menor caminho) poderia ser escolhido e, apesar de ser maior em número de saltos, atende à requisição de banda associada ao CT2 sem a necessidade de nenhuma preempção ser efetuada na rede. Apesar de não ser geralmente uma situação real, a entrada de tráfego apenas por um roteador da rede (normalmente existe tráfego de interferência entrando na rede a partir de outros nós), é possível perceber a partir dos cálculos do exemplo que a escolha do menor caminho em termos de saltos pode vir a trazer um número de preempções indesejadas em uma rede DS-TE que utilize o modelo RDM.

## 5 APRESENTAÇÃO DAS SOLUÇÕES PROPOSTAS: ALGORITMOS ADAPT-RDM, CSPF E MINPREPTDSTE

Neste capítulo serão apresentadas propostas de soluções para os problemas descritos no capítulo anterior, dentro do contexto das redes DS-TE que utilizam o modelo de restrição de banda RDM. Os problemas apresentados foram os seguintes:

- a) necessidade de gerenciar a banda compartilhada entre CTs, visando garantir as configurações do modelo RDM independentemente do nível de tráfego na rede;
- b) falta de integração de uma estratégia de gerência de banda compartilhada entre CTs, com os algoritmos de seleção de caminhos de LSPs; e
- c) possibilidade de ocorrência de uma quantidade desnecessária de preempções, devido ao algoritmo de seleção de caminhos não levar em consideração a propriedade de compartilhamento de banda entre CTs do modelo RDM.

As soluções são propostas na forma de algoritmos, da seguinte maneira:

- a) um algoritmo com uma estratégia de gerência da banda compartilhada entre CTs para o modelo RDM, com base na preempção de LSPs, de forma que a preempção de LSPs ocorra corretamente nos CTs que têm suas restrições de banda (BCs) desrespeitadas no momento da disputa por recursos na rede;
- b) a integração da estratégia de gerência de banda compartilhada proposta com um algoritmos de seleção de caminhos de LSPs baseado em restrições; e
- c) um algoritmo para a seleção de caminho de LSPs que tenta minimizar a quantidade de preempções ocorridas em rede, tornando a seleção de caminhos ciente das configurações do modelo RDM configurado.

### 5.1 CARACTERÍSTICAS GERAS DAS SOLUÇÕES PROPOSTAS

Os algoritmos propostos têm as suas operações baseadas no controle centralizado (SCOGLIO *et al.*, 2004; TABATABEE *et al.*, 2005), no estilo *bandwidth broker* (NICHOLS *et al.*, 1999), a partir de uma entidade gerenciadora, como ilustrado na Figura 21.

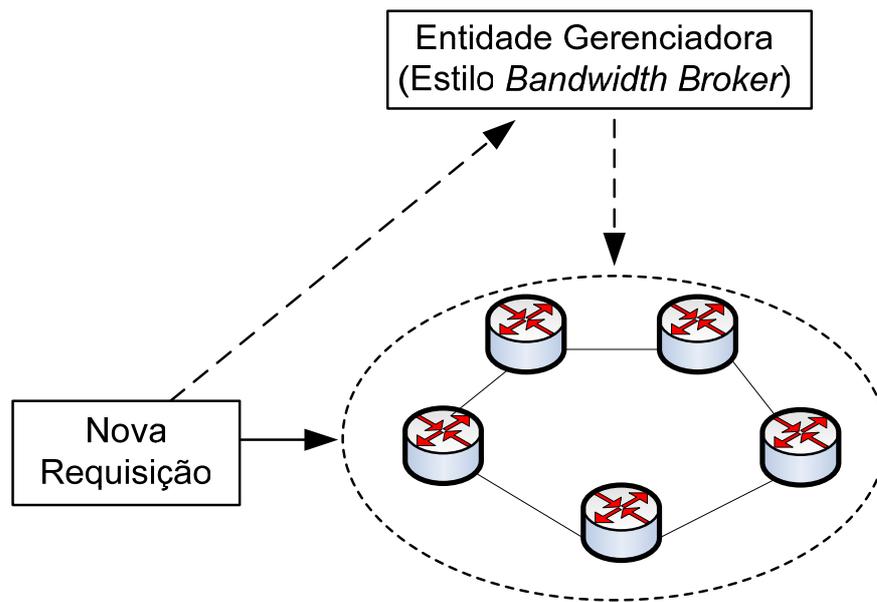


Figura 21 - Gerência Centralizada de um Domínio DS-TE (Elaborado pelo autor, 2008)

Para a aplicação das propostas de algoritmos, considerando que os mesmos possuem características centralizadoras, é necessário que uma entidade gerenciadora tenha o conhecimento da topologia da rede, dos LSPs que estão estabelecidas e dos parâmetros do modelo RDM a serem configurados em rede. Dessa forma a entidade gerenciadora possui as informações necessárias para saber se uma nova requisição de LSP pode ou não encontrar um caminho para se estabelecer na rede.

A entidade gerenciadora (gerenciadora da banda dos enlaces e CTs configurados em rede) é a responsável pela execução dos algoritmos propostos, e deve possuir uma estrutura de armazenamento de LSPs, as restrições de banda (BCs), e um conjunto de variáveis de controle em consonância com as configurações do modelo RDM armazenadas em sua base de dados.

Para o controle da manutenção das restrições de banda (BCs) especificadas para os CTs em rede, a base de dados da entidade gerenciadora contém os LSPs estabelecidos em rede agrupados por CT, em cada enlace aos quais os mesmos estão associados, conforme a estrutura ilustrada na Figura 22.

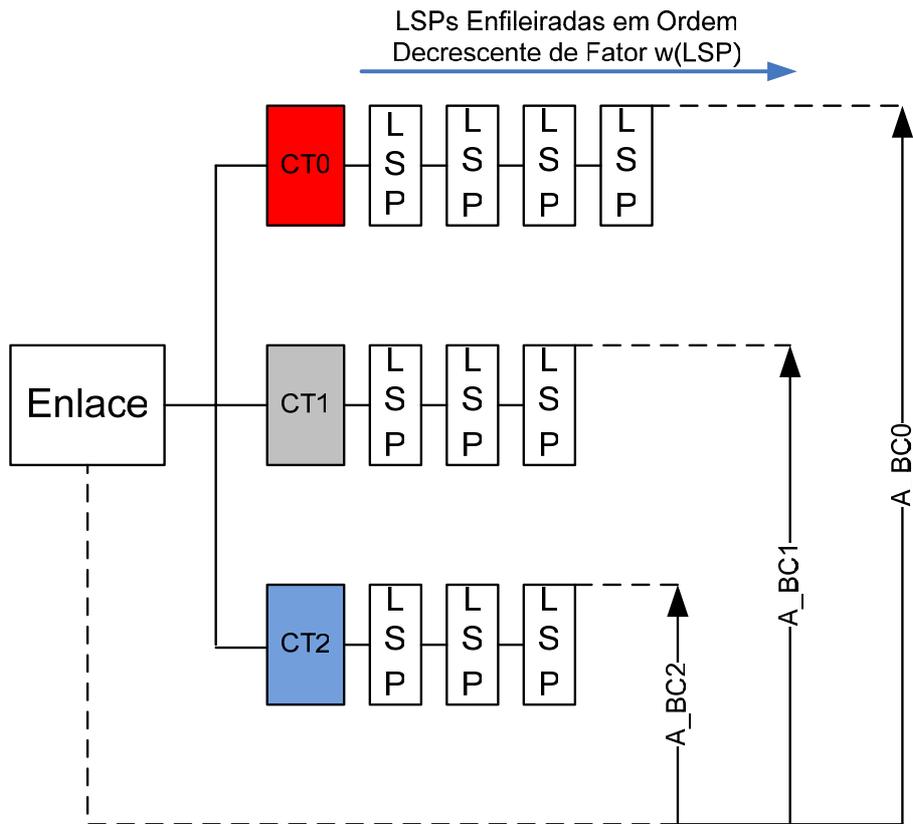


Figura 22 - LSPs agrupados por CT na base de dados da Entidade Gerenciadora (Elaborado pelo autor, 2008)

A base de dados da entidade gerenciadora possui uma lista de LSPs estabelecidos na rede, denominada de *Estab\_List*, que está estruturada da seguinte maneira:

- a) cada enlace da rede possui uma lista de CTs que estão configurados neles;
- b) cada CT possui uma lista com os LSPs associados a eles;
- c) os LSPs são ordenados em ordem decrescente com base em um fator, denominado fator " $w(lsp)$ ", apresentado na próxima seção;
- d) os LSPs que passarem por mais de um enlace da rede, estarão associados aos CTs correspondentes em cada um dos enlaces;

Um conjunto de variáveis de controle estão armazenadas na entidade gerenciadora visando o cálculo de caminhos e a gerência da largura de banda compartilhada entre CTs, as quais são listadas no quadro 16.

Variável	Descrição
A_BC[n]	Vetor com o valor atual de banda ocupada por uma restrição de banda BC, existente para cada enlace da rede. Onde “n” pode variar de 0 (zero) até o número máximo de CTs configurados em rede.
MAX_BC[n]	Vetor com o valor máximo permitido para uma restrição BC, existente para cada enlace da rede. Onde “n” pode variar de 0 (zero) até o número máximo de CTs configurados em rede.
New_LSP_Band	Banda exigida por uma nova requisição de LSP
Max_Caminhos_Disponíveis	Número máximo de caminhos disponíveis entre a origem e o destino de uma nova requisição de LSP

Quadro 16 - Variáveis de Controle da Entidade Gerenciadora (Elaborado pelo autor, 2008)

Cada par origem-destino possui associado a ele uma matriz de caminhos disponíveis armazenada na base de dados da entidade gerenciadora, de acordo com o formato do quadro 17. Na implementação, os enlaces são representados por letras e os caminhos por números. Os caminhos entre um par origem-destino de LSP está ordenado em ordem crescente de número de saltos na matriz de caminhos.

Caminho	Enlaces			
1	B	C	-	-
2	A	E	F	-
3	B	C	G	H

Quadro 17 - Estrutura da Matriz de Caminhos para um par Origem-Destino de LSP (Elaborado pelo autor, 2008)

Considerando uma implementação prática e questões de escalabilidade, a abordagem centralizada adotada para os algoritmos implementados faz com que eles sejam mais adequados para topologias de rede de pequeno e médio porte (tais como *backbones* de redes corporativas), com um conjunto de classes de serviço bem definidas. Detalhes sobre questões de escalabilidade quando se adota um controle centralizado para a Engenharia de Tráfego estão fora do propósito deste trabalho.

## 5.2 ALGORITMO PARA GERENCIAMENTO DE BANDA COMPARTILHADA ENTRE CTS PARA O MODELO RDM

Esta seção descreve a proposta de um algoritmo com uma estratégia de gerência de banda compartilhada entre CTs. O algoritmo de verificação e gerência de banda compartilhada entre CTs denominado **ADAPT-RDM** (PINTO NETO; MARTINS, 2008a; PINTO NETO; MARTINS, 2008b) é proposto.

O **ADAPT-RDM** garante que os valores de restrições de banda (BCs) e a propriedade de compartilhamento prevista no modelo RDM estejam sempre sendo respeitados de acordo com o especificado, independentemente do nível de disputa por recursos entre os LSPs das diferentes CTs configurados em rede.

O **ADAPT-RDM** segue a recomendação do modelo RDM, onde os CTs com maior valor numérico possuem maior prioridade ou nível de proteção de tráfego com relação aos CTs com menor valor numérico em situações de disputa por recursos na rede.

O **ADAPT-RDM** se baseia na política de preempção especificada em (DE OLIVEIRA *et al.*, 2007), em caso da necessidade da preempção de LSPs em rede visando a adequação as restrições de banda BCs configuradas para as diferentes CTs em rede.

Após a escolha do novo caminho pelo algoritmo de seleção de caminhos, o **ADAPT-RDM** verifica nos enlaces do caminho escolhido a necessidade ou não de preempção de LSPs em CTs hierarquicamente inferiores, que possam vir a estar utilizando a banda não-utilizada momentânea do CT hierarquicamente superior (CT ao qual a nova requisição de LSP está associada). Essa verificação é feita independentemente do critério utilizado para o cálculo do caminho por parte do algoritmo de cálculo de caminho baseado em restrição que for adotado, e somente após a sua execução é que a autorização para o estabelecimento propriamente dito do LSP pode ser realizado (o estabelecimento do LSP é realizado via algum protocolo de sinalização ciente da Engenharia de Tráfego, como o RSVP-TE).

Em caso do limite de alguma BC ser excedida, a preempção de LSPs é acionada de acordo com a seguinte estratégia:

- O algoritmo checa a necessidade de preempção de banda nos BCs de acordo com a ordem decrescente de tamanho (de BC[n-1], BC[n-2], ..., até BC[0]). Dessa forma, tenta-se prevenir/evitar a preempção de mais largura de banda do que a necessária devido às configurações do modelo RDM, onde um BC de maior tamanho contém uma BC de menor tamanho.

Em paralelo à obtenção de uma melhor eficiência da utilização da banda dos enlaces da rede com a utilização da propriedade de compartilhamento do RDM, também é objetivo do **ADAPT-RDM** o respeito às máximas configurações por parte dos CTs de cada uma das

restrições de banda (BCs) configuradas em rede. Dessa forma, é possível atingir simultaneamente eficiência na utilização da banda dos enlaces e proteção do tráfego dos CTs mais prioritários respeitando as BCs configuradas no modelo RDM.

O algoritmo **ADAPT-RDM** deve ser utilizado em conjunto com o algoritmo de seleção de caminhos para LSPs baseado em restrições, como por exemplo, o algoritmo do menor caminho baseado em restrições, visando tornar a escolha do caminho ciente da propriedade de compartilhamento existente no modelo RDM.

O algoritmo de seleção de caminhos sempre está ciente das configurações de banda (BCs) associadas aos CTs do modelo RDM, tornando possível a consideração da utilização da propriedade de compartilhamento de banda entre CTs prevista na sua recomendação do modelo RDM como critério para a escolha do melhor caminho. Assim um LSP associado a um CT menos prioritário dentro da hierarquia RDM pode vir a ser estabelecido no melhor caminho, caso não haja nenhum outro LSP associado a um CT mais prioritário já estabelecido no mesmo. O algoritmo ADAPT-RDM realiza os ajustes necessários (gerencia a banda compartilhada entre CTs) visando disponibilizar a banda não-utilizada por CTs mais prioritários para CTs menos prioritários.

Por outro lado, uma nova requisição de LSP associado a um LSP mais prioritário, tem a garantia de que poderá ser estabelecida pelo melhor caminho, mesmo que haja um outro LSP associado a um CT menos prioritário já estabelecida no mesmo. Para tal, o algoritmo ADAPT-RDM realiza os ajustes necessários (gerencia a banda compartilhada entre CTs) visando disponibilizar a banda necessária para um LSP associado a um CT mais prioritário na hierarquia do modelo RDM, e de uma forma que nenhuma restrição de banda (BC) seja desrespeitada para tal.

A Figura 23 mostra um pseudo-código descrevendo a estratégia do ADAPT-RDM. O algoritmo recebe como parâmetro o tipo de classe (**CT**), a largura de banda (**New\_LSP\_Band**), e o caminho escolhido pelo algoritmo de seleção de caminho (**New\_LSP\_Path**) no qual o novo LSP será estabelecido. Na linha 1 são inicializadas três variáveis auxiliares. A primeira é um vetor que armazena temporariamente a quantidade de banda “preemptada” em cada enlace do caminho (**Band\_Preempted[nEnl]**) durante a execução do algoritmo. A segunda variável armazena temporariamente a quantidade de banda

acima de cada restrição de banda (BC) do modelo RDM (**BandUP**), para cada necessidade de verificação do algoritmo. A última variável é um *flag* (**prept**) que é utilizado para indicar se foi necessário ou não a ocorrência de preempção para atender a nova requisição de LSP.

```

Alg. ADAPT-RDM (CT, New_LSP_Band, New_LSP_Path)
1.  Band_Preempted[nEnl]=0, BandUP=0; Prept=0;
2.  Calc_w(LSP)_Factor (A,B);
3.  For all Links in New_LSP_Path
4.  {
5.    For(n= New_LSP_CT-1;n≥0;n--)
6.    {
7.      BandUP = A_BC[n]- MAX_BC[n];
8.      If(BandUP>0)
9.      {
10.        Prept=1;
11.        LSP_Preemption (BandUP, nEnl, n);
12.      }
13.    }
14.  }
15.  return (prept);

```

Figura 23 - Pseudo-código do algoritmo ADAPT-RDM (Elaborado pelo autor, 2008)

Na linha 2 o fator “ $w(lsp)$ ” é calculado para a nova requisição de LSP, com o intuito de definir a ordem na qual o novo LSP a ser estabelecido será armazenado no seu CT correspondente na base de dados da entidade gerenciadora (*Estab\_List*). Esse fator é utilizado para selecionar quais LSPs têm preferência para serem “preemptadas” quando ocorre a disputa por recursos entre LSPs de diferentes CTs. O fator é calculado baseado em dois parâmetros: a prioridade e a largura de banda reservada do LSP, como mostrado na Equação (2).

$$(2) \quad w(lsp) = A.p(lsp) + B.b(lsp).$$

Na equação (2), “ $p(lsp)$ ” é a prioridade do LSP, e está associada com a necessidade de preemptar preferencialmente aos LSPs de baixa prioridade de determinado CT (o intuito é permitir a utilização de diferentes níveis de prioridade dentro de um mesmo CT. “ $b(lsp)$ ” é a largura de banda reservada por cada LSP estabelecido na rede, e está associada ao fato de se preemptar LSPs com pequena largura de banda. Os LSPs selecionados para preempção em determinado CT são aquelas com o menor fator “ $w(lsp)$ ” para acomodar uma nova requisição associada a um CT hierarquicamente superior de acordo com o modelo RDM. “ $A$ ” e “ $B$ ” são

pesos que podem ser configurados, a depender da necessidade do gerente da rede. Por exemplo, para  $A=1$  e  $B=0$ , significa que apenas a prioridade dos LSPs é importante quando há necessidade de ocorrência de preempção.

Nas linhas 3-14 é verificada a necessidade de ajustar os BCs do modelo RDM para acomodar a nova requisição de LSP, se algum dos BCs tiverem seus valores máximos excedidos. Na linha 7 é calculada a quantidade de banda que está acima do valor limite de cada BC. Na linha 11 a preempção de LSPs de acordo com o fator “ $w(lsp)$ ” é executada nos CTs para adequar os valores das restrições de banda (BCs) que teriam seus valores ultrapassados pós-estabelecimento da nova requisição. A função que executa a preempção é para a retirada do(s) LSP(s) a ser(em) preemptado(s) da lista de LSPs estabelecidos (*Estab\_List*) na base de dados, e recebe como parâmetros o total de banda a ser preemptada (*BandUP*), o número atual do enlace que está sendo testado/avaliado (*nEnl*), e o CT correspondente aos quais os LSPs a serem preemptados estão associadas (*n*).

Os LSPs que sofrem preempção são retirados da lista de LSP estabelecidos mantidas na entidade gerenciadora (*Estab\_List*) por CT em ordem decrescente de fator “ $w(lsp)$ ”. Ao final, na linha 15, um flag (*prept*) é retornado informando se foi necessária a preempção de algum LSP para acomodar a nova requisição.

### 5.3 ALGORITMO PARA SELEÇÃO DE CAMINHOS BASEADO NO MENOR NÚMERO DE SALTOS CIENTE DO RDM

Esta seção apresenta uma proposta de algoritmo baseado na implementação do algoritmo de cálculo do melhor caminho baseado em restrições (CSPF) em consonância com o modelo de restrição de banda RDM. O algoritmo busca um caminho que atenda ao requisito de banda do LSP e que minimize a quantidade de saltos entre o nó origem e o nó de destino. O algoritmo baseia-se no controle centralizado e utiliza a proposta do ADAPT-RDM para torná-lo aplicável a uma rede DS-TE habilitada com o modelo RDM.

A seguir, na Figura 24, é mostrado um fluxograma com a idéia do funcionamento do algoritmo de cálculo do melhor caminho baseado em restrições (CSPF) integrado com a estratégia ADAPT-RDM. Após a chegada de uma nova requisição de LSP, o algoritmo de seleção de caminhos busca pelo caminho com a menor quantidade de saltos. Caso exista pelo

menos um caminho que respeite a restrição de banda (BC) exigida pelo LSP, o algoritmo ADAPT-RDM é acionado para fazer a gerência da banda compartilhada entre classes, com a finalidade de garantir os parâmetros e propriedades do modelo de restrição de banda RDM configurado na rede. Caso contrário, se não houver nenhum caminho com a restrição de banda (BC) exigida, a requisição de LSP é bloqueada.

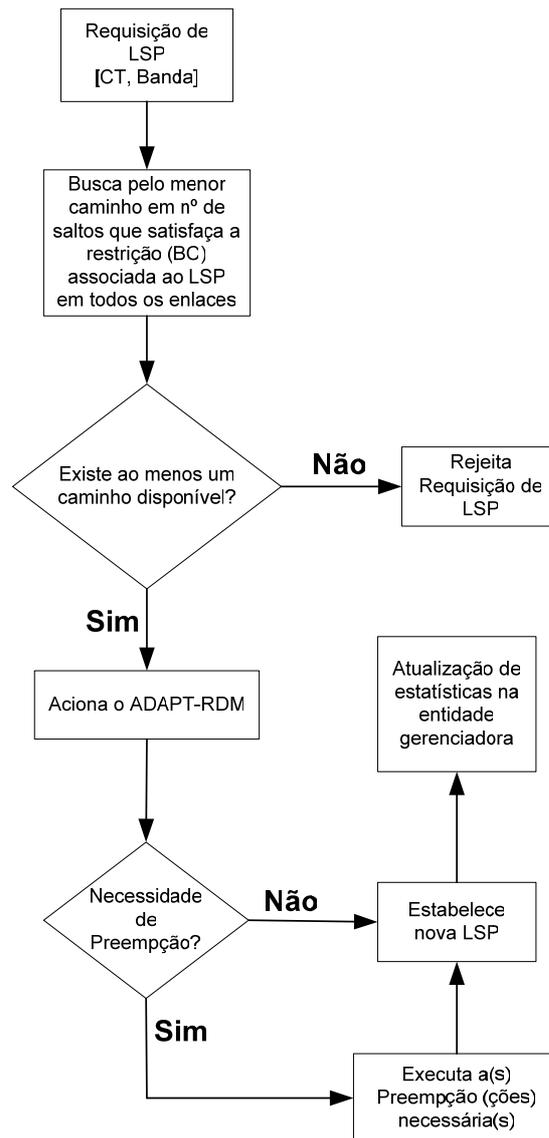


Figura 24 - Funcionamento da integração da seleção com base no menor caminho e o ADAPT-RDM (Elaborado pelo autor, 2008)

A Figura 25 mostra um pseudo-código descrevendo a implementação proposta do algoritmo. O algoritmo recebe como parâmetros o tipo de classe associado (**CT**), a largura de banda (**New\_LSP\_Band**), e os caminhos disponíveis (**Max\_Caminhos\_Disponíveis**) para o estabelecimento de um novo LSP. Dentro da perspectiva do controle centralizado, é assumido

que a entidade centralizadora possua o conhecimento de todos possíveis caminhos existentes entre dois nós na rede.

Na linha 1 são inicializados o flag (**establish**) que indicará se é possível ou não o estabelecimento do novo LSP, a variável (**Melhor\_Caminho**), que é o indexador para o menor caminho que satisfaça as condições do LSP dentro dos possíveis caminhos existentes, e uma variável auxiliar (**auxbreak**) utilizada para a computação do menor caminho por parte do algoritmo.

Entre as linhas 2-20 acontece a tentativa de encontrar o menor caminho (em termos de número de saltos) que tenha banda suficiente para acomodar a nova requisição. O caminho com menor número de saltos e que simultaneamente tenha a condição da linha 6 ( $\text{New\_LSP\_Band} + \text{A\_BC[CT]} > \text{MAX\_BC[CT]}$ ) respeitada em todos os seus enlaces será o escolhido. Essa condição garante que realmente há espaço para a nova requisição de LSP dentro da restrição de banda BC na qual esteja associada.

Nas linhas 21-22, em caso de nenhum dos caminhos da rede satisfazer a condição da linha 6, a requisição é bloqueada (LSP não pode ser estabelecido por falta de recursos). Uma variável global (**Total\_LSPs\_Blocked**) é utilizada para contabilizar o número total de LSPs bloqueados.

As linhas 23-29 do algoritmo são executadas no caso de existir pelo menos um caminho no qual a nova requisição de LSP possa ser estabelecida (i.e., caminho cujas BCs atendam os requisitos de banda da nova requisição de LSP). O algoritmo ADAPT-RDM deve ser acionado (linha 25), com o intuito de verificar se será ou não necessário preemptar LSPs de CTs hierarquicamente inferiores nos enlaces do caminho escolhido, visando disponibilizar banda para atender à nova requisição. Dessa forma é possível assegurar simultaneamente proteção do tráfego especificado para cada BC, e uma melhor eficiência da utilização de banda baseada nas propriedades de compartilhamento do modelo RDM.

Após essa verificação, a nova requisição de LSP é estabelecida e as estatísticas das cargas atuais dos enlaces e dos BCs (**A\_BC**) são atualizadas (linhas 26- 28). Ao final, um flag (**establish**) é retornado informado se o estabelecimento foi possível ou não (linha 30).

```

Alg. Implementação_CSPF (CT, New_LSP_Band, Max_Caminhos_Disponíveis)
1.  establish=0, Melhor_Caminho=0, auxbreak=0;
2.  For (i=0; i< Max_Caminhos_Disponíveis;i++)
3.  {
4.    For (nEnl=0; nEnl< Max_Enlaces_do_Caminho; nEnl++)
5.    {
6.      if(New_LSP_Band + A_BC[CT]> MAX_BC[CT])
7.      {
8.        auxbreak=1;
9.        break;
10.     }
11.    else
12.      auxbreak=0;
13.    }
14.    if(auxbreak==0)
15.    {
16.      Melhor_Caminho=i;
17.      establish=1;
18.      break;
19.    }
20.  }
21.  if(establish ==0)
22.    Total_LSPs_Blocked++;
23.  else
24.  {
25.    prept=ADAPT-RDM (CT, New_LSP_Band, Indice_Melhor_Caminho);
26.    Establish_New_LSP_Request;
27.    Update_A_BC (nEnl)
28.    Update_link_values;
29.  }
30.  return(establish)

```

Figura 25 - Pseudo-código para uma implementação do algoritmo CSPF integrada com o ADAPT-RDM (Elaborado pelo autor, 2008)

#### 5.4 ALGORITMO PARA SELEÇÃO DE CAMINHOS QUE MINIMIZA A OCORRÊNCIA DE PREEMPÇÃO CIENTE DO MODELO RDM

Esta seção apresenta uma proposta de algoritmo para o cálculo do caminho de LSPs baseado em restrições, denominado **MinPreptDSTE** (PINTO NETO *et al.*, 2008), que busca minimizar a ocorrência de preempções em uma rede DS-TE utilizando o modelo RDM. O algoritmo baseia-se no controle centralizado e utiliza a proposta do **ADAPT-RDM** para torná-lo aplicável a uma rede DS-TE habilitada com o modelo RDM.

O objetivo do **MinPreptDSTE** (PINTO NETO *et al.*, 2008) é minimizar o número de preempções quando comparado com o cálculo do caminho de LSPs com base na minimização da quantidade de saltos. A redução do número de preempções em uma rede multi-serviço é de interesse, pois evita-se assim a interrupção de serviço e reduz-se o tráfego de sinalização na rede (à medida que reduzindo o número de LSPs preemptados reduz-se também a necessidade do re-roteamento posterior dos mesmos).

A busca da minimização do número de preempções ocorridas na rede é feita pelo algoritmo **MinPreptDSTE** levando-se em conta a quantidade de banda compartilhada entre os CTs em cada enlace no cálculo do caminho. De forma geral, terá preferência na escolha o caminho que possuir enlaces com a menor banda compartilhada entre o CT correspondente à nova requisição de LSP e os CTs hierarquicamente inferiores de LSPs já estabelecidos. Diferentemente do cálculo de caminhos baseado na minimização da quantidade de saltos apenas, a proposta do **MinPreptDSTE** faz que o cálculo do caminho esteja ciente do nível de compartilhamento de banda entre CTs nos enlaces da rede.

Um pseudo-código do algoritmo MinPreptDSTE é mostrado na Figura 26. O algoritmo recebe como parâmetros o tipo de classe associado (**CT**), a largura de banda (**New\_LSP\_Band**), e os caminhos disponíveis (**Max\_Caminhos\_Disponíveis**) para o estabelecimento da nova requisição de LSP. Dentro da perspectiva de controle centralizado é assumido que a entidade centralizadora possua o conhecimento de todos possíveis caminhos existentes entre dois nós na rede.

Na linha 1 são inicializados o flag (**establish**) que indicará se é possível ou não o estabelecimento do novo LSP, e um conjunto de variáveis auxiliares para o cálculo do melhor caminho escolhido para o mesmo, em termos de minimização da quantidade de preempções. Entre as linhas 2-17, para todos os caminhos disponíveis, é calculada a banda total compartilhada (**TotalSharedBand**) (linhas 6-9) entre o CT da nova requisição e os CTs hierarquicamente inferiores, ou seja, o quão de banda do CT associado à nova requisição de LSP está sendo utilizada por CTs hierarquicamente inferiores.

Também é calculada a quantidade de banda disponível dentro das restrições de banda existentes (**MenorValorBC**) para cada enlace (linhas 10-11), i.e., a largura de banda ainda disponível para a utilização em cada um dos enlaces do caminho para uma determinada BC. Em cada caminho disponível para o estabelecimento de um LSP entre um par origem-destino, o enlace com menor (**MenorValorBC**) será aquele que possui a menor quantidade disponível subtraindo a máxima restrição de banda (**MAX\_BC[CT]**) da banda atual utilizada por determinado BC (**A\_BC[CT]**). Dessa forma é possível saber se a menor restrição de banda entre todos os enlaces de um caminho existente é suficiente para acomodar a largura de banda da nova requisição de LSP.

```

Alg. MinPreptDSTE (CT, New_LSP_Band, Max_Caminhos_Disponíveis)
1.  establish=0, MenorBandaCompart=0, MenorValorBC=0, BandaCompart=0, Vetor_
    MenorValorBC [Max_Caminhos_Disponíveis ];
2.  For (i=0; i< Max_Caminhos_Disponíveis;i++)
3.  {
4.    For (nEnl=0; nEnl< Max_Enlaces_do_Caminho; nEnl++)
5.    {
6.      For(k=0;k<LSP->item.CT;k++)
7.        TotalSharedBand = (A_BC[k]- A_BC [k+1]) + TotalSharedBand;
8.      if(BandaCompart <= TotalSharedBand)
9.        BandaCompart = TotalSharedBand;
10.     if(MenorValorBC <= MAX_BC[CT] - A_BC[CT])
11.       MenorValorBC = (MAX_BC[CT] - A_BC[CT]);
12.     TotalSharedBand=0;
13.   }
14.   Vetor_ BandaCompart [i] = BandaCompart;
15.   Vetor_ MenorValorBC [i] = MenorValorBC;
16.   BandaCompart = MenorValorBC= 0;
17. }
18. For (i=0; i< Max_Caminhos_Disponíveis;i++)
19. {
20.   if(New_LSP_Band <= Vetor_ MenorValorBC [i])
21.   {
22.     establish=1;
23.     if(MenorBandaCompart <= Vetor_ BandaCompart [i])
24.     {
25.       MenorBandaCompart = Vetor_ BandaCompart [i];
26.       Indice_Melhor_Caminho = i;
27.     }
28.   }
29. }
30. if(establish ==0)
31.   Total_LSPs_Blocked++;
32. else
33. {
34.   prept=ADAPT-RDM (CT, New_LSP_Band, Indice_Melhor_Caminho);
35.   Establish_New_LSP_Request;
36.   Update_ A_BC (nEnl)
37.   Update_link_values;
38. }
39. return(establish)

```

Figura 26 - Pseudo-código da proposta de algoritmo MinPreptDSTE. (Elaborado pelo autor, 2008)

Portanto, duas condições são exigidas para a escolha de fato ou não de um caminho para uma nova requisição (linhas 18-28):

- a) ter largura de banda disponível dentro das restrições do modelo RDM (BCs);
- b) ter a menor banda compartilhada com o CT da nova requisição de LSP para todos os enlaces do caminho.

Se a primeira condição for atingida, o caminho que tiver o enlace com o menor valor de banda compartilhada entre o CT da nova requisição e os CTs hierarquicamente inferiores será o

escolhido, visando à minimização do número de preempções necessárias na rede para o estabelecimento da nova requisição. Se nenhum dos caminhos possíveis obedecerem à primeira condição a nova requisição de LSP é bloqueada (linhas 30 e 31). Uma variável global (**Total\_LSPs\_Blocked**) é utilizada para contabilizar o número total de LSPs bloqueados.

As linhas 32-38 do algoritmo são executadas no caso de existir pelo menos um caminho no qual a nova requisição de LSP possa ser estabelecida (i.e., caminho cujas BCs atendam os requisitos de banda da nova requisição de LSP). O algoritmo ADAPT-RDM deve ser acionado (linha 34), com o intuito de verificar se será ou não necessário preemptar LSPs de CTs hierarquicamente inferiores nos enlaces do caminho escolhido, visando disponibilizar banda para atender à nova requisição segundo o modelo RDM. Dessa forma é possível assegurar simultaneamente proteção do tráfego especificado para cada BC, e uma melhor eficiência da utilização de banda baseada nas propriedades de compartilhamento do modelo RDM.

Após essa verificação, a nova requisição de LSP é estabelecida e as estatísticas das cargas atuais dos enlaces e dos BCs (**A\_BC**) são atualizadas (linhas 35- 37). Ao final, um *flag* é retornado informado se o estabelecimento foi possível ou não (linha 39).

Nos capítulos seguintes será apresentada uma validação dos algoritmos propostos nesse capítulo através de simulações.

## 6 VALIDAÇÃO DOS ALGORITMOS

A alternativa utilizada para a análise de desempenho dos algoritmos propostos foi a simulação. Este capítulo descreve uma implementação de um simulador de propósito específico (biblioteca de funções em linguagem C) que foi desenvolvido com a finalidade de avaliar o desempenho dos algoritmos propostos neste trabalho. O simulador de propósito específico tem o intuito de representar computacionalmente o funcionamento de uma rede DS-TE habilitada com o modelo RDM. As principais funções (código) do simulador são apresentadas no ANEXO A desse trabalho.

A opção pelo desenvolvimento de um simulador específico foi realizada por dois motivos principais:

- a) dificuldade de encontrar na literatura um simulador que já tivesse implementado as características desejadas de uma rede DS-TE habilitada com o modelo de restrição de banda RDM.
- b) considerou-se que a partir de um desenvolvimento próprio seria mais fácil a intervenção no código visando implementar as particularidades dos problemas propostos nessa dissertação, bem como a avaliação de desempenho das soluções propostas.

Vale ressaltar que o desenvolvimento de um simulador de redes genérico não é propósito desta dissertação. Foi efetivamente desenvolvido um conjunto de funções em linguagem C com a finalidade explícita de estudar e avaliar os problemas apresentados e os algoritmos propostos nesta dissertação. Também não é proposta dessa dissertação nenhuma facilidade gráfica para a entrada de dados e para a apresentação dos resultados da simulação.

### 6.1 O SIMULADOR

O simulador de propósito específico desenvolvido está baseado na execução de um conjunto de eventos discretos no tempo (MACDOUGALL, 1987). A idéia é que a partir da “execução cíclica” de um conjunto de eventos ordenados em uma lista (denominada cadeia de eventos) em ordem cronológica, seja possível representar computacionalmente e realizar a modelagem desejada de uma rede DS-TE utilizando o modelo RDM.

A inserção e retirada de eventos da cadeia de execução de eventos caracteriza o processo de funcionamento da máquina de eventos do simulador. A implementação do simulador de propósito específico desenvolvido é baseada na filosofia de funcionamento de execução da cadeia de eventos da linguagem de simulação SMPL, especificada em (MACDOUGALL, 1987). O SMPL é uma extensão funcional (biblioteca de funções) da linguagem C de propósito geral, adequado para modelos de simulação pequenos ou médios. Uma ilustração do funcionamento da execução da máquina de eventos é mostrada na Figura 27 a seguir.

A partir do processamento cíclico de um conjunto de eventos discretos associados aos LSPs, é possível simular o sistema desejado. Um evento a ser processado é retirado do topo da cadeia de eventos, e dependendo do tipo ao qual pertença, passará por determinado processamento. Após o processamento de um evento, um outro tipo de evento é gerado se existir o encadeamento de eventos. No caso do exemplo da Figura 27, após o processamento dos eventos tipo 1 e 2, é gerado um outro tipo de evento. Após o processamento do evento tipo 3 não há encadeamento de eventos. Após o processamento de cada evento as estatísticas de simulação são atualizadas.

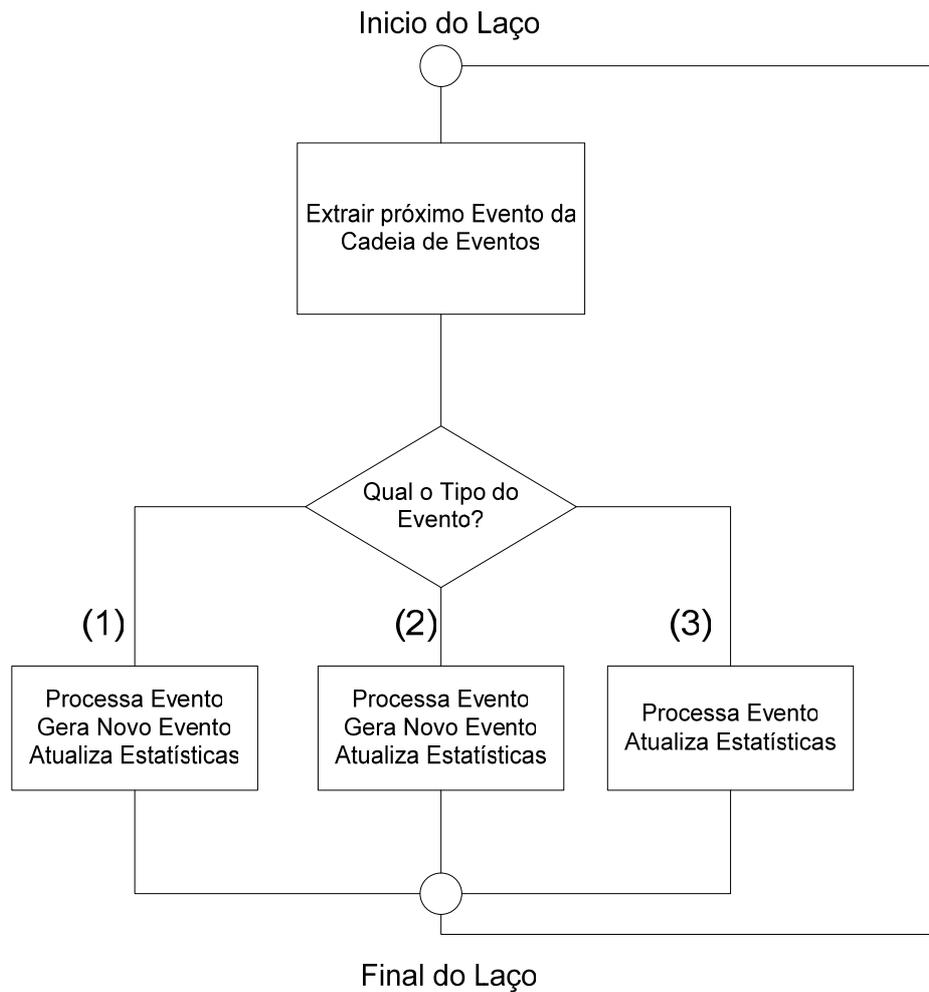


Figura 27 - Máquina de Eventos do Simulador (Elaborado pelo autor, 2008)

O simulador de propósito específico desenvolvido utiliza-se do gerador de números aleatórios da “*srand()*” implementado como biblioteca da linguagem C. As geração de variáveis aleatórias segundo as distribuições Uniforme e Exponencial são implementadas utilizando-se dos fragmentos de códigos da linguagem de simulação SMPL definida em (MACDOUGALL, 1987), cujos trechos dos códigos são mostrados no ANEXO II desse trabalho.

## 6.2 SIMULAÇÃO - DESCRIÇÃO

Esta seção descreve as etapas para preparar uma simulação utilizando o simulador de propósito específico utilizado para simular o comportamento de uma rede DS-TE utilizando o modelo RDM.

O preparo da simulação passa pela configuração da topologia da rede e das particularidades de uma rede DS-TE utilizando o modelo RDM e está associado a um conjunto de etapas, descritas a seguir:

**a) primeira etapa: Definição da Topologia e das Configurações DS-TE/RDM**

- Construção da topologia de rede que deseja-se simular criando os nós (roteadores) e os enlaces que conectam os nós do modelo;
- Associação dos geradores de LSPs aos respectivos nós aos quais eles estão associados no modelo;
- Definição do número de CTs configurados em rede;
- Definição das quantidades máximas dos BCs configurados em rede.

**b) segunda etapa: Inicialização de LSPs por CT**

- Para a utilização da biblioteca de funções implementada, é necessária a inicialização dos LSPs por CT para cada nó origem de tráfego na rede. A inicialização feita no programa principal consiste da geração manual de LSPs no tempo de simulação 0 (zero) antes do início da execução da simulação (pré-requisito para o início do funcionamento da máquina de eventos). Em cada nó origem de tráfego na rede é inicializado um LSP para cada CT configurado.

**c) terceira etapa: Execução do Laço de Simulação**

- Retirada do evento atual a ser processado do final da cadeia de eventos. Três eventos são possíveis: (evento 1) Geração de um LSP; (evento 2) Tentativa de Estabelecimento de LSP; (evento 3) Desativação de LSP. Esses eventos são detalhados na seção seguinte.
- Processamento do evento retirado do topo da cadeia de eventos;
- Geração de um novo evento para a cadeia de eventos:
  - Após a execução do evento 1 (Geração de LSP) dois novos eventos são gerados. É gerado um evento 2 (Tentativa de Estabelecimento de LSP) para a tentativa de estabelecimento do LSP gerado, e é gerado um novo evento 1 que define quando um próximo LSP nas mesmas condições do LSP ao qual foi posto em tentativa de estabelecimento deverá ser gerado.

- Após a execução do evento 2 (Tentativa de Estabelecimento de LSP) em caso de sucesso no estabelecimento do LSP, é gerado um evento 3 (Desestabelecimento de LSP) com o tempo no qual o LSP deverá permanecer estabelecido (i.e., tempo de vida do LSP). Em caso de insucesso no estabelecimento do LSP nenhum outro evento é gerado, sendo contabilizado o bloqueio do LSP por falta de recursos. As estatísticas dos enlaces, dos CTs e das BCs são atualizados a cada estabelecimento de LSP.
- Após a execução do evento 3 (Desestabelecimento de LSP) o LSP é desestabelecido e as estatísticas dos enlaces, dos CTs e das BCs são atualizados. Nenhum outro evento é gerado, após a execução de um evento 3.

**c) Quarta Etapa: Coleta das Estatísticas de Simulação**

- Ao final da execução do laço (final da simulação), o programa coleta os dados de performance de interesse (Utilização dos CTs, utilização das BCs, quantidade de preempções de LSPs ocorridas em rede) obtidos para análise pós-simulação.

### 6.3 EVENTOS E MÓDULOS DA SIMULAÇÃO

Os eventos que são passíveis de ocorrência e os módulos do simulador juntamente com a filosofia de funcionamento implementada são detalhados nesta seção. Um fluxograma para facilitar o entendimento é mostrado na Figura 28:

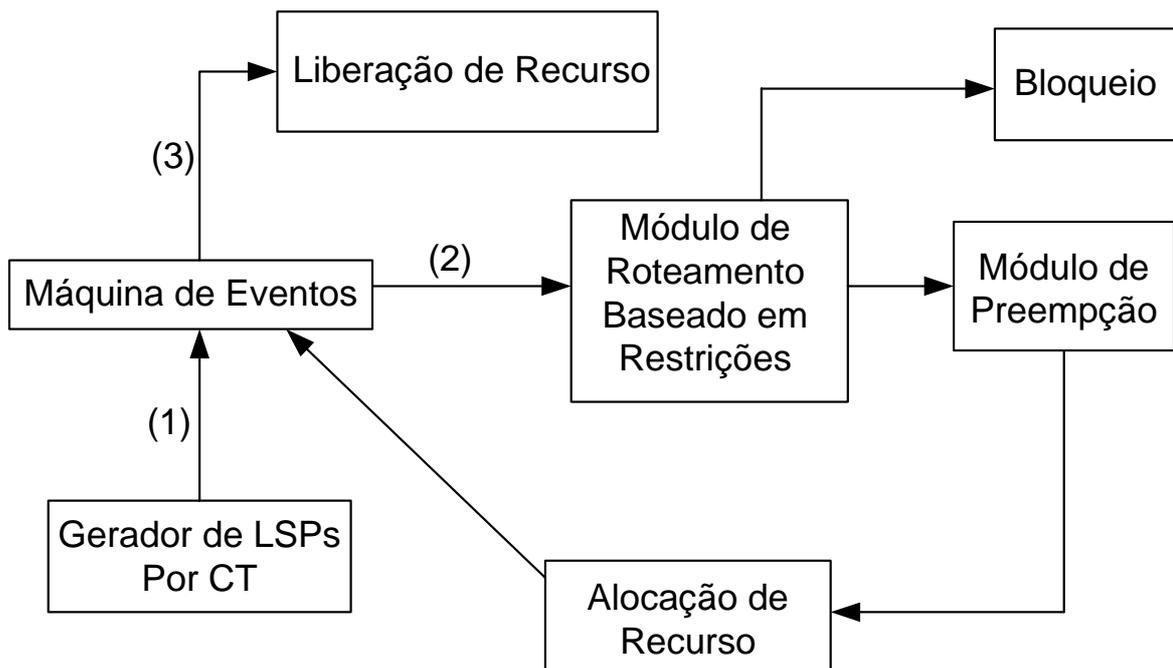


Figura 28 - Eventos, módulos e principais etapas da simulação (Elaborado pelo autor, 2008)

### 6.3.1 Evento 1: Geração de um novo LSP para a rede

Neste evento é realizada a geração de LSPs associado a um determinado nó (roteador) da rede. Os LSPs são gerados por CT com diferentes tempos de interchegada baseados na distribuição exponencial. O tempo médio de interchegada na geração de LSPs pode variar a depender da configuração de cada um dos CTs configurados em rede.

As seguintes definições são necessárias para a execução do Evento 1 (geração de LSPs), considerando a forma como a implementação foi realizada e as particularidades de interesse de uma rede DS-TE utilizando o modelo RDM:

- a) definição do nó de origem do LSP na rede: Os LSPs terão como possíveis origens os nós geradores de LSPs que forem definidos na simulação;
- b) definição do nó de destino do LSP na rede: Os LSPs terão como possíveis destinos os nós sorvedouros de LSPs que forem definidos na simulação;
- c) associação à matriz de caminhos: O LSP é associado a uma tabela que contém os caminhos disponíveis entre o nó de origem e o nó de destino através de um indexador (A Matriz de Caminhos está armazenada na entidade gerenciadora);

- d) largura de banda do LSP: Definida com base na distribuição uniforme, onde existe um valor de banda mínimo e um valor de banda máximo especificado para cada CT configurado em rede;
- e) prioridade do LSP: Define a prioridade do LSP para o caso da necessidade futura de preempção.

### 6.3.2 Evento 2: Tentativa de estabelecimento de LSP

Nesse evento é tentado o estabelecimento de uma nova requisição de LSP que foi gerado. A partir desse evento são executados três módulos principais: O módulo de seleção de caminhos, o módulo de verificação de necessidade de preempção de banda e o módulo de alocação de recursos.

O **módulo de seleção de caminhos** é o responsável pelo cálculo do “melhor caminho”, a depender das exigências do algoritmo de roteamento baseado em restrições que estiver sendo utilizado (CSPF ou MinPreptDSTE), dentro do conjunto de caminhos possíveis desde o nó de origem até o nó de destino da nova requisição de LSP. Se não existir um caminho que atenda os requisitos na nova requisição de LSP associada a determinado CT, a mesma será bloqueada. São parâmetros do módulo de seleção de caminhos:

- a) nó origem e nó de destino da nova requisição;
- b) apontador para a Matriz de Caminhos possíveis entre um nó de origem e um nó de destino;
- c) banda exigida e CT correspondente à nova requisição de LSP;
- d) banda máxima permitida aos BCs configurados em rede (configurações do modelo RDM);
- e) capacidade não utilizada dos enlaces da rede por BC. Em outras palavras, a banda atual ocupada por cada BC configurado em rede.

O **módulo de verificação da necessidade de preempção de banda** é o responsável por checar se existe a necessidade de adaptação de banda em CTs hierarquicamente inferiores ao CT da nova requisição de LSP. A verificação é feita em cada enlace do caminho escolhido, visando manter as configurações do modelo RDM. A execução do algoritmo **ADAPT-RDM** proposto é a responsável por essa verificação da necessidade de adaptação de banda, sendo que a adaptação é feita através da preempção de LSPs estabelecidos em rede, associados aos

CTs cujos BCs estariam com suas máximas configurações de banda sendo ultrapassadas. São parâmetros do módulo de verificação da necessidade de preempção:

- a) caminho escolhido pelo módulo de seleção de caminho;
- b) banda exigida e CT correspondente à nova requisição de LSP;
- c) banda máxima permitida aos BCs configurados em rede (configurações do modelo RDM);
- d) banda atual ocupada por cada BC configurado em rede;
- e) fator  $w(\text{LSP})$  da nova requisição de LSPs visando compará-lo com o Fator  $w(\text{LSP})$  de LSPs já estabelecidas em caso de necessidade de preempção de LSPs em determinado enlace.

O **módulo de alocação de recursos** é o responsável por simular o estabelecimento de fato da nova requisição de LSP. Atualiza as estatísticas da entidade gerenciadora de banda pós-estabelecimento. A atualização das variáveis de controle da entidade gerenciadora de banda representam no simulador o estabelecimento de fato do LSP, e a posse do recursos dos enlaces pelo mesmo. Algumas atualizações na entidade gerenciadora são feitas para representar o estabelecimento no simulador de um LSP:

- a) armazenamento do novo LSP estabelecido na base de dados de LSPs estabelecidos agrupados por CT, em todos os enlaces associados ao novo caminho;
- b) atualização da banda dos enlaces do caminho escolhido para a nova requisição de LSP;
- c) atualização da banda atual ocupada por cada BC configurada em rede nos enlaces do caminho da nova requisição de LSP;
- d) tempo em que o LSP permanecerá estabelecido (tempo de vida do LSP).

### **6.3.3 Evento 3: Desativação de um LSP**

Nesse evento é realizada a desativação de um LSP. O Evento 3 pode ser executado em duas situações distintas na implementação do simulador.

A primeira é a desativação de um LSP devido ao estouro de seu “tempo de vida”, ou, em outras palavras, quando o tempo de duração de um LSP se esgota. O tempo de duração do LSP é definido no momento em que ele é estabelecido (Evento 2) na simulação. As seguintes

atualizações na entidade gerenciadora após o desestabelecimento de um LSP devido ao estouro de seu “tempo de vida” devem ser realizadas:

- a) retirada do LSP da base de dados de LSPs estabelecidos agrupados por CT, em todos os enlaces aos quais os mesmos estiverem associados;
- b) atualização da variável A\_BC[n] (banda atual ocupada por determinada restrição de banda) nos enlaces do caminho do LSP que teve seu tempo de vida (duração) estourado.

A outra possibilidade de ocorrência do Evento 3, é quando há necessidade de preempção de LSPs estabelecidos. As seguintes atualizações na entidade gerenciadora após o desestabelecimento de um LSP devido à necessidade de preempção de LSPs:

- a) retirada dos LSPs preemptados da base de dados de LSPs estabelecidos agrupados por CT, em todos os enlaces aos quais os mesmos estiverem associados;
- b) atualização da variável A\_BC[n] (banda atual ocupada por determinada restrição de banda) nos enlaces do caminho que tiveram LSP(s) preemptado(s).

## **7 RESULTADOS DA AVALIAÇÃO DOS ALGORITMOS (ADAPT-RDM, CSPF E MINPREPTDSTE)**

Este capítulo apresenta os resultados das simulações realizadas com o simulador de propósito específico para avaliar os algoritmos ADAPT-RDM, CSPF e MinpreptDSTE. A avaliação de desempenhos dos algoritmos foi realizada em duas etapas:

- a) primeiramente o algoritmo ADAPT-RDM para a gerência de banda compartilhada entre CTs em uma rede DS-TE utilizando o modelo RDM foi avaliado individualmente. A intenção foi verificar se a implementação do algoritmo estava de acordo com os objetivos propostos. Os objetivos propostos para o ADAPT-RDM são o respeito às máximas restrições de banda de cada um dos BCs configuradas em rede e, simultaneamente, permitir o compartilhamento de banda entre CTs.
- b) em uma segunda etapa, as propostas de algoritmos de seleção de caminhos baseados em restrições, CSPF e MinPreptDSTE, que utilizam o algoritmo ADAPT-RDM para se adequarem a uma rede DS-TE com o modelo RDM foram avaliadas. A intenção foi observar se o algoritmo MinPreptDSTE é capaz de reduzir a quantidade de preempções ocorridas em rede quando comparado com o algoritmo CSPF que seleciona caminhos para os LSPs baseado na minimização da quantidade de saltos entre a origem e o destino. O algoritmo CSPF pode encontrar caminhos baseados em diferentes métricas, sendo que neste trabalho utilizou-se o número de saltos (hops) como um elemento de avaliação dos resultados.

### **7.1 AVALIAÇÃO DO ALGORITMO ADAPT-RDM**

Como cenário para a avaliação da implementação do algoritmo ADAPT-RDM foi considerada a topologia de rede mostrada na Figura 29. A escolha dessa topologia deveu-se ao fato de que nesta avaliação não foi de interesse a possibilidade de escolha de mais de um caminho para uma requisição de LSP, e sim avaliar individualmente o desempenho do algoritmo ADAPT-RDM em uma rede DS-TE utilizando o modelo RDM. Diferentes cargas de tráfego na rede foram geradas no intuito de avaliar o comportamento do algoritmo diante de cenários de não-saturação e saturação dos enlaces da rede.

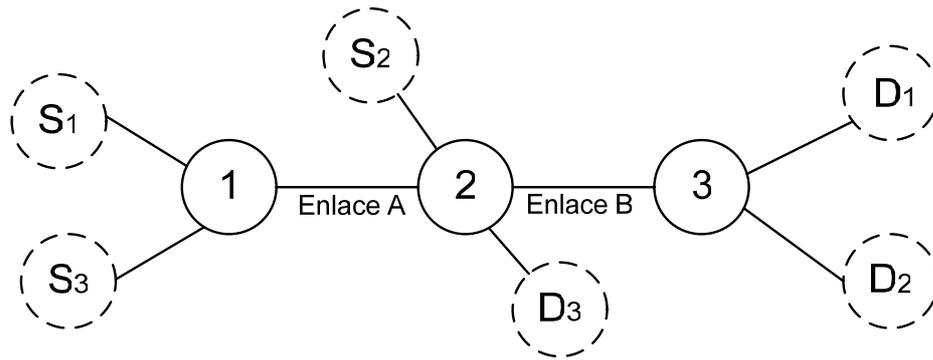


Figura 29 - Topologia Utilizada para Avaliação do Algoritmo ADAPT-RDM (Elaborado pelo autor, 2008)

A proposta de simulação visa criar situações com o gradual aumento de disputa por recursos nos enlaces da rede, de forma a validar a aplicabilidade do algoritmo ADAPT-RDM como um mecanismo de gerência de banda compartilhada, para garantir que as configurações do modelo RDM estejam sendo atendidas. S1, S2 e S3 são as possíveis origens de LSPs na rede, tendo D1, D2 e D3 como seus respectivos destinos.

Os enlaces A e B foram configurados com 622 Mbps cada (valor equivalente ao STM-4 da tecnologia SDH). Três CTs foram configurados na rede (CT0, CT1 e CT2). A máxima porcentagem de banda alocada para os BCs (MAX\_BC[n]) de acordo com o modelo RDM para os enlaces A e B, foram configurados de acordo com o quadro 18.

BC	MAX_BC (%)	MAX_BC (Equivalente em Mbps)
0	100	622
1	70	435,4
2	40	248,8

Quadro 18 - Configurações dos BCs para o Cenário de Simulação (Elaborado pelo autor, 2008)

As três fontes de tráfego (S1, S2 e S3) geram aleatoriamente requisições de estabelecimentos de LSPs por CT. A quantidade de banda de cada LSP gerado varia uniformemente entre 5 Mbps e 20 Mbps.

A carga de chegada de LSPs nas simulações foi gerada da seguinte maneira:

- a) um número entre 1 e 100 foi gerado segundo a distribuição uniforme. Com 70% de probabilidade, um LSP associado ao CT0 é gerado em cada fonte, e com 50% de probabilidade um LSP associado ao CT1 é gerado em cada fonte. A carga dos LSPs associados com o CT2 (mais alta hierarquia entre os três CTs) foi aumentada

gradativamente de 20% até 70% de probabilidade nas simulações. O intuito com esse aumento gradativo foi verificar a evolução dos BCs e a necessidade ou não de ocorrência de preempção de LSPs em situações de não-disputa e disputa por recursos (largura de banda) entre diferentes CTs;

- b) foi configurado que existe tempo de vida infinito para os LSPs estabelecidos, i.e., os LSPs uma vez estabelecidos permanecem nesse estado até o final deste cenário de simulação. Essa opção foi feita pois assumiu-se que dessa forma haveria um maior controle da saturação gradual dos enlaces, possibilitando uma análise mais confiável da funcionalidade do algoritmo ADAPT-RDM.

Nas simulações realizadas, as métricas de desempenho avaliadas foram a utilização da largura de banda por CT e a utilização da largura de banda por BC. O critério de parada de simulação utilizado foi o tempo de simulação.

Para cada conjunto de números definidos nos parâmetros, foram realizadas cinco simulações, variando-se a semente do gerador de números aleatórios utilizado (variadas através da função “*srand()*” que é uma biblioteca da linguagem C). Então, calculou-se a média de cada métrica de desempenho desejada, o desvio padrão e o intervalo de confiança de 90%, usando a distribuição de *Student* (SPIEGEL, 1993).

Com relação ao fator de preempção “**w(lsp)**”, duas configurações diferentes dos seus parâmetros foram consideradas:

- a) na primeira configuração, os pesos do fator de preempção “**w(lsp)**” em cada CT foram configurados como  $A = 1$  e  $B = 0$ , o que significa que nessas simulações foi considerado importante preemptar LSPs de baixa prioridade associados aos CTs, em caso da necessidade da preempções para adequação às máximas configurações das BCs.
- b) na segunda configuração, os pesos do fator de preempção “**w(lsp)**” em cada CT foram configurados como  $A = 1$  e  $B = 1$ , o que significa que nessa simulação foi considerado importante preemptar LSPs de baixa prioridade e baixa largura de banda associados aos CTs, em caso da necessidade da preempções para adequação às máximas configurações das BCs.

Os resultados para a primeira configuração de parâmetros do fator “**w(lsp)**” (com  $A=1$  e  $B=0$ ) são mostrados no quadro 19 a seguir.

		<b>Enlace A</b>																	
		<b>Média</b>						<b>Desvio Padrão</b>						<b>Intervalo de Confiança (90%)</b>					
<b>Variação (%)</b>	<b>Carga CT2</b>	<b>Utilização (%) BC</b>			<b>Utilização (%) CT</b>			<b>Utilização (%) BC</b>			<b>Utilização (%) CT</b>			<b>Utilização (%) BC</b>			<b>Utilização (%) CT</b>		
		BC[0]	BC[1]	BC[2]	CT[0]	CT[1]	CT[2]	BC[0]	BC[1]	BC[2]	CT[0]	CT[1]	CT[2]	BC[0]	BC[1]	BC[2]	CT[0]	CT[1]	CT[2]
	20	97.72	53.54	12.06	44.18	41.48	12.06	3.07	6.36	8.43	3.64	7.45	8.43	0.17	0.36	0.47	0.20	0.42	0.47
	30	98.88	60.32	14.92	38.55	45.40	14.92	0.92	4.59	5.25	4.32	1.34	5.25	0.05	0.26	0.29	0.24	0.08	0.29
	40	98.42	64.50	26.21	33.92	38.30	26.21	0.54	5.08	7.07	5.19	4.91	7.07	0.03	0.29	0.40	0.29	0.28	0.40
	50	98.49	66.79	37.04	31.70	29.74	37.04	1.09	3.92	1.68	4.54	5.22	1.68	0.06	0.22	0.09	0.26	0.29	0.09
	60	98.84	68.56	39.07	30.29	29.49	39.07	0.93	1.54	0.47	1.91	1.79	0.47	0.05	0.09	0.03	0.11	0.10	0.03
	70	98.81	68.71	39.20	30.10	29.52	39.20	0.83	0.91	0.38	1.16	0.91	0.38	0.05	0.05	0.02	0.06	0.05	0.02
		<b>Enlace B</b>																	
		<b>Média</b>						<b>Desvio Padrão</b>						<b>Intervalo de Confiança (90%)</b>					
<b>Variação (%)</b>	<b>Carga CT2</b>	<b>Utilização (%) BC</b>			<b>Utilização (%) CT</b>			<b>Utilização (%) BC</b>			<b>Utilização (%) CT</b>			<b>Utilização (%) BC</b>			<b>Utilização (%) CT</b>		
		BC[0]	BC[1]	BC[2]	CT[0]	CT[1]	CT[2]	BC[0]	BC[1]	BC[2]	CT[0]	CT[1]	CT[2]	BC[0]	BC[1]	BC[2]	CT[0]	CT[1]	CT[2]
	20	96.72	52.96	12.67	43.76	40.29	12.67	4.94	4.73	8.19	0.66	6.88	8.19	0.28	0.27	0.46	0.04	0.39	0.46
	30	98.97	60.29	16.11	38.68	44.18	16.11	0.44	5.48	5.80	5.75	2.10	5.80	0.02	0.31	0.33	0.32	0.12	0.33
	40	99.04	64.02	24.95	35.56	38.52	24.98	0.48	5.20	5.51	5.08	6.42	5.51	0.03	0.29	0.31	0.29	0.36	0.31
	50	99.10	66.65	37.91	32.44	28.75	37.91	0.59	5.30	2.19	5.71	6.11	2.19	0.03	0.30	0.12	0.32	0.34	0.12
	60	99.04	69.10	39.00	29.94	30.10	39.00	0.59	0.46	0.52	0.80	0.62	0.52	0.03	0.03	0.03	0.04	0.03	0.03
	70	99.45	68.23	39.52	31.22	28.71	39.52	0.59	1.06	0.24	1.34	1.24	0.24	0.03	0.06	0.01	0.08	0.07	0.01

Quadro 19 - Resultados de Simulação – Algoritmo ADAPT-RDM (Fatores “w(LSP)” A=1 e B=0) (Elaborado pelo autor, 2008)

Na Figura 30 são apresentados gráficos baseados nos resultados médios das métricas de desempenho de interesse nas simulações realizadas na primeira configuração de parâmetros do fator “ $w(lsp)$ ” (com  $A=1$  e  $B=0$ ), para a avaliação da implementação do algoritmo ADAPT-RDM.

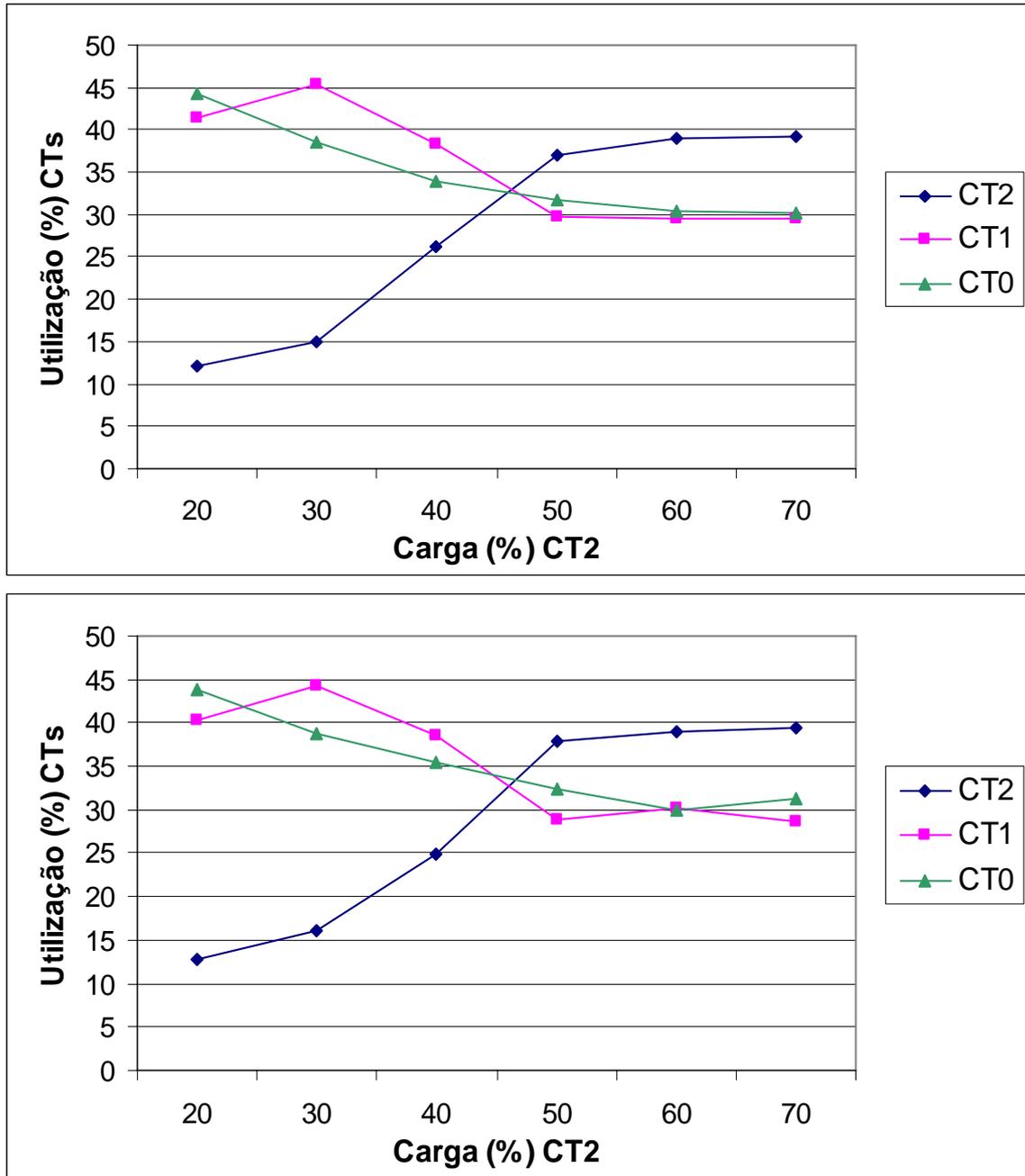


Figura 30 - Evolução da Largura de Banda Ocupada por CT para o Enlaces A e B Respectivamente (com  $A=1$  e  $B=0$ ) (Elaborado pelo autor, 2008)

A partir da análise dos resultados de simulação dos gráficos das Figura 30, chega-se às seguintes conclusões:

- a) quando existe largura de banda não-utilizada disponível no CT2 dos enlaces A e B, este recurso pode ser utilizado por CTs hierarquicamente inferiores (CT0 e CT1 no caso), em função da capacidade de compartilhamento de banda prevista no modelo RDM e assegurada pela implementação do algoritmo ADAPT-RDM;
- b) entretanto, em cenários de disputa por recursos, à medida que a carga associada ao CT2 aumenta, as cargas associadas aos CTs 0 e 1 começam a decair (a preempção de LSPs associado aos CTs 0 e 1 ocorre no algoritmo ADAPT-RDM) com a intenção de preservar o isolamento de tráfego configurada para o CT2 (BC2 que contém apenas o CT2 pode ocupar até 40% do valor dos enlaces);
- c) nota-se que em um primeiro momento há uma tendência dos LSPs associados ao CT0 serem preemptadas mais fortemente do que os LSPs associados ao CT1. Isso deve-se ao fato de que inicialmente ainda havia um maior espaço para o aumento de tráfego em BC1 do que para o aumento de tráfego em BC0.

A Figura 31 mostra a evolução da “utilização” dos enlaces pelos BCs com o aumento da carga associada ao CT2 na primeira configuração de parâmetros do fator “ $w(lsp)$ ” (com  $A=1$  e  $B=0$ ), para as mesmas configurações de carga:

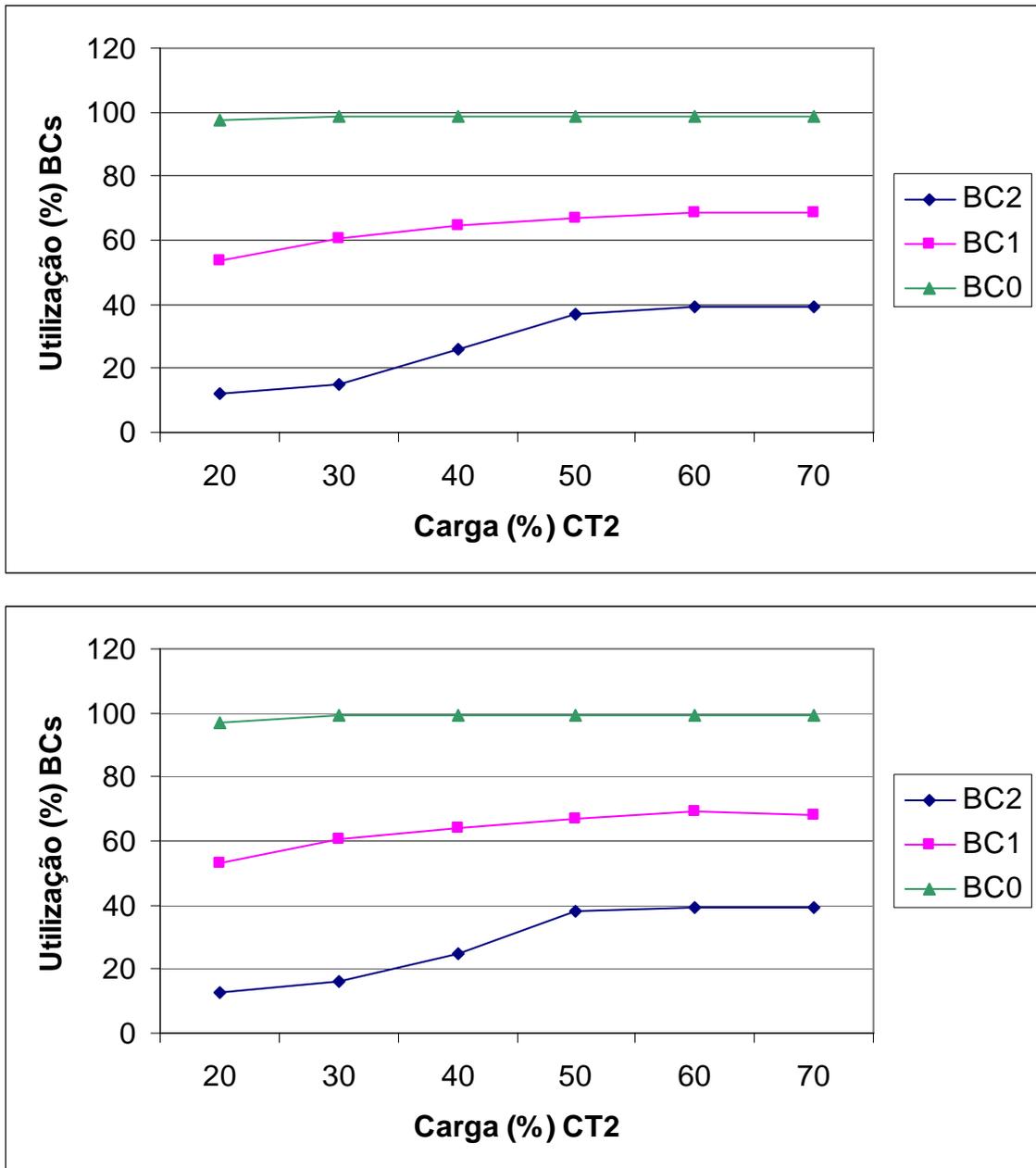


Figura 31 - Evolução da Largura de Banda Ocupada por BC para os Enlaces A e B Respectivamente (com  $A=1$  e  $B=0$ ) (Elaborado pelo autor, 2008)

A partir da análise dos resultados de simulação dos gráficos das Figura 31, chega-se às seguintes conclusões:

- a) quando a carga associada ao CT2 chega ao máximo valor BC2 ( $MAX\_BC[2] = 40\%$  da largura de banda do enlace), a preempção de banda para acomodar o tráfego de CT2 passa a ser não mais necessária. Isso se deve aos seguintes fatores:
  - O tráfego associado ao CT2 já passou a ocupar o máximo espaço reservado para ele (BC2) segundo as configurações do modelo RDM;

- A partir daí as novas requisições de LSPs associados ao CT2 que chegarem aos enlaces A e B passarão a ser bloqueadas.
- b) desde a carga inicial do CT2 (20%), o BC0 está utilizando aproximadamente o valor máximo especificado (100%), devido à saturação dos enlaces causada principalmente pela alta carga de LSPs associados ao CT0. De acordo com o modelo RDM, a carga do BC0 corresponde ao somatório das cargas dos LSPs associados aos CT2, CT1 e CT0.
- c) inicialmente BC1 e BC2 não estão utilizando seus valores máximos, mas quando a carga associada ao CT2 começa a aumentar, BC1 e BC2 aumentam somente até atingir seus limites máximos especificados, mostrando que as configurações do modelo RDM são preservadas.
- d) após todos os três BCs (BC0, BC1 e BC2) atingirem seus máximos valores de utilização (respectivamente 100%, 70% e 40%) não mais torna-se necessário a preempção de LSPs, as novas requisições de LSP não podem mais serem estabelecidas e começam a ser bloqueadas. Isso porque uma condição deste cenário de simulação é que os LSPs uma vez estabelecidos são mantidos nesse estado até o final da simulação, ou seja, não há tempo de vida limite para o LSP.

Os resultados para a segunda configuração de parâmetros (fator “**w(lsp)**” com A=1 e B=1) são mostrados na quadro 20 a seguir.

<b>Enlace A</b>																			
<b>Varição (%) Carga CT2</b>		<b>Média</b>						<b>Desvio Padrão</b>						<b>Intervalo de Confiança (90%)</b>					
		<b>Utilização (%) BC</b>			<b>Utilização (%) CT</b>			<b>Utilização (%) BC</b>			<b>Utilização (%) CT</b>			<b>Utilização (%) BC</b>			<b>Utilização (%) CT</b>		
		BC[0]	BC[1]	BC[2]	CT[0]	CT[1]	CT[2]	BC[0]	BC[1]	BC[2]	CT[0]	CT[1]	CT[2]	BC[0]	BC[1]	BC[2]	CT[0]	CT[1]	CT[2]
20	96.24	46.53	13.79	49.71	32.73	13.79	3.45	4.44	4.63	2.23	4.22	4.63	0.18	0.23	0.24	0.11	0.22	0.24	
30	98.36	54.40	20.87	43.96	33.54	20.87	2.45	6.94	5.75	5.45	7.21	5.75	0.13	0.36	0.30	0.28	0.37	0.30	
40	99.33	65.60	29.55	33.73	36.05	29.55	0.39	4.12	6.43	4.35	7.78	6.43	0.02	0.21	0.33	0.22	0.40	0.33	
50	98.81	67.85	33.50	30.97	34.34	33.50	0.91	2.45	7.66	3.15	9.14	7.66	0.05	0.13	0.39	0.16	0.47	0.39	
60	99.52	67.53	38.33	31.99	29.20	38.33	0.59	2.58	2.31	2.90	4.36	2.31	0.03	0.13	0.12	0.15	0.22	0.12	
70	99.00	68.55	38.43	30.45	30.13	38.43	0.86	0.82	0.93	1.41	1.58	0.93	0.04	0.04	0.05	0.07	0.08	0.05	
<b>Enlace B</b>																			
<b>Varição (%) Carga CT2</b>		<b>Média</b>						<b>Desvio Padrão</b>						<b>Intervalo de Confiança (90%)</b>					
		<b>Utilização (%) BC</b>			<b>Utilização (%) CT</b>			<b>Utilização (%) BC</b>			<b>Utilização (%) CT</b>			<b>Utilização (%) BC</b>			<b>Utilização (%) CT</b>		
		BC[0]	BC[1]	BC[2]	CT[0]	CT[1]	CT[2]	BC[0]	BC[1]	BC[2]	CT[0]	CT[1]	CT[2]	BC[0]	BC[1]	BC[2]	CT[0]	CT[1]	CT[2]
20	97.24	48.94	14.28	48.30	34.66	14.28	5.21	4.40	5.31	4.84	3.48	5.31	0.27	0.23	0.27	0.25	0.18	0.27	
30	99.39	55.14	20.84	44.25	34.31	20.84	0.40	7.52	5.78	7.84	7.29	5.78	0.02	0.39	0.30	0.40	0.37	0.30	
40	99.04	64.69	28.23	34.34	36.46	28.23	0.89	5.93	6.33	5.55	8.58	6.33	0.05	0.30	0.32	0.28	0.44	0.32	
50	99.33	67.88	34.21	31.45	33.67	34.21	0.38	3.43	8.68	3.20	10.28	8.68	0.02	0.18	0.45	0.16	0.53	0.45	
60	99.26	67.24	38.26	32.03	28.97	38.26	0.43	3.10	2.35	2.95	4.03	2.35	0.02	0.16	0.12	0.15	0.21	0.12	
70	99.78	69.23	39.33	30.55	29.90	39.33	0.33	0.40	0.18	0.51	0.51	0.18	0.02	0.02	0.01	0.03	0.03	0.01	

Quadro 20 - Resultados de Simulação – Algoritmo ADAPT-RDM (Fatores “w(LSP)” A=1 e B=1) (Elaborado pelo autor, 2008)

Na Figura 32 são apresentados gráficos baseados nos resultados médios das métricas de desempenho de interesse nas simulações realizadas na segunda configuração do fator “ $w(lsp)$ ” (com  $A=1$  e  $B=1$ ), para a avaliação da implementação do algoritmo ADAPT-RDM.

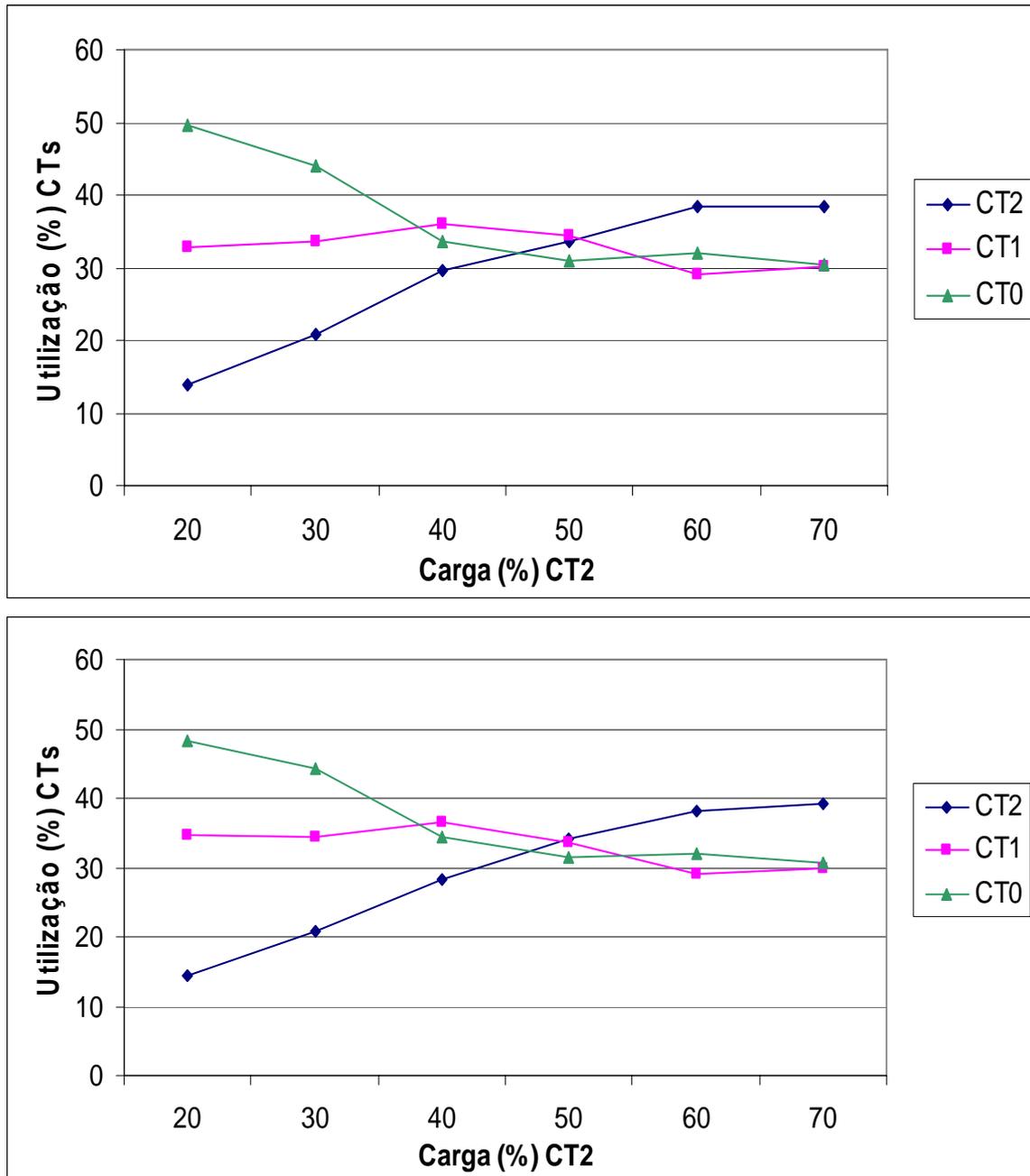


Figura 32 - Evolução da Largura de Banda Ocupada por CT para o Enlaces A e B Respectivamente (com  $A=1$  e  $B=1$ ) (Elaborado pelo autor, 2008)

A Figura 33 mostra a evolução da “utilização” dos enlaces pelos BCs com o aumento da carga associada ao CT2 na segunda configuração de parâmetros do fator “ $w(lsp)$ ” (com  $A=1$  e  $B=1$ ), para as mesmas configurações de carga.

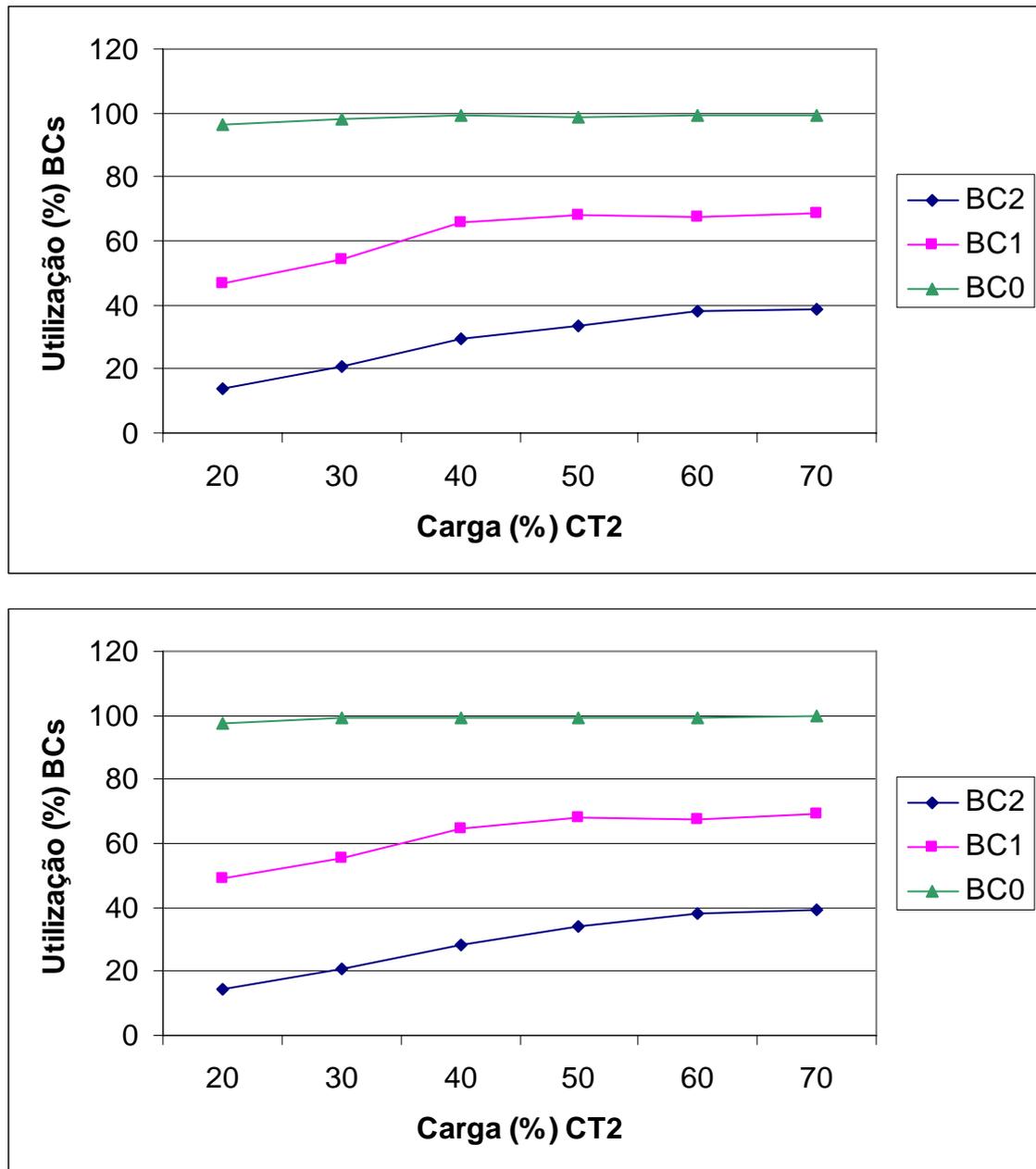


Figura 33 - Evolução da Largura de Banda Ocupada por BC para os Enlaces A e B Respectivamente (com A=1 e B=1) (Elaborado pelo autor, 2008)

A partir da análise dos resultados de simulação dos gráficos, chega-se às seguintes conclusões:

- a) as conclusões atingidas na análise das simulações, considerando a primeira configuração do fator “w(lsp)” (com A=1 e B=0), são mantidas. Quando existe recursos disponível (não-utilizado) pelo CT2 o mesmo pode ser utilizado pelos LSPs associados aos CTs 0 e 1. Quando começa o aumento da carga associada ao CT2, as cargas associadas aos CTs 0 e 1 começam a decair (a preempção de LSPs associados

- aos CTs 0 e 1 ocorrerá no algoritmo ADAPT-RDM) com a intenção de preservar o isolamento de tráfego configurado para o CT2 (BC2 que contém apenas o CT2 pode ocupar até 40% do valor dos enlaces);
- b) a preempção dos LSPs nos CTs 0 e 1 ocorre até os BC2 atingir seu valor máximos de utilização especificado (40%). A partir desse momento não mais torna-se necessário a preempção de LSPs, as novas requisições de LSP não podem mais serem estabelecidas e começam a ser bloqueadas;
  - c) a segunda configuração do fator “w(lsp)” (com A=1 e B=1) varia com relação a primeira configuração do fator “w(lsp)” (com A=1 e B=0) na utilização da largura de banda dos CTs, até a máxima configuração de BC2 (40% de ocupação dos enlaces) ser atingida. Isso pode-se justificar pelo fato do perfil dos LSPs preemptados serem alterados, já que nessa configuração uma baixa largura de banda, além da baixa prioridade, é considerada como critério de preempção quando há a ocorrência de disputa por recursos entre CTs.

## 7.2 AVALIAÇÃO DE ALGORITMOS DE SELEÇÃO DE CAMINHOS BASEADOS EM RESTRIÇÕES – CSPF E MINPREPTDSTE

Como cenário para a avaliação das implementações dos algoritmos de cálculo de caminho baseados em restrições CSPF e MinPreptDSTE foi proposta a topologia da Figura 34.

Esta topologia é de interesse para este cenário pois permite a possibilidade de mais de um caminho, com diferentes quantidades de saltos, entre o nó origem e o nó de destino das requisições de LSPs. As origens e destinos de LSPs na rede foram configuradas da seguinte forma:

- a) três fontes geradoras de LSPs foram admitidas na simulação (S1, S2 e S3), todas com o mesmo destino (D);
- b) “S1” tem três opções de caminho, onde cada caminho tem uma quantidade distinta de saltos até o destino (caminho “C”, caminho “A-B”, e caminho “D-E-F”);
- c) “S2” e “S3” são tráfegos de interferência e apresentam cada um deles apenas uma opção de caminho até o destino (D). “S2” pelo caminho “B” e “S3” pelo caminho “E-F”.

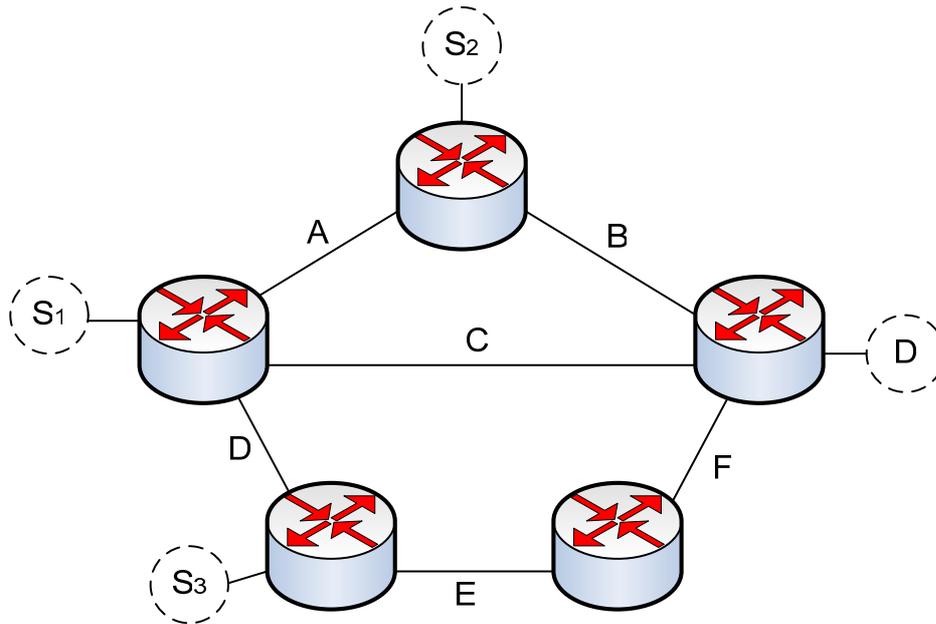


Figura 34 - Topologia de rede utilizada para a avaliação do algoritmo MinPreptDSTE (Elaborado pelo autor, 2008)

Os enlaces da rede A, B, C, D, E e F foram configurados com 622 Mbps cada (valor equivalente ao STM-4 da tecnologia SDH). Três CTs foram configurados na rede (CT0, CT1 e CT2). A máxima porcentagem de banda alocada para os BCs (MAX\_BC[n]) de acordo com o modelo RDM para os enlaces A e B, foram configurados de acordo com o quadro 21.

BC	MAX_BC (%)	MAX_BC (Equivalente em Mbps)
0	100	622
1	70	435,4
2	40	248,8

Quadro 21 - Configurações dos BCs para o Cenário de Simulação (Elaborado pelo autor, 2008)

Neste cenário de simulação foi realizada uma comparação de desempenho entre os dois algoritmos de seleção de caminhos propostos (CSPF e MinPreptDSTE), que utilizam a estratégia do algoritmo ADAPT-RDM para se adequarem ao modelo RDM. A métrica de desempenho considerada foi a quantidade de preempções ocorridas em rede para diferentes situações de carga de tráfego.

A carga de chegada de LSPs foi simulada da seguinte maneira:

- a) o intervalo entre chegadas de LSPs foi modelado exponencialmente tendo como média os seguintes valores:
  - 2 segundos para LSPs associados ao CT0,

- 4 segundos para LSPs associados ao CT1, e
  - 8 segundos para LSPs associados ao CT2.
- b) a duração do estabelecimento de cada LSP na rede foi modelada exponencialmente com média de 100 segundos;
- c) a largura de banda de cada LSP foi distribuída uniformemente entre 5Mbps e 20 Mbps.;
- d) o número de LSPs gerados na simulação foi variado entre 200 e 600 (incrementando-se o número de LSPs gerados de 50 em 50). O critério de parada de simulação utilizado foi portanto o número LSPs gerados.

O intuito desta configuração foi simular situações onde o compartilhamento de banda esteja ocorrendo, por exemplo, LSPs associados ao CT0 utilizam a banda momentaneamente disponível nos CTs 1 e 2. Da mesma forma, os LSPs associados ao CT1 utilizam a banda momentaneamente disponível no CT2. Por isso as diferenças no tempo dos intervalos de chegada entre os LSPs para os três CTs configurados em rede. Com essa configuração há uma tendência em que os LSPs associados ao CT0 utilizem a largura de banda não-utilizada pelos CTs 1 e 2, e da mesma forma, que os LSPs associados ao CT1 utilizem a largura de banda não utilizada pelo CT2. Dessa forma, em uma situação de disputa por recursos entre CTs, a necessidade de preempção na rede para o cumprimento das restrições de banda (BCs) do modelo RDM pode ser avaliada.

Para cada conjunto de números definidos nos parâmetros, foram realizadas quatro simulações, variando-se a semente do gerador de números aleatórios utilizado (variadas através da função “*srand()*” implementada como biblioteca da linguagem C). Então, calculou-se a média de cada métrica de desempenho desejada, o desvio padrão e o intervalo de confiança de 90%, usando a distribuição de *Student* (Spiegel, 1993). Os resultados são mostrados no quadro 22.

N° de LSPs Geradas	Quantidade Total de Preempções na Rede													
	Semente1		Semente2		Semente3		Semente4		Média		Desvio Padrão		IC (90%)	
	CSPF	MinPrept DSTE	CSPF	MinPrept DSTE	CSPF	MinPrept DSTE	CSPF	MinPrept DSTE	CSPF	MinPrept DSTE	CSPF	MinPrept DSTE	CSPF	MinPrept DSTE
200	43	36	43	36	31	25	35	30	38	32	6.00	5.32	0.38	0.33
250	59	50	53	51	48	38	52	49	53	47	4.55	6.06	0.29	0.38
300	66	50	67	60	58	42	62	58	63	53	4.11	8.23	0.26	0.52
350	73	55	79	75	69	58	78	63	75	63	4.65	8.81	0.29	0.55
400	79	68	86	84	83	68	84	71	83	73	2.94	7.63	0.18	0.48
450	93	76	93	87	92	77	90	78	92	80	1.41	5.07	0.09	0.32
500	102	86	99	91	98	85	93	87	98	87	3.74	2.63	0.24	0.17
550	110	94	107	93	106	90	98	91	105	92	5.12	1.83	0.32	0.11
600	114	109	116	104	113	98	103	98	112	102	5.80	5.32	0.36	0.33

Quadro 22 - Resultados de Simulação – Algoritmos CSPF e MinpreptDSTE (Fatores “w(LSP)” A=1 e B=0) (Elaborado pelo autor, 2008)

Os pesos do fator de preempção em cada CT (fator “ $w(lsp)$ ”) foram configurados como  $A = 1$  e  $B = 0$ , o que significa que nessas simulações foi considerado importante a preempção de LSPs de baixa prioridade associados aos CTs, em caso da necessidade da preempções para adequação às máximas configurações das BCs.

Na Figura 35 são mostrados os resultados comparativos (para os valores médios das simulações) entre os dois algoritmos avaliados, com o número de preempções ocorridas na rede (métricas de desempenho utilizada) para diferentes cenários de carga de tráfego na rede.

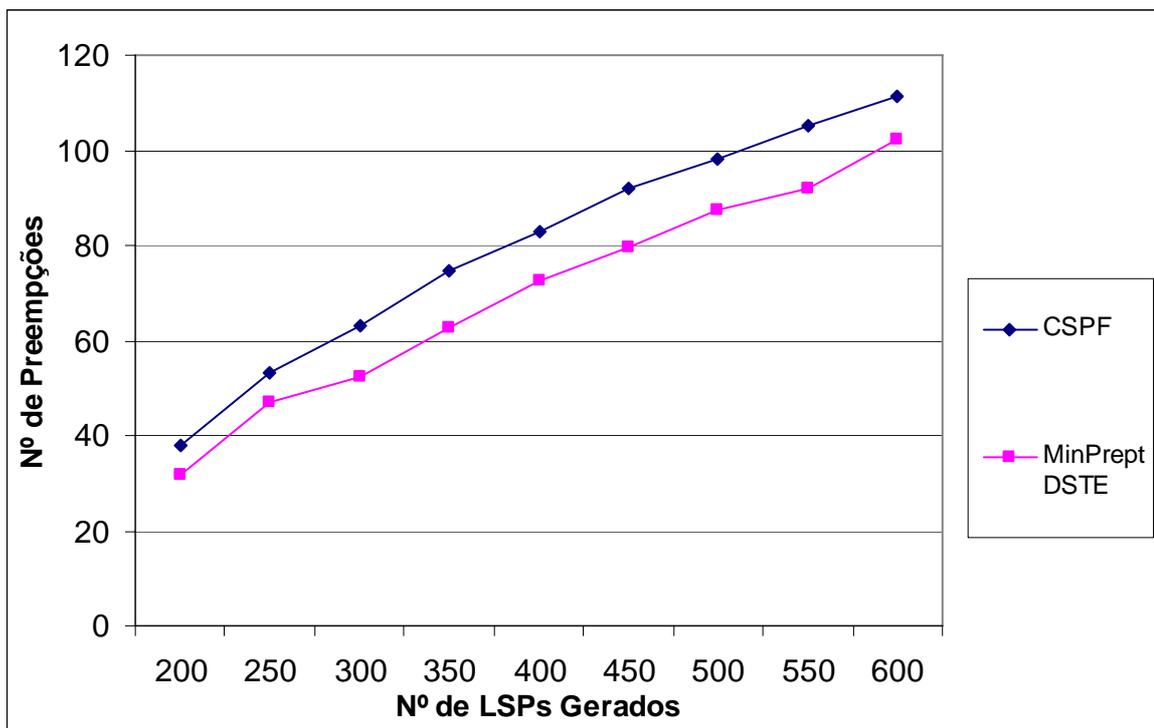


Figura 35 - Comparação do número de preempções entre os dois algoritmos de seleção de caminhos (Elaborado pelo autor, 2008)

Avaliou-se também a quantidade de preempções de LSPs ocorridas individualmente em cada CT, e os resultados são mostrados nas Figuras 36 e 37.

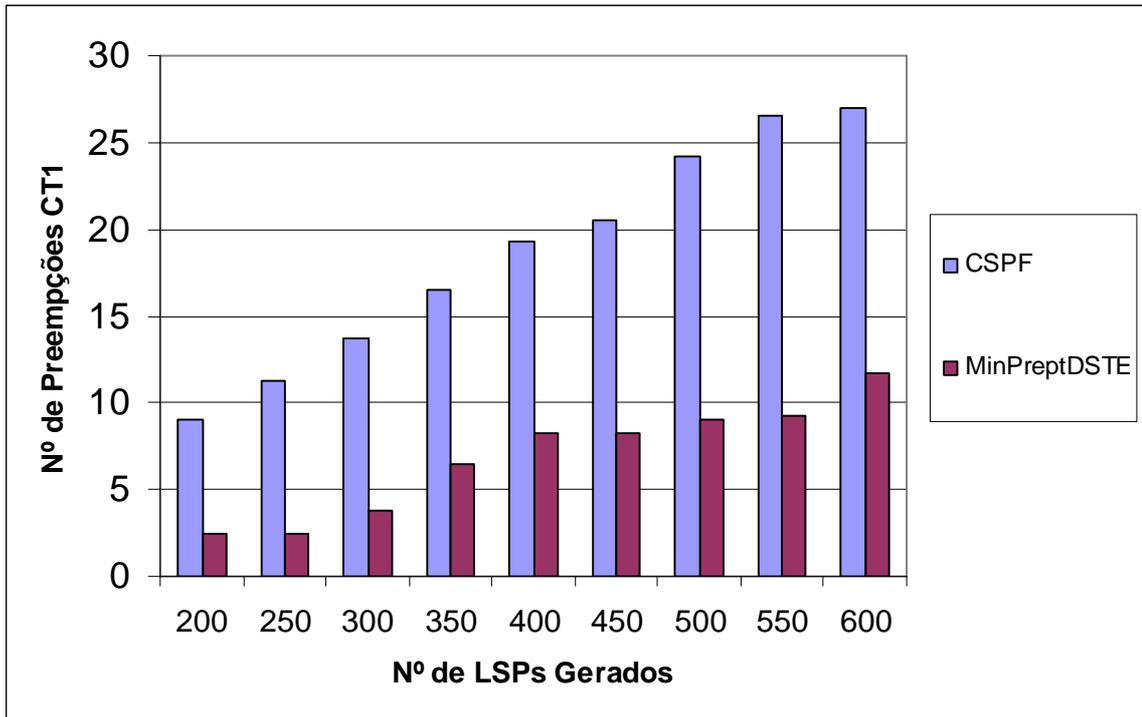


Figura 36 - Evolução do Número de Preempções Ocorridas nos LSPs associados ao CT1 (Elaborado pelo autor, 2008)

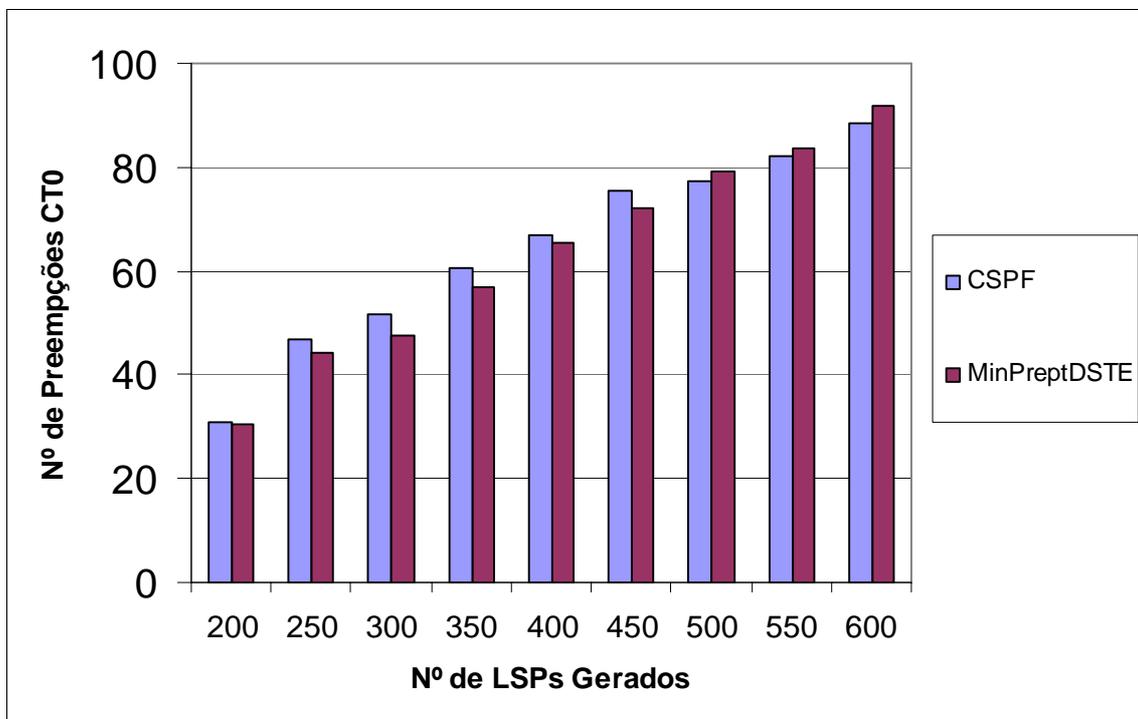


Figura 37 - Evolução do Número de Preempções Ocorridas nos LSPs associados ao CT0 (Elaborado pelo autor, 2008)

A partir da análise dos resultados de simulação dos gráficos das Figura 36 e 37, chegou-se às seguintes conclusões:

- a) o algoritmo MinPreptDSTE proposto apresentou um desempenho superior quando comparado ao algoritmo CSPF, baseado apenas na minimização do número de saltos, apresentando uma quantidade total menor de preempções ocorridas na rede para todos os cenários de carga de tráfego avaliados.
- b) através dos resultados mostrados na Figura 36, observa-se que um grande diferencial do algoritmo MinpreptDSTE quando comparado com o CSPF foi a minimização da quantidade de preempções ocorridas no CT1. Esse resultado vem ratificar o propósito do algoritmo MinpreptDSTE de buscar pelo caminho que apresente os enlaces com a menor quantidade de banda compartilhada entre tipos de classes distintos. Assim, por exemplo, LSPs associados ao CT2 (classe mais prioritária) buscam prioritariamente por caminhos em que a banda compartilhada com CT1 seja a menor possível, seja este caminho mais longo ou mais curto (independente de uma maior ou menor quantidade de saltos).
- c) a quantidade de preempções considerando apenas os LSPs associados ao CT0, mostrada na Figura 37, apresentou resultados mais próximos para ambos algoritmos avaliados, com até mesmo o algoritmo CSPF (menor quantidade de saltos) apresentando desempenho superior ao MinpreptDSTE nos casos de maior carga gerada (maior nº de LSPs gerados). Isso pode-se justificar pelos seguintes fatores:
- O intervalo entre chegadas de LSPs associados ao CT0 ser o menor entre todos os CTs, sendo o tráfego do CT0 na simulação o principal causador de saturação e disputa por recursos nos enlaces da rede. Como o tempo de permanência de estabelecimento dos LSPs foi igualmente configurado para todos os CTs (100 segundos), significa que a alta demanda de LSPs associados ao CT0 na rede, faz com que os enlaces estejam sempre com uma alta ocupação da restrição de banda (BC) reservada para o CT0 na hierarquia do modelo RDM. Lembrando que o espaço de restrição de banda que é reservado exclusivamente para LSPs associados ao CT0 é calculado subtraindo-se a restrição de banda BC0 da restrição de banda BC1.
  - Os LSPs associados ao CT0 dos geradores de tráfego de interferência “S2” e “S3” apresentam apenas uma opção de caminho até o destino (D). Apenas para o gerador de tráfego “S1” possui mais de um caminho até o destino (D). Nesses casos não há a possibilidade de um caminho alternativo ao de menor quantidade de saltos ser escolhido. Acredita-se que esse aspecto, aliado à alta taxa de geração de LSPs do CT0 (menor valor entre chegadas de LSPs entre os

três CTs), tenha contribuído para o desempenho mais próximo dos dois algoritmos nos resultados de simulação nessa comparação. É esperado que para outras topologias de rede, onde para cada nó origem-destino exista mais de uma opção de caminho, o algoritmo MinpreptDSTE apresente desempenho mais superior em termos de minimização da ocorrência de preempções para os LSPs do CT0 configurados em rede. Para tal, cabe uma avaliação comprovativa.

## 8 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

A utilização da tecnologia DS-TE nas redes de backbones é uma alternativa que visa prover uma infra-estrutura de rede multiserviço única capaz de suportar aplicações multimídia com requisitos distintos de Qualidade de Serviço (QoS). A partir da implementação do modelo de restrição de banda RDM nos enlaces de uma rede DS-TE é possível uma melhor eficiência na utilização dos recursos, simultaneamente com o isolamento do tráfego entre diferentes tipos de classes (CTs) configurados na rede, que requisitam diferentes necessidades de serviços em rede.

Nesta dissertação buscou-se propor e avaliar algoritmos para gerência de banda compartilhada entre classes e seleção de caminhos para LSPs, em um cenário de implantação conforme ao DS-TE habilitado com modelo de restrição de banda RDM. A avaliação de desempenho foi feita através da implementação de um simulador de propósito específico em linguagem C, desenvolvido exclusivamente para avaliar os algoritmos propostos.

Uma estratégia de gerência de banda compartilhada entre CTs suportada pelo algoritmo proposto ADAPT-RDM na seleção do melhor caminho para um LSP associado a determinado CT, por sua vez, é de fundamental importância para que haja a manutenção dos parâmetros e compatibilidade com o modelo RDM. A estratégia de gerência de banda compartilhada proposta é integrada com o algoritmo de seleção de caminho de LSPs. A estratégia de gerência de banda compartilhada ADAPT-RDM foi avaliada e mostrou-se eficiente quanto aos seus objetivos de permitir o compartilhamento de banda entre CTs distintos em cenários de não-disputa por recursos, bem como garantir os parâmetros de banda configurados para cada uma delas em momentos de disputa por recursos.

Uma implementação do algoritmo para a seleção de caminho *Constrained Shortest Path First* (CSPF), baseado no menor número de saltos, integrado com a estratégia de gerência de banda compartilhada ADAPT-RDM foi proposto. Posteriormente um segundo algoritmo para a seleção de caminho MinPreptDSTE, também integrado com a estratégia de gerência de banda compartilhada ADAPT-RDM, foi proposto. Na proposta do algoritmo MinPreptDSTE a minimização do número de preempções de LSPs é o objetivo, o que, dentre outras vantagens, reduz a interrupção do serviço, diminui a necessidade do tráfego de sinalização na rede bem como a necessidade de re-roteamento (quando esta funcionalidade está habilitada) dos LSPs

preemptados. Na avaliação dos algoritmos realizada, o algoritmo de seleção de caminho proposto MinPreptDSTE mostrou desempenho superior ao algoritmo CSPF, o qual é baseado apenas no critério de menor número de saltos (*hops*). Com esta validação inicial positiva é possível em trabalhos futuros partir para a análise do desempenho dos algoritmos propostos em outras topologias de rede.

Devido à característica centralizadora da solução DS-TE proposta, a mesma é aplicável de forma eficiente em redes de *backbones* de pequeno e médio porte, como por exemplo, o *backbone* regional ou nacional de uma rede corporativa. Para implementações visando topologias de rede de maior porte é necessário um estudo no que diz respeito à escalabilidade da solução proposta que foge ao escopo deste trabalho.

São identificadas como perspectivas de continuidade deste trabalho os seguintes trabalhos futuros:

- a) simulação do comportamento dos algoritmos propostos com uma Matriz de Tráfego real associada a uma topologia de rede já existente, nos moldes do projeto CHESF (Projeto Chesf-Universidade Salvador), desenvolvido no Núcleo de Pesquisa em Redes e Computação (NUPERC) da Universidade Salvador. Desta forma, seria possível a avaliação do comportamento dos algoritmos propostos com uma matriz de tráfego de uma rede real coletada via medição;
- b) tentativa de reproduzir numa rede DS-TE em um backbone experimental as propostas de avaliação de desempenho feitas via simulação neste trabalho visando comparar os resultados obtidos com os resultados da avaliação de desempenho dos algoritmos feitos com a técnica de medição.
- c) avaliar o desempenho dos algoritmos propostos através de simulação em outras topologias de rede.
- d) avaliação do comportamento dos algoritmos propostos com outros tipos de geradores de tráfego e outras distribuições de probabilidade para a geração de tráfego de LSPs.

## REFERÊNCIAS

- ANDERSSON, L.; SWALLOW, G. The Multiprotocol Label Switching (MPLS) Working Group decision on MPLS signaling protocols, **IETF**, RFC 3468, 2003.
- AWDUCHE, D.; MALCOLM, J.; AGOGBUA, J.; O'DELL, M.; MCMANUS, J.; Requirements for Traffic Engineering over MPLS, **IETF**, RFC 2702, 1999a.
- AWDUCHE, D; "MPLS and Traffic Engineering in IP Networks **IEEE Communications Magazine**, Volume 37, nº 12, páginas 42-47, ISSN: 0163-6804, 1999b.
- AWDUCHE, D; CHIU, A.; ELWALID, A.; WIDJAJA, I.; XIAO, X. Overview and Principles of Internet Traffic Engineering, **IETF**, RFC 3272, 2002.
- AWDUCHE, D.; BERGER, L.; GAN, D.; LI, T.; SRINIVASAN, V.; SWALLOW, G. "RSVP-TE: Extensions to RSVP for LSP Tunnels", **IETF**, RFC3209, 2001.
- BLAKE, S.; BLACK, D.; CARLSON, M.; DAVIES, E.; WANG, Z.; Weiss, W. An Architecture for Differentiated Services, RFC 2475, 1998.
- DE OLIVEIRA, J.C.; SCOGLIO, C.; AKYILDIZ, I.F.; UHL, G.; SMITH, J.A., A New Topology-Aware LSP Preemption Policy for DiffServ-MPLS Networks. In: **Proceedings... IEEE NETWORKS 2002** (Joint Conference: ICWLHN 2002 and ICN 2002), Atlanta, Georgia, EUA, Agosto, 2002.
- DE OLIVEIRA, J.; SCOGLIO, C.; AKYILDIZ, I.; UHL, G. New Preemption Policies for DiffServ-Aware Traffic Engineering to Minimize Rerouting in MPLS Networks, **IEEE/ACM Transactions on Networking**, Volume 12, Nº. 4, páginas 733-745, 2004.
- DE OLIVEIRA, J.; VASSEUR, JP.; CHEN, L.; SCOGLIO, C. Label Switched Path (LSP) Preemption Policies for MPLS Traffic Engineering, **IETF**, RFC 4829, 2007.
- GUIDO, Marcelo. **Proposta para estender as funcionalidades do RSVP-TE**, 2004. 89p. Dissertação de Mestrado em Engenharia de Computação - Instituto de Pesquisas Tecnológicas de São Paulo – IPT, São Paulo 2004.
- HEINANEN, J.; BAKER, F.; WEISS, W.; WROCLAWSKI, J.; Assured Forwarding PHB Group, IETF, RFC 2597, 1999.
- JACOBSON, V.; NICHOLS, K.; PODURI, K.; "An Expedited Forwarding PHB, **IETF**, RFC 2598, 1999.
- KE, Y.; LIN, Z.; HUI-MIN, Z. A Preemption-Aware Path Selection Algorithm for DiffServ/MPLS Networks, **IEEE, IP Operations and Management**, Páginas: 129- 133, ISBN: 0-7803-8836-4, 2004.
- KUROSE, James; ROSS, Keith. **Redes de computadores e a internet: uma abordagem top-down**, 3a edição, Editora Addison-Wesley, Páginas: 656, ISBN: 9788588639188, 2006.

LAI, W. Bandwidth Constraints Models for DiffServ-aware MPLS Traffic Engineering: Performance Evaluation, **IETF**, RFC 4128, 2005.

LE FAUCHEUR, F. Russian Dolls Bandwidth Constraints Model for DiffServ-Aware MPLS Traffic Engineering, **IETF**, RFC 4127, 2005a.

LE FAUCHEUR, F. Protocol Extensions for Support of DiffServ-aware MPLS Traffic Engineering, **IETF**, RFC 4124, 2005b.

LE FAUCHEUR, F.; LAI, W. Maximum Allocation Bandwidth Constraints Model for DiffServ-Aware MPLS Traffic Engineering, **IETF**, RFC 4125, 2005.

LE FAUCHEUR, F.; LAI, W. Requirements for Support of Differentiated Services-Aware MPLS Traffic Engineering, **IETF**, RFC 3564, 2003.

LE FAUCHEUR, F.; WU, L., DAVIE, B.; DAVARI, S.; VAANANEN, P.; KRISHNAN, R.; CHEVAL, P.; HEINANEN, J. Multi-Protocol Label Switching (MPLS) Support of Differentiated Services, **IETF**, RFC 3270, 2002.

MACDOUGALL, M. H. **Simulating Computer Systems Techniques and Tools**, The Massachusetts Institute of Technology Press. 1987.

MARTINS, Joberto. **Qualidade de serviço em redes IP**: princípios básicos, parâmetros e mecanismos. Setembro, 1999. Disponível em: <http://www.jsmnet.com>. acesso em: junho 10, 2006.

MARTINS, J. ; AIDAROUS, S. ; PLEVYAK, T. ; LEVINE, P. ; STILLER, B.; FUMAGALLI, A.; ARACIL, J. ; VALCARENGHI, L. **Managing IP Networks - Challenges and Opportunities** 1. ed. New Jersey - USA: John Wiley & Sons/ IEEE Press, 2003.

MEDHI, Deepankar; PIÓRO, Michal. **Routing, Flow, and Capacity Design in Communication and Computer Networks**. p. 756, San Francisco CA, EUA, ISBN: 0125571895, 2004.

MINEI, Ina; LUCEK, Julian. **MPLS-Enabled Applications**: emerging developments and New Technologies, John Wiley & Sons Ltd, p. 432, ISBN-10: 0470014539, 2005.

MINGYING, Z.; XIAOMINBG H.; WU Y. Minimizing Preemption Cost for DiffServ-Aware MPLS Traffic Engineering", **IEEE**, First International Conference on Communications and Networking in China (ChinaCom), Pequim 2006, Páginas: 1-5, ISBN: 1-4244-0463-0, 2006.

MIRAS, D. A Survey on Network QoS Needs of Advanced Internet Applications, 2002. Disponível em: <http://www.cs.ucl.ac.uk/staff/D.Miras>. acesso em: Dez. 10, 2007.

NICHOLS, K.; JACOBSON, V.; ZHANG, L. “A Two-bit Differentiated Services Architecture for the Internet”, IETF, RFC 2638, 1999.

OLIVEIRA, A.; MATEUS, G. **Using Genetic Algorithm to LSP Setup in MPLS Networks**, Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos – SBRC, Curitiba-Paraná, 2006.

OSBORNE, Eric; SIMBA, Ajay. **Engenharia de Tráfego com MPLS – Projeto, Configuração e Gerenciamento do MPLS-TE para Otimização de Desempenho de Rede**, Cisco System. São Paulo: Editora Campus Ltda, 2003.

PINTO NETO, W.; MARTINS, J.; BRITO, S. **Algoritmos de Seleção de Caminho e Gerenciamento de Banda Compartilhada conforme ao Modelo RDM para Classes de Tráfego em Rede DS-TE**, publicado nos anais do 26º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC, Rio de Janeiro-RJ, 2008.

PINTO NETO, Walter; MARTINS, Joberto. A RDM-like Bandwidth Management Algorithm for Traffic Engineering with DiffServ and MPLS Support. In: proceedings of 15th International Conference on Telecommunications – ICT 2008, Páginas 1-6, St. Peterburg, Rússia, ISBN: 978-1-4244-2035-3 , 2008a.

PINTO NETO, Walter; MARTINS, Joberto. **Adapt-RDM: a Bandwidth Management Algorithm suitable for DiffServ Services Aware Traffic Engineering**, IEEE/IFIP Network Operations & Management Symposium, NOMS 2008, Salvador-Bahia, Páginas: 975-978, ISSN: 1542-1201, 2008b.

PROJETO CHESF-UNIVERSIDADE SALVADOR. **Metodologia de Dimensionamento e Identificação de Modelos de Suporte à Qualidade de Serviço (QoS) em Redes IP com tecnologias DiffServ/MPLS**. Projeto de Pesquisa desenvolvido na Universidade Salvador para a Companhia Hidro-Elétrica do São Francisco (CHESF). 2004-2006.

RAMAKRISHNAN, K.; FLOYD, S.; BLACK, D. The Addition of Explicit Congestion Notification (ECN) to IP, IETF, RFC 3168, 2001.

ROSEN, E.; VISWANATHAN, A; CALLON, R. Multiprotocol Label Switching Architecture, IETF, RFC 3031, 2001.

ROUGHAN, M.; ZHANG, Y.; DIFFIELD, N.G.; GREENBERG, A., Fast Accurate Computation of Large-scale IP Traffic Matrices from Link Loads, **ACM SIGMETRICS**, **03**, v. 31, n. 1, p. 206–217, ISSN:0163-5999, 2003.

SCOGLIO, C.; DE OLIVEIRA, J.; I. AKYILDIZ, F.; UHL, G. **A New Preemption Policy for DiffServ-Aware Traffic Engineering to Minimize Rerouting**. In: Proceedings of IEEE INFOCOM 2002, Volume 2, Páginas 695- 704, ISSN: 0743-166X, 2002.

SCOGLIO, C.; ANJALI, T.; DE OLIVEIRA, J.C.; AKYILDIZ, I.F.; UHL, G. TEAM: A Traffic Engineering Automated Manager for DiffServ-Based MPLS Networks, **IEEE Communications Magazine**, v. 42, n. 10, p. 134-145, ISSN: 0163-6804, 2004.

SHENKER S.; PARTRIDGE, C.; GUERIN, R. Specification of Guaranteed Quality of Service, **IETF**, RFC 2212, 1997.

SPIEGEL, Murray R. **Estatística**. tradução e revisão técnica de Pedro Constantino, 3ª ed, São Paulo: Makron books, 1993.

SZETO, W.; BOUTABA, R.; IRAQI, Y. **Dynamic Online Routing Algorithm for MPLS Traffic Engineering**. In: Proceedings of the Second International IFIP-TC6 Networking Conference on Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; and Mobile and Wireless Communications, 2002. p. 936 - 946, Portal ACM, ISBN:3-540-43709-6.

SZVIATOVSZKI, B.; SZENTESI, A.; JUTTNER, A. Minimizing Re-routing in MPLS Networks with Preemption-Aware Constraint-based Routing, **Computer Communications**, v. 25, n. 11-12, p. 1076-1084, 2002.

TABATABAEE, V.; BHATTACHARJEE, B.; La, R.J.; SHAYMAN, M.A. Differentiated Traffic Engineering for QoS Provisioning, In: **Proceedings...** 24th IEEE Annual Joint Conference of the IEEE Computer and Communications Societies - INFOCOM, 2005., v. 4, p. 2349- 2359, 2005.

TANENBAUM, Andrew S. “Redes de Computadores”, 4 ed. São Paulo: Editora Campus, 2003.

TRIMINTZIOS, P.; ANDRIKOPOULOS I.; PAVLOU, G.; FLEGKAS, P.; GRIFFIN, D.; GEORGATSOS, P.; GODERIS, D.; T'JOENS, Y.; GEROGIADIS, L.; JACQUENET, C.; EGAN, R. A Management and Control Architecture for Providing IP Differentiated Services in MPLS-Based Networks, **IEEE Communications Magazine**, v. 39, n. 5, p. 80-88, ISSN: 0163-6804, 2001.

XIAOMING H.; QINGXIN C.; MINGYING Z. **A Improved Algorithm for Minimizing Preemption Cost in MPLS Networks**, *Wireless Communications, Networking and Mobile Computing*, 2006. In: WiCOM 2006. International Conference v. n. 22-24 Sept. 2006 p. 1 – 4 Digital Object Identifier 10.1109/WiCOM.2006.373.

XIPENG X.; HANNAN, A.; BAILEY, B.; NI, L.M. Traffic Engineering with MPLS in the Internet Network, *IEEE*, v. 14, n. 2, 2000, p. 28 - 33 Digital Object Identifier 10.1109/65.826369

ZHANG, D.; IONESCU, D. **QoS Performance Analysis in Deployment of DiffServ-Aware MPLS Traffic Engineering**, *IEEE/Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, 2007.

## ANEXO A – CÓDIGO DAS PRINCIPAIS FUNÇÕES UTILIZADAS PARA SIMULAÇÃO

```

/***** ESTRUTURAS E DEFINIÇÕES *****/
#define MaxCaminhos 10
#define MaxSaltos 5
#define MaxH 50
#define MaxClassType 3
#define LINKS 6
# define MAX_SIMULATIONS 1
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <malloc.h>
#include <conio.h>

typedef double real;

struct lsp {
    double Carga;
    double CargaReduzida;
    int H;
    int src;
    int dest;
    int NumParOD;
    int PrioridadeOD;
    int NumCaminhosOD;
    int NumHops;
    char MatCaminhos[MaxCaminhos + 1][MaxSaltos + 1];
    char CaminhoEscolhido [MaxSaltos + 1];
    int CT;
    int BC;
};

struct no {
    struct lsp item;
    struct no *prox;
};

struct lista {
    struct no *primeiro;
    struct no *ultimo;
};

struct BC_RDM {
    struct lista LPreemptbla[MaxH + 1];
};

struct link {
    double CargaEnlace;
    double CargaEnlaceAtual;
    double CargaEnlaceAnterior;
    double CargaResidual;
    double BandaPreemptada;
    double TotalBandaPreemptada;
    double BandaPreemptadaCT[MaxClassType];
    char DescEnlace;
    int NumEnlace;
    int CustoEnlace;
};

```

```

        int nosrc;
        int nodest;
        struct BC_RDM CT[MaxClassType];
        struct lista LLSPs;
        struct lista LPreempted;
        double BCAtual[MaxClassType];
        double BCACumulado[MaxClassType];
        double BCACumuladoAUX[MaxClassType];

} lnk [LINKS+1];

struct simul {
    char name [50];
    struct evchain *evc_begin;
    struct evchain *evc_end;
} sim [MAX_SIMULATIONS];

struct evchain {
    double ev_time;
    int ev_tkn;
    int ev_type;
    struct evchain *ev_prior;
    struct evchain *ev_next;
    struct no *ev_tkn_p;
};

/***** VARIÁVEIS GLOBAIS *****/
char arquivo_saida[20];
struct lista LPreempted;
struct lista LBlocked;
int Total_LSPs_Preempted=0;
int Total_LSPs_Blocked=0;
int N_LSPs_Preempted[MaxClassType];
int N_LSPs_Blocked[MaxClassType];

double TotalBandaReduct = 0;
double TotalBandaPreempted = 0;
double TotalBandaAdapted = 0;
int VetorLSPsPrempatadas[MaxH +1];
double BC[MaxClassType];

//Parâmetros da Cadeia de Eventos
char arquivo_saida[20];
double clock=0;
int event;
int sn = 0;
int LSPs_Geradas=0;
int LSP_Number=0;

FILE *fp;

/***** CRIACAO DA LSP E ALOCACAO DE AREA DE MEMÓRIA *****/
struct no *create_LSP (void)
{
    struct no *LSP;
    LSP = malloc(sizeof(struct no));
    if (LSP == NULL)
    {
        printf ("\n Erro - fc creat packet - nao tem mais memoria para alocar para pkt");
    }
}

```

```

        exit (0);
    }
    return (LSP);
}

/***** PROGRAMA PRINCIPAL *****/
main()
{
    int Establish;
    struct lista CT0, CT1, CT2, L1ordem;
    struct lista LNovaReq;
    double ie_t=0.0, ia_t=0.0;
    struct no *dados;
    int i,j,n_lnk=0;
    int nEnl;

    double TotalBandaPreemptada = 0;
    sprintf(arquivo_saida, "Estatisticas.txt");
    fp = fopen(arquivo_saida,"w");

    /**** INICIALIZAÇÃO LISTAS PARÂMETROS DA MATRIZ DE TRÁFEGO POR CT *****/
    inicia_lista(&CT0);
    inicia_lista(&CT1);
    inicia_lista(&CT2);
    inicia_lista(&L1ordem);
    inicia_lista(&LNovaReq);
    inicia_lista(&LPreempted);
    inicia_lista(&LBlocked);
    Init_evchain();
    for (j=1; j<= LINKS; j++)
        inicia_lista(&lnk[j].LLSPs);
    for (j=1; j<= MaxH; j++)
        VetorLSPsPrempatadas[j] =0;
    for (nEnl=1; nEnl<= LINKS; nEnl++)
    {
        for (j=0; j< MaxClassType; j++)
            lnk[nEnl].BCAcumuladoAUX[j];
    }
    // Inicializando o modelo de restrição de banda RDM
    BC[2] = 40; // CT2
    BC[1] = 70; // CT1 + CT2
    BC[0] = 100; // CT0 + CT1 + CT2

    /***** CONFIGURAÇÃO DOS ENLACES *****/
    // E1 Enlace A (1->2)
    lnk[1].DescEnlace = 'A';
    lnk[1].NumEnlace = 1;
    lnk[1].CustoEnlace = 1;
    lnk[1].CargaEnlace = 622;
    lnk[1].CargaEnlaceAtual = 0;
    lnk[1].CargaEnlaceAnterior = 0;
    lnk[1].CargaResidual = 0;
    lnk[1].TotalBandaPreemptada = 0;
    lnk[1].BandaPreemptada = 0;
    lnk[1].nosrc = 1;
    lnk[1].nodest = 2;
    n_lnk++;

    // E2 Enlace B (2->3)
    lnk[2].DescEnlace = 'B';

```

```
Ink[2].NumEnlace = 2;
Ink[2].CustoEnlace = 1;
Ink[2].CargaEnlace = 622;
Ink[2].CargaEnlaceAtual = 0;
Ink[2].CargaEnlaceAnterior = 0;
Ink[2].CargaResidual = 0;
Ink[2].TotalBandaPreemptada = 0;
Ink[2].BandaPreemptada = 0;
Ink[2].nosrc = 2;
Ink[2].nodest = 3;
n_ink++;
```

```
// E3 Enlace C (1->3)
Ink[3].DescEnlace = 'C';
Ink[3].NumEnlace = 3;
Ink[3].CustoEnlace = 1;
Ink[3].CargaEnlace = 622;
Ink[3].CargaEnlaceAtual = 0;
Ink[3].CargaEnlaceAnterior = 0;
Ink[3].CargaResidual = 0;
Ink[3].TotalBandaPreemptada = 0;
Ink[3].BandaPreemptada = 0;
Ink[3].nosrc = 1;
Ink[3].nodest = 3;
n_ink++;
```

```
// E4 Enlace D (1->4)
Ink[4].DescEnlace = 'D';
Ink[4].NumEnlace = 4;
Ink[4].CustoEnlace = 1;
Ink[4].CargaEnlace = 622;
Ink[4].CargaEnlaceAtual = 0;
Ink[4].CargaEnlaceAnterior = 0;
Ink[4].CargaResidual = 0;
Ink[4].TotalBandaPreemptada = 0;
Ink[4].BandaPreemptada = 0;
Ink[4].nosrc = 1;
Ink[4].nodest = 4;
n_ink++;
```

```
// E5 Enlace E (4->5)
Ink[5].DescEnlace = 'E';
Ink[5].NumEnlace = 5;
Ink[5].CustoEnlace = 1;
Ink[5].CargaEnlace = 622;
Ink[5].CargaEnlaceAtual = 0;
Ink[5].CargaEnlaceAnterior = 0;
Ink[5].CargaResidual = 0;
Ink[5].TotalBandaPreemptada = 0;
Ink[5].BandaPreemptada = 0;
Ink[5].nosrc = 4;
Ink[5].nodest = 5;
n_ink++;
```

```
// E6 Enlace E (5->3)
Ink[6].DescEnlace = 'F';
Ink[6].NumEnlace = 6;
Ink[6].CustoEnlace = 1;
Ink[6].CargaEnlace = 622;
Ink[6].CargaEnlaceAtual = 0;
```

```

lnk[6].CargaEnlaceAnterior = 0;
lnk[6].CargaResidual = 0;
lnk[6].TotalBandaPreemptada = 0;
lnk[6].BandaPreemptada = 0;
lnk[6].nosrc = 5;
lnk[6].nodest = 3;
n_lnk++;

/***** INICIALIZACAO DAS LSPs (1->3) *****/
/***** LSP associada ao CTO *****/
LSP_Number++;
dados = create_LSP();
dados->item.CargaReduzida = 0;
dados->item.src = 1;
dados->item.dest = 3;
dados->item.NumCaminhosOD = 3;
dados->item.NumParOD = LSP_Number;
dados->item.PrioridadeOD = 0;
dados->item.H = 8 - dados->item.PrioridadeOD;
dados->item.NumHops = 0;
dados->item.CT = 0;
dados->item.Carga = 0;

// INICIALIZAÇÃO DA MATRIZ DE CAMINHO
for (i=1; i<= MaxCaminhos; i++)
{
    for (j=1; j<= MaxSaltos; j++)
    {
        dados->item.CaminhoEscolhido[j] = 0;
        dados->item.MatCaminhos[i][j] = 0;
    }
}

// DEFINIÇÃO DA MATRIZ DE CAMINHO
dados->item.MatCaminhos[1][1] = 'C'; // Caminho 1

dados->item.MatCaminhos[2][1] = 'A'; // Caminho 2
dados->item.MatCaminhos[2][2] = 'B';

dados->item.MatCaminhos[3][1] = 'D'; // Caminho 3
dados->item.MatCaminhos[3][2] = 'E';
dados->item.MatCaminhos[3][3] = 'F';

schedulep (1, 0.0, LSP_Number, dados);

/***** LSP associada ao CT1 *****/
LSP_Number++;
dados = create_LSP();
dados->item.CargaReduzida = 0;
dados->item.src = 1;
dados->item.dest = 3;
dados->item.NumCaminhosOD = 3;
dados->item.NumParOD = LSP_Number;
dados->item.PrioridadeOD = 0;
dados->item.H = 8 - dados->item.PrioridadeOD;
dados->item.NumHops = 0;
dados->item.CT = 1;
dados->item.Carga = 0;

// INICIALIZAÇÃO DA MATRIZ DE CAMINHO

```

```

for (i=1; i<= MaxCaminhos; i++)
{
    for (j=1; j<= MaxSaltos; j++)
    {
        dados->item.CaminhoEscolhido[j] = 0;
        dados->item.MatCaminhos[i][j] = 0;
    }
}

// DEFINIÇÃO DA MATRIZ DE CAMINHO
dados->item.MatCaminhos[1][1] = 'C'; // Caminho 1

dados->item.MatCaminhos[2][1] = 'A'; // Caminho 2
dados->item.MatCaminhos[2][2] = 'B';

dados->item.MatCaminhos[3][1] = 'D'; // Caminho 3
dados->item.MatCaminhos[3][2] = 'E';
dados->item.MatCaminhos[3][3] = 'F';

schedulep (1, 0.0, LSP_Number, dados);

/***** LSP associada ao CT2 *****/
LSP_Number++;
dados = create_LSP();
dados->item.CargaReduzida = 0;
dados->item.src = 1;
dados->item.dest = 3;
dados->item.NumCaminhosOD = 3;
dados->item.NumParOD = LSP_Number;
dados->item.PrioridadeOD = 0;
dados->item.H = 8 - dados->item.PrioridadeOD;
dados->item.NumHops = 0;
dados->item.CT = 2;
dados->item.Carga = 0;

// INICIALIZAÇÃO DA MATRIZ DE CAMINHO
for (i=1; i<= MaxCaminhos; i++)
{
    for (j=1; j<= MaxSaltos; j++)
    {
        dados->item.CaminhoEscolhido[j] = 0;
        dados->item.MatCaminhos[i][j] = 0;
    }
}

// DEFINIÇÃO DA MATRIZ DE CAMINHO
dados->item.MatCaminhos[1][1] = 'C'; // Caminho 1

dados->item.MatCaminhos[2][1] = 'A'; // Caminho 2
dados->item.MatCaminhos[2][2] = 'B';

dados->item.MatCaminhos[3][1] = 'D'; // Caminho 3
dados->item.MatCaminhos[3][2] = 'E';
dados->item.MatCaminhos[3][3] = 'F';

schedulep (1, 0.0, LSP_Number, dados);
/***** SIMULAÇÃO *****/
srand(11);

//while(simtime(<20000)

```

```

while(LSPs_Geradas<2000)
{
    imprime_evchain();
    fprintf(fp,"\n\n");
    dados = causep (&event, &LSP_Number);
    switch(event)
    {
        case 1:

            LSPs_Geradas++;
            schedulep (2, 0.0, dados->item.NumParOD, dados);
            LSP_Number++;

            if(dados->item.CT == 0)
            {
                ie_t = expntl(2);
                /***** LSP associada ao CT0 *****/
                dados = create_LSP();
                dados->item.CargaReduzida = 0;
                dados->item.src = 1;
                dados->item.dest = 3;
                dados->item.NumCaminhosOD = 3;
                dados->item.NumParOD = LSP_Number;
                dados->item.PrioridadeOD = (int)uniform(0,3);
                dados->item.H = 8 - dados->item.PrioridadeOD;
                dados->item.NumHops = 0;
                dados->item.CT = 0;
                dados->item.Carga = (int)uniform(5,20);

                // INICIALIZAÇÃO DA MATRIZ DE CAMINHO
                for (i=1; i<= MaxCaminhos; i++)
                {
                    for (j=1; j<= MaxSaltos; j++)
                    {
                        dados->item.CaminhoEscolhido[j] = 0;
                        dados->item.MatCaminhos[i][j] = 0;
                    }
                }

                // DEFINIÇÃO DA MATRIZ DE CAMINHO
                if(dados->item.src == 1)
                {
                    dados->item.MatCaminhos[1][1] = 'C'; // Caminho 1

                    dados->item.MatCaminhos[2][1] = 'A'; // Caminho 2
                    dados->item.MatCaminhos[2][2] = 'B';

                    dados->item.MatCaminhos[3][1] = 'D'; // Caminho 3
                    dados->item.MatCaminhos[3][2] = 'E';
                    dados->item.MatCaminhos[3][3] = 'F';
                }

                if(dados->item.src == 2)
                    dados->item.MatCaminhos[1][1] = 'B'; // Caminho 1

                if(dados->item.src == 4)
                {
                    dados->item.MatCaminhos[1][1] = 'E'; // Caminho 1
                    dados->item.MatCaminhos[1][2] = 'F';
                }
            }
        }
    }
}

```

```

        schedulep (1, ie_t, dados->item.NumParOD, dados);

        break;
    }

    if(dados->item.CT == 1)
    {
        ie_t = expntl(4);
        /***** LSP associada ao CT1 *****/
        dados = create_LSP();
        dados->item.CargaReduzida = 0;
        dados->item.src = 1;
        dados->item.dest = 3;
        dados->item.NumCaminhosOD = 3;
        dados->item.NumParOD = LSP_Number;
        dados->item.PrioridadeOD = (int)uniform(0,3);
        dados->item.H = 8 - dados->item.PrioridadeOD;
        dados->item.NumHops = 0;
        dados->item.CT = 1;
        dados->item.Carga = (int)uniform(5,20);

        // INICIALIZAÇÃO DA MATRIZ DE CAMINHO
        for (i=1; i<= MaxCaminhos; i++)
        {
            for (j=1; j<= MaxSaltos; j++)
            {
                dados->item.CaminhoEscolhido[j] = 0;
                dados->item.MatCaminhos[i][j] = 0;
            }
        }

        // DEFINIÇÃO DA MATRIZ DE CAMINHO
        if(dados->item.src == 1)
        {
            dados->item.MatCaminhos[1][1] = 'C'; // Caminho 1

            dados->item.MatCaminhos[2][1] = 'A'; // Caminho 2
            dados->item.MatCaminhos[2][2] = 'B';

            dados->item.MatCaminhos[3][1] = 'D'; // Caminho 3
            dados->item.MatCaminhos[3][2] = 'E';
            dados->item.MatCaminhos[3][3] = 'F';
        }

        if(dados->item.src == 2)
            dados->item.MatCaminhos[1][1] = 'B'; // Caminho 1

        if(dados->item.src == 4)
        {
            dados->item.MatCaminhos[1][1] = 'E'; // Caminho 1
            dados->item.MatCaminhos[1][2] = 'F';
        }

        schedulep (1, ie_t, dados->item.NumParOD, dados);
        break;
    }

    if(dados->item.CT == 2)
    {
        ie_t = expntl(8);
    }

```

```

/***** LSP associada ao CT2 *****/
dados = create_LSP();
dados->item.CargaReduzida = 0;
dados->item.src = 1;
dados->item.dest = 3;
dados->item.NumCaminhosOD = 3;
dados->item.NumParOD = LSP_Number;
dados->item.PrioridadeOD = (int)uniform(0,3);
dados->item.H = 8 - dados->item.PrioridadeOD;
dados->item.NumHops = 0;
dados->item.CT = 2;
dados->item.Carga = (int)uniform(5,20);

// INICIALIZAÇÃO DA MATRIZ DE CAMINHO
for (i=1; i<= MaxCaminhos; i++)
{
    for (j=1; j<= MaxSaltos; j++)
    {
        dados->item.CaminhoEscolhido[j] = 0;
        dados->item.MatCaminhos[i][j] = 0;
    }
}

// DEFINIÇÃO DA MATRIZ DE CAMINHO
if(dados->item.src == 1)
{
    dados->item.MatCaminhos[1][1] = 'C'; // Caminho 1
    dados->item.MatCaminhos[2][1] = 'A'; // Caminho 2
    dados->item.MatCaminhos[2][2] = 'B';

    dados->item.MatCaminhos[3][1] = 'D'; // Caminho 3
    dados->item.MatCaminhos[3][2] = 'E';
    dados->item.MatCaminhos[3][3] = 'F';
}

if(dados->item.src == 2)
    dados->item.MatCaminhos[1][1] = 'B'; // Caminho 1

if(dados->item.src == 4)
{
    dados->item.MatCaminhos[1][1] = 'E'; // Caminho 1
    dados->item.MatCaminhos[1][2] = 'F';
}

    schedulep (1, ie_t, dados->item.NumParOD, dados);
    break;
}

else
{
    printf("CT %d nao definida na simulacao!\n", dados->item.CT);
    break;
}

case 2:

Establish =TryPath_MinDSTEPreempt (dados);
if(Establish==1)
{
    if(dados->item.CT==0)
        ia_t = expntl(100);
}

```

```

        if(dados->item.CT==1)
            ia_t = expntl(100);
        if(dados->item.CT==2)
            ia_t = expntl(100);
        schedulep (3, ia_t,dados->item.NumParOD, dados);
    }

    break;

    case 3:

        RetiraLSPDesestablish(LSP_Number, dados);
        Atualiza_BCs_porEnlace_posDesestabelecimento(dados);
        Atualiza_CargadosEnlaces();
        break;
    }
}

/** FUNÇÕES AUXILIARES DOS ALGORITMOS DE ROTEAMENTO E PREEMPÇÃO */
void inicia_lista(struct lista *L)
{
    L->primeiro = (struct no *)malloc(sizeof (struct no));
    L->ultimo = L->primeiro;
    L->primeiro->prox = NULL;
}

void insere_lista(struct lsp x, struct lista *L )
{
    L->ultimo->prox = (struct no*)malloc(sizeof(struct no));
    L->ultimo=L->ultimo->prox;
    L->ultimo->item = x;
    L->ultimo->prox = NULL;
}

struct no *Retira_Lista_Final(struct lista *L)
{
    struct no *aux, *retorno;
    aux = L->primeiro;
    if (aux->prox == NULL)
    {
        printf("Lista Vazia");
        return(NULL);
    }
    else
    {
        while(aux->prox->prox != L->ultimo->prox )
            aux = aux->prox;
        retorno=L->ultimo;
        L->ultimo = aux;
        retorno=L->ultimo->prox;
        L->ultimo->prox = NULL;
        return(retorno);
    }
}

void insere_lista_ordem_decrescente(struct lsp x,struct lista *L )
{
    struct no *aux, *aux_busca;
    aux = (struct no*)malloc(sizeof(struct no));

```

```

aux->item = x;
if(aux->item.Carga <= L->ultimo->item.Carga || L->primeiro->prox == NULL)
{
    L->ultimo->prox = aux;
    L->ultimo=L->ultimo->prox;
    L->ultimo->prox = NULL;
}
else
{
    aux_busca = L->primeiro;
    while (aux_busca->prox->item.Carga > aux->item.Carga)
        aux_busca = aux_busca->prox;
    aux->prox = aux_busca->prox;
    aux_busca->prox = aux;
}
}

struct no *RetiraLSPdaLista(int L_Number, struct lista *L)
{
    struct no *aux, *retorno;
    aux = L->primeiro;
    while(aux->prox != NULL)
    {
        if(aux->prox->item.NumParOD == L_Number)
            break;
        else
            aux = aux->prox;
    }
    retorno=aux->prox;
    if(retorno!=NULL && retorno->prox != NULL)
        aux->prox = retorno->prox;
    else
    {
        L->ultimo = aux;
        L->ultimo->prox = NULL;
    }
    return(retorno);
}

void DesestabeceLSP(int L_Number, struct no *LSP)
{
    int nEnl, j;
    for (nEnl=1;nEnl<=LINKS;nEnl++);
    {
        for (j=1; j<= MaxSaltos; j++)
        {
            if (lnk[nEnl].DescEnlace == LSP->item.CaminhoEscolhido[j])
            {
                printf("LSP a ser retirada: %d\n",LSP->item.NumParOD);
                RetiraLSPdaLista(LSP->item.NumParOD, &lnk[nEnl].LLSPs);
            }
        }
    }
}

void RetiraLSPDesestablish(int LSPNumber,struct no *LSP)
{
    struct no *retorno, *LSP_Desestabelecido;
    struct evchain *retono_cancel;
    int j, nEnl;
}

```

```

for (nEnl=1;nEnl<=LINKS;nEnl++)
{
    for (j=1; j<= MaxSaltos; j++)
    {
        if (lnk[nEnl].DescEnlace == LSP->item.CaminhoEscolhido[j])
        {
            LSP_Desestabelecido = RetiraLSPdaLista(LSPNumber,&lnk[nEnl].LLSPs);
            if(LSP_Desestabelecido!=NULL)
                printf("LSP%d foi desestabelecida\n",LSP_Desestabelecido->item.NumParOD);
        }
    }
}

void insere_LSP_TodasListas (struct no *retorno, int caminhoatual)
{
    int j, nEnl;
    for (j=1; j<= MaxSaltos; j++)
    {
        for (nEnl=1;nEnl<=LINKS;nEnl++)
        {
            if (lnk[nEnl].DescEnlace == retorno->item.MatCaminhos[caminhoatual][j])
                insere_lista (retorno->item, &lnk[nEnl].LPreempted);
        }
    }
}

void Atualiza_BCAUXs_porEnlace_posPreempcao(struct no *LSP, int n, double BandadasLSPsPreemptadas)
{
    int k, j, nEnl;
    for (k=1; k<= MaxSaltos; k++)
    {
        for (nEnl=1;nEnl<=LINKS;nEnl++)
        {
            if (lnk[nEnl].DescEnlace == LSP->item.CaminhoEscolhido[k])
            {
                for(j=0; j<=n; j++)
                    lnk[nEnl].BCAcumuladoAUX[j] = lnk[nEnl].BCAcumuladoAUX[j] -
                    BandadasLSPsPreemptadas;
            }
        }
    }
}

void Atualiza_BCs_porEnlace_posDesestabelecimento(struct no *LSP)
{
    int w, j, nEnl;
    for (nEnl=1;nEnl<=LINKS;nEnl++)
    {
        for(j=0; j<MaxSaltos; j++)
        {
            if (lnk[nEnl].DescEnlace == LSP->item.CaminhoEscolhido[j])
            {
                for(w=0; w<=LSP->item.CT; w++)
                {
                    if(lnk[nEnl].BCAcumuladoAUX[w] - LSP->item.Carga>0)
                    {
                        lnk[nEnl].BCAcumuladoAUX[w] =
                        lnk[nEnl].BCAcumuladoAUX[w] - LSP->item.Carga;
                    }
                }
            }
        }
    }
}

```

```

else
    lnk[nEnl].BCAcumuladoAUX[w] = 0;
lnk[nEnl].BCAcumulado[w] = lnk[nEnl].BCAcumuladoAUX[w];
    }
    }
}

void Atualiza_BCs_porEnlace_posNovoEstabelecimento()
{
    int j, nEnl;
    for (nEnl=1;nEnl<=LINKS;nEnl++)
    {
        for(j=0; j<MaxClassType; j++)
            lnk[nEnl].BCAcumulado[j] = lnk[nEnl].BCAcumuladoAUX[j];
    }
}

void Atualiza_CargadosEnlaces()
{
    int nEnl;
    for(nEnl=1;nEnl<=LINKS;nEnl++)
    {
        lnk[nEnl].CargaEnlaceAtual = lnk[nEnl].BCAcumulado[0];
        lnk[nEnl].CargaResidual = lnk[nEnl].CargaEnlace - lnk[nEnl].CargaEnlaceAtual;
    }
}

void Retira_Lista_Elemento_TodasListas(struct no *LSP)
{
    struct no *aux, *retorno;
    int nEnl;
    int k;

    for (k=1;k<=MaxH;k++)
    {
        for (nEnl=1;nEnl<=LINKS;nEnl++)
        {
            aux = lnk[nEnl].CT[LSP->item.CT].LPreemptbla[k].primeiro;

            while(aux->prox != NULL)
            {
                if(aux->prox->item.NumParOD == LSP->item.NumParOD)
                {
                    lnk[nEnl].BandaPreemptadaCT[LSP->item.CT] =
                    lnk[nEnl].BandaPreemptadaCT[LSP->item.CT] + LSP->item.Carga;
                    break;
                }
                aux = aux->prox;
            }
            retorno=aux;
            if (retorno!=NULL)
            {
                if(retorno->prox != NULL)
                {

```

```

        retorno=aux->prox;
        aux->prox = retorno->prox;
    }
    else
        aux->prox = NULL;
    }
}
}
}

```

```

void Cria_ListaPreemptaveis_porEnlaceporCT(struct no *LSP, int Indice_Melhor_Caminho)

```

```

{
    int i, p, nEnl;
    struct no *aux;
    for(p=1;p<=MaxSaltos;p++)
    {
        for (nEnl=1;nEnl<=LINKS;nEnl++)
        {
            if(LSP->item.MatCaminhos[Indice_Melhor_Caminho][p] == lnk[nEnl].DescEnlace)
            {
                for(i=0;i<=MaxH;i++)
                {
                    aux = lnk[nEnl].LLSPs.primeiro;
                    while(aux->prox!=NULL)
                    {
                        if (aux->prox->item.H == i)
                            insere_lista_ordem_decrescente (aux->prox->item,&lnk[nEnl].CT[aux->prox->item.CT].LPreemptebla[i]);
                        aux = aux->prox;
                    }
                }
            }
        }
    }
}

```

```

void imprime_LSPs_Preemptados(struct lista *L) // Imprime elemento

```

```

{
    struct no *aux;

    aux = L->primeiro;
    while(aux->prox!=NULL)
    {
        //printf("%d -> ", aux->item.dest);
        printf("LSP%d:CT%d -> ", aux->prox->item.NumParOD, aux->prox->item.CT);
        aux = aux->prox;
    }
}

```

```

void imprime_BCAcumulado()

```

```

{
    int nEnl, i;
    fprintf(fp, "----- Valor Atual dos BCs por Enlace ----- \n\n");
    for(nEnl=1;nEnl<=LINKS;nEnl++)
    {
        fprintf(fp,"----- Enlace%d ----- \n", lnk[nEnl].NumEnlace);
        for(i=0;i<MaxClassType;i++)
        {
            if(i!=MaxClassType-1)

```

```

        fprintf(fp,"BC[%d] = %.0lf(%.2lf por cento) | CT[%d] = %lf(%.2lf por
cento)\n", i, lnk[nEnl].BCAcumulado[i],
100*lnk[nEnl].BCAcumulado[i]/lnk[nEnl].CargaEnlace, i,
lnk[nEnl].BCAcumulado[i]-lnk[nEnl].BCAcumulado[i+1],
100*(lnk[nEnl].BCAcumulado[i]-
lnk[nEnl].BCAcumulado[i+1])/lnk[nEnl].CargaEnlace);
    else
        fprintf(fp,"BC[%d] = %.0lf(%.2lf por cento) | CT[%d] = %lf(%.2lf por
cento)\n", i,
lnk[nEnl].BCAcumulado[i],100*lnk[nEnl].BCAcumulado[i]/lnk[nEnl].Carg
aEnlace, i, lnk[nEnl].BCAcumulado[i],
100*lnk[nEnl].BCAcumulado[i]/lnk[nEnl].CargaEnlace);
    }
    fprintf(fp,"\n");
}
fprintf(fp, "\n-----\n");
}
}
void imprime_lista(struct lista *L) // Imprime elemento
{
    struct no *aux;
    aux = L->primeiro->prox;
    if (aux == NULL)
        fprintf(fp, "Lista Vazia");
    else
    {
        while(aux!=NULL)
        {
            fprintf(fp, "LSP%d: %.3lf(%d-%d)(CT=%d) -> ", aux->item.NumParOD, aux-
>item.Carga, aux->item.src, aux->item.dest, aux->item.CT);
            aux = aux->prox;
        }
    }
}

int Verify_RDM_Preemption_Need (struct no *LSP, int Indice_Melhor_Caminho)
{
    struct no *aux, *retorno;
    int Prept = 0;
    int i,n, j, nEnl, w;
    int auxbraek=0;
    double BandaAcimaCT;
    Cria_ListaPreemptaveis_porEnlaceporCT(LSP, Indice_Melhor_Caminho);

    for (nEnl=1;nEnl<=LINKS;nEnl++)
    {
        for (j=1; j<= MaxSaltos; j++)
        {
            if (lnk[nEnl].DescEnlace == LSP->item.MatCaminhos[Indice_Melhor_Caminho][j])
            {
                insere_lista(LSP->item,&lnk[nEnl].LLSPs);
                for(n=0;n<MaxClassType;n++)
                {
                    if(n<=LSP->item.CT)
                        lnk[nEnl].BCAcumuladoAUX[n] = lnk[nEnl].BCAcumulado[n] +
                        LSP->item.Carga;
                    else
                        lnk[nEnl].BCAcumuladoAUX[n] = lnk[nEnl].BCAcumulado[n];
                }
            }
        }
    }
}

```

```

}

for (nEnl=1;nEnl<=LINKS;nEnl++)
{
    fprintf(fp,"Enlace%d \n", nEnl);
    imprime_lista(&lnk[nEnl].LLSPs);
    fprintf(fp, "\n");
}

for (nEnl=1;nEnl<=LINKS;nEnl++)
{
    for (j=1; j<= MaxSaltos; j++)
    {
        if (lnk[nEnl].DescEnlace == LSP->item.MatCaminhos[Indice_Melhor_Caminho][j])
        {
            for(n=LSP->item.CT-1;n>=0;n--)
            {
                BandaAcimaCT = lnk[nEnl].BCAcumuladoAUX[n] -
                ((BC[n]/BC[0])*lnk[nEnl].CargaEnlace);
                if(BandaAcimaCT>0)
                {
                    Prept = 1;
                    for(i=0;i<=MaxH;i++)
                    {
                        aux = lnk[nEnl].CT[n].LPreemptbla[i].primeiro;
                        while(aux->prox!=NULL)
                        {
                            if(BandaAcimaCT <= lnk[nEnl].BandaPreemptada)
                            {
                                break;
                                auxbraek=1;
                            }
                        }
                        retorno = Retira_Lista_Final(&lnk[nEnl].CT[n].LPreemptbla[i]);
                        Retira_Lista_Elemento_TodasListas(retorno);
                        insere_lista (retorno->item, &LPreempted);
                        lnk[nEnl].TotalBandaPreemptada = lnk[nEnl].TotalBandaPreemptada +
                        retorno->item.Carga;
                        lnk[nEnl].BandaPreemptada = lnk[nEnl].BandaPreemptada + retorno->
                        item.Carga;
                        Atualiza_BCAUXs_porEnlace_posPreempcao(retorno, n,
                        lnk[nEnl].BandaPreemptada);

                        if(auxbraek==1)
                            break;
                    }
                }
            }

            lnk[nEnl].BandaPreemptada=0; // Reinicializa a variável BandaPreemptada para
            futuros processamentos
        }
    }
}

return(Prept);
}

```

```

void RetiraLSPsPreempted()
{
    struct no *retorno, *LSP_Preemptado;
    struct evchain *retono_cancel;
    int j, nEnl, w;
    fprintf(fp, "%da PREEMPCAO NECESSARIO\n", contador);
    contador++;
    while (LPreempted.primeiro->prox!= NULL)
    {
        retorno = Retira_Lista_Final(&LPreempted);
        for(w=0;w<MaxClassType;w++)
        {
            if(w==retorno->item.CT)
                N_LSPs_Preempted[w]++;
        }
        Total_LSPs_Preempted++;
        fprintf(fp, "LSP%d(CT%d): %.0lf - Preemptada\n", retorno->item.NumParOD, retorno->item.CT, retorno->item.Carga);

        for (nEnl=1;nEnl<=LINKS;nEnl++)
        {
            for (j=1; j<= MaxSaltos; j++)
            {
                if (lnk[nEnl].DescEnlace == retorno->item.CaminhoEscolhido[j])
                    LSP_Preemptado = RetiraLSPdaLista(retorno->item.NumParOD,&lnk[nEnl].LLSPs);
            }
        }
    }
    fprintf(fp, "\n\n");
}

/***** ALGORITMOS PARA SELEÇÃO DE CAMINHOS *****/
int TryPath_MinDSTEPreempt (struct no *LSP)
{
    double menorBorrowedBand=0, menorValorBCDisponivel=0;
    double maior = 0;
    int establish=0;
    int i,j,w,k, Prept;
    int nEnl;
    int Indice_Melhor_Caminho=0;
    int num_hops_caminho =0;
    int EstabelecimentoPossivel=0;
    double Vetor_Menor_BorrowedBand_por_Caminho[MaxCaminhos+1];
    double Vetor_Menor_BC_por_Caminho[MaxCaminhos+1];
    double TotalSharedBand=0;

    for (nEnl=1;nEnl<=LINKS;nEnl++)
    {
        for(k=0;k<MaxClassType;k++)
        {
            for(i=0;i<=MaxH;i++)
                inicia_lista(&lnk[nEnl].CT[k].LPreemptebla[i]);
        }
    }

    for(i=1; i<= MaxCaminhos /*i<=LSP->item.NumCaminhosOD*/; i++)
    {
        for (j=1; j<= MaxSaltos; j++)
        {
            for (nEnl=1;nEnl<=LINKS;nEnl++)

```

```

    {
        if (lnk[nEnl].DescEnlace == LSP->item.MatCaminhos[i][j])
        {
            for(k=0;k<LSP->item.CT;k++)
            {
                TotalSharedBand = (lnk[nEnl].BCAcumulado[k]-
                lnk[nEnl].BCAcumulado[k+1]) + TotalSharedBand;
            }
            if(menorBorrowedBand <= TotalSharedBand)
                menorBorrowedBand = TotalSharedBand;
            if(menorValorBCDisponivel <= ((BC[LSP-
            >item.CT]/BC[0])*lnk[nEnl].CargaEnlace) -
            lnk[nEnl].BCAcumulado[LSP->item.CT])
                menorValorBCDisponivel = ((BC[LSP-
            >item.CT]/BC[0])*lnk[nEnl].CargaEnlace) -
            lnk[nEnl].BCAcumulado[LSP->item.CT];
        }
        TotalSharedBand=0;
    }
}
Vetor_Menor_BorrowedBand_por_Caminho[i] = menorBorrowedBand;
Vetor_Menor_BC_por_Caminho[i] = menorValorBCDisponivel;

menorBorrowedBand = menorValorBCDisponivel = 0;
}

for(i=1; /*i<= 2MaxCaminhos*/ i<=LSP->item.NumCaminhosOD; i++)
{
    if(LSP->item.Carga <= Vetor_Menor_BC_por_Caminho[i])
    {
        EstabelecimentoPossivel =1;
        if(maior <= Vetor_Menor_BorrowedBand_por_Caminho[i])
        {
            maior = Vetor_Menor_BorrowedBand_por_Caminho[i];
            Indice_Melhor_Caminho = i;
        }
    }
}

if(EstabelecimentoPossivel==0)
{
    insere_lista(LSP->item, &LBlocked);
    for(w=0;w<MaxClassType;w++)
    {
        if(w==LSP->item.CT)
            N_LSPs_Blocked[w]++;
    }
    Total_LSPs_Blocked++;
    return(establish);
}
else
{
    fprintf(fp, "LSP%d(CT=%d): %.0lf - Estabelecida\n", LSP->item.NumParOD, LSP->item.CT,
    LSP->item.Carga);

    fprintf(fp, "\n");
    for (w=1;w<=MaxSaltos;w++)
    {

```

```

        LSP->item.CaminhoEscolhido[w] = LSP-
        >item.MatCaminhos[Indice_Melhor_Caminho][w];
        LSP->item.NumHops++;
    }
    Prept = Verify_RDM_Preemption_Need (LSP, Indice_Melhor_Caminho);
    if(Prept == 1)
        RetiraLSPsPreempted();
    Atualiza_BCs_porEnlace_posNovoEstabelecimento();
    Atualiza_CargadosEnlaces();
    imprime_BCAcumulado(); //TESTE DE IMPRESSAO
    return(establish);
}
}

int TryPath_CSPF_WithPreemption (struct no *LSP)
{
    int auxbreak1=0, auxbreak2=0, establish=0;
    int i,j,w,k, Prept;
    int nEnl;
    int Indice_Melhor_Caminho=-1;
    int num_hops_caminho =0;
    for (nEnl=1;nEnl<=LINKS;nEnl++)
    {
        for(k=0;k<MaxClassType;k++)
        {
            for(i=0;i<=MaxH;i++)
                inicia_lista(&lnk[nEnl].CT[k].LPreemptebila[i]);
        }
    }
    for(i=1; /*i<= MaxCaminhos*/ i<=LSP->item.NumCaminhosOD; i++)
    {
        for (j=1; j<= /*1*/MaxSaltos; j++)
        {
            for (nEnl=1;nEnl<=LINKS;nEnl++)
            {
                if (lnk[nEnl].DescEnlace == LSP->item.MatCaminhos[i][j])
                {
                    if(LSP->item.Carga + lnk[nEnl].BCAcumulado[LSP->item.CT] >
                    (BC[LSP->item.CT]/BC[0])*lnk[nEnl].CargaEnlace)
                    {
                        auxbreak1=1;
                        break;
                    }
                    else
                        auxbreak1=0;
                }
            }
            if(auxbreak1==1)
            {
                break;
            }
        }
        if(auxbreak1==0)
        {
            fprintf(fp, "LSP%d(CT=%d): %.0lf - Estabelecida\n", LSP->item.NumParOD, LSP-
            >item.CT, LSP->item.Carga);
            fprintf(fp, "\n");
            Indice_Melhor_Caminho = i;
            establish=1;
        }
    }
}

```

```

        for (w=1;w<=MaxSaltos;w++)
        {
            LSP->item.CaminhoEscolhido[w] = LSP-
            >item.MatCaminhos[Indice_Melhor_Caminho][w];
            LSP->item.NumHops++;
        }
        Prept = Verify_RDM_Preemption_Need (LSP, Indice_Melhor_Caminho);
        break;
    }
}

if(establish==1)
{
    if(Prept == 1)
        RetiraLSPsPreempted();
    Atualiza_BCs_porEnlace_posNovoEstabelecimento();
    Atualiza_CargadosEnlaces();
    imprime_BCAcumulado();
}

else
{
    for(w=0;w<MaxClassType;w++)
    {
        if(w==LSP->item.CT)
            N_LSPs_Blocked[w]++;
    }
    Total_LSPs_Blocked++;
}
return(establish);
}

/***** GERAÇÃO DE NÚMEROS ALEATORIOS *****/
real ranf()
{
    real num_alea;
    num_alea = rand ();
    if ( num_alea < 0.)
    {
        printf ("\n\nFc ranf() - Erro grave no gerador do S.O.");
        exit(0);
    }
    while (num_alea == 0.)
    {
        num_alea = rand ();
    }
    return ( (float) num_alea / RAND_MAX );
}

/***** FUNÇÕES ADAPTADAS DO SMPL *****/
struct evchain *cancelp_tkn(int tkn)
{
    struct evchain *evc, *evc_tkn_srv;
    if ( tkn == 0 )
    {
        printf ("\n Erro - fc cancelp_tkn - erro fatal token = 0 ");
        exit (0);
    }
    if ( (sim[sn].evc_begin == NULL) && (sim[sn].evc_end == NULL) )
    {

```

```

        printf ("\n Erro - fc cancelp - cadeia de eventos vazia");
        exit (0);
    }
    evc = sim[sn].evc_begin;
    evc_tkn_srv = NULL;
    while (evc != NULL)
    {
        if ( evc->ev_tkn == tkn)
        {
            evc_tkn_srv = evc;
            break;
        }
        evc = evc->ev_next;
    }
    if ( evc_tkn_srv == NULL )
    {
        printf ("\n Erro - fc cancelp - tkn em servico nao encontrada na cadeia de eventos");
        exit (0);
    }
    if ((evc_tkn_srv->ev_prior == NULL) && (evc_tkn_srv->ev_next == NULL))
    {
        sim[sn].evc_begin = NULL;
        sim[sn].evc_end = NULL;
        return (evc_tkn_srv);
    }
    if ((evc_tkn_srv->ev_prior == NULL) && (evc_tkn_srv->ev_next != NULL))
    {
        sim[sn].evc_begin = evc_tkn_srv->ev_next;
        evc_tkn_srv->ev_next->ev_prior = NULL;
        return (evc_tkn_srv);
    }
    if ((evc_tkn_srv->ev_prior != NULL) && (evc_tkn_srv->ev_next == NULL))
    {
        sim[sn].evc_end = evc_tkn_srv->ev_prior;
        evc_tkn_srv->ev_prior->ev_next = NULL;
        return (evc_tkn_srv);
    }
    if ((evc_tkn_srv->ev_prior != NULL) && (evc_tkn_srv->ev_next != NULL))
    {
        evc_tkn_srv->ev_prior->ev_next = evc_tkn_srv->ev_next;
        evc_tkn_srv->ev_next->ev_prior = evc_tkn_srv->ev_prior;
        return (evc_tkn_srv);
    }
    printf ("\n Erro - fc cancelp - elemento fora de opcoes de retirada");
    exit (0);
}

Init_evchain()
{
    sn = sn + 1;
    if ( sn > MAX_SIMULATIONS )
    {
        printf ("\n Erro - fc smp1 - numero de simulacoes extrapolou MAX_SIMULATIONS");
        exit (0);
    }
    sim[sn].evc_begin = NULL;
    sim[sn].evc_end = NULL;
    event = 0;
}

```

```

void schedulep(int ev, double te, int tkn, struct no *tkp)
{
    struct evchain *evc, *evc_aux;
    double st;
    st = clock + te;
    if ( tkn == 0 )
    {
        printf ("\n Erro - fc schedulep - erro fatal token = 0 ");
        exit (0);
    }
    if (te < 0.0)
    {
        printf ("\n Erro - fc schedule - tempo simulado menor que zero");
        exit (0);
    }
    evc = malloc(sizeof(struct evchain));
    if (evc == NULL)
    {
        printf ("\n Erro - fc schedulep - nao tem mais memoria para alocar para evento");
        exit (0);
    }
    evc->ev_time = st;
    evc->ev_tkn = tkn;
    evc->ev_tkn_p = tkp;
    evc->ev_type = ev;
    if ( (sim[sn].evc_begin == NULL) && (sim[sn].evc_end == NULL) )
    {
        sim[sn].evc_begin = evc;
        sim[sn].evc_end = evc;
        evc->ev_next = NULL;
        evc->ev_prior = NULL;
        return;
    }
    evc_aux = sim[sn].evc_end;

    if ( ( st > evc_aux->ev_time ) || ( st == evc_aux->ev_time ) )
    {
        evc_aux->ev_next = evc;
        evc->ev_prior = evc_aux;
        evc->ev_next = NULL;
        sim[sn].evc_end = evc;
        return;
    }
    evc_aux = sim[sn].evc_begin;

    if ( st < evc_aux->ev_time )
    {
        evc_aux->ev_prior = evc;
        evc->ev_prior = NULL;
        evc->ev_next = evc_aux;
        sim[sn].evc_begin = evc;
        return;
    }

    if ( st == evc_aux->ev_time )
    {
        evc->ev_next = evc_aux->ev_next;
        evc->ev_prior = evc_aux;
        evc_aux->ev_next->ev_prior = evc;
    }
}

```

```

        evc_aux->ev_next = evc;
        return;
    }
    evc_aux = sim[sn].evc_end;

    while (1)
    {
        if (( st > evc_aux->ev_time ) || ( st == evc_aux->ev_time ))
        {
            evc->ev_next = evc_aux->ev_next;
            evc->ev_prior = evc_aux;
            evc_aux->ev_next->ev_prior = evc;
            evc_aux->ev_next = evc;
            return;
        }
        evc_aux = evc_aux->ev_prior;
    }
    printf ("\n Erro - fc schedulep - erro no tempo nao inserido na lista de eventos");
    exit (0);
}

struct no *causep(int *ev, int *tkn)
{
    struct evchain *evc;
    struct no *tkp;
    evc = sim[sn].evc_begin;
    *tkn = evc->ev_tkn;
    *ev = evc->ev_type;
    clock = evc->ev_time;
    tkp = evc->ev_tkn_p;
    sim[sn].evc_begin = evc->ev_next;
    if (evc->ev_next == NULL)
    {
        sim[sn].evc_begin = NULL;
        sim[sn].evc_end = NULL;
    }
    else
    {
        evc->ev_next->ev_prior = NULL;
    }
    free(evc);
    return (tkp);
}

double simtime(void)
{
    return(clock);
}

void imprime_evchain()
{
    struct evchain *aux;
    aux = sim[1].evc_begin->ev_next;
    if (aux == NULL)
    {
        fprintf(fp, "Lista Vazia");
    }
    else
    {
        while(aux!=NULL)

```

```
        {
            fprintf(fp, "(Ev=%d|Time=%.2f|LSP%d|CT%d|Banda=%.0lf) -> ", aux->ev_type,
                aux->ev_time, aux->ev_tkn_p->item.NumParOD, aux->ev_tkn_p->item.CT,aux-
                >ev_tkn_p->item.Carga);
            aux = aux->ev_next;
        }
    }
}
```

## ANEXO B – FRAGMENTOS DE CÓDIGOS UTILIZADOS DA LINGUAGEM DE SIMULAÇÃO SMPL

```
/****** DISTRIBUIICOES DE PROBABILIDADE UTILIZADAS *****/
/*----- EXPONENTIAL RANDOM VARIATE GENERATOR -----*/
real expntl(real x)
{ /* 'expntl' returns a psuedo-random variate from a negative */
  /* exponential distribution with mean x. */
  return(-x*log(ranf()));
}

/*----- UNIFORM [a, b] RANDOM VARIATE GENERATOR -----*/
real uniform(real a, real b)
{ /* 'uniform' returns a psuedo-random variate from a uniform */
  /* distribution with lower bound a and upper bound b. */
  //if (a>b) then error(0,"uniform Argument Error: a > b");
  return(a+(b-a)*ranf());
}
```