

# Uma Plataforma para Engenharia de Tráfego com Qualidade de Serviço em Redes MPLS

Eduardo N. F. Zagari, Tomás A. C. Badan\*, Rodrigo C. M. Prado,

Eleri Cardozo e Maurício Magalhães

DCA - FEEC - UNICAMP - Caixa Postal 6101

Campinas, SP, Brasil, CEP 13083-970

Email: {zagari,tomas,prado,eleri,mauricio}@dca.fee.unicamp.br

## Abstract

Traffic engineering (TE) is becoming an important tool in the operation of large IP backbones. TE allows to direct network traffic into paths different from those computed by the IP routing in such a way that a better traffic balance inside the network is achieved, congestion is avoided, and network resources are optimized. Traffic engineering consists in the implantation of operation policies, aiming quality of service, such as constraint-based routing and resource reservation for certain classes of traffic. MPLS (Multiprotocol Label Switching) is an IP forwarding technology suitable for traffic engineering. In MPLS, IP packets are tagged and, based on this tag, are routed and forwarded in an appropriate way. This paper presents an implementation of MPLS for microcomputer-based networks emphasizing its traffic engineering and network management capabilities.

**Keywords:** Quality of Service, MPLS, Traffic Engineering, Constraint-based Routing, Network Management.

## Sumário

Engenharia de tráfego (ET) está se tornando uma importante ferramenta na operação de grandes *backbones* IP. ET permite direcionar o tráfego da rede para caminhos diferentes daqueles estabelecidos pelo roteamento IP convencional, de forma a distribuir melhor o tráfego na rede, evitar pontos de congestionamento e otimizar a utilização dos recursos de rede. Engenharia de tráfego consiste na implantação de políticas de operação, visando qualidade de serviço, tais como roteamento baseado em restrições e reserva de recursos para determinadas classes de tráfego. MPLS (*Multiprotocol Label Switching*) é uma tecnologia de encaminhamento IP apropriada para a engenharia de tráfego. Em MPLS, pacotes IP são marcados e, com base nesta marca, são roteados e encaminhados de forma apropriada. Este artigo apresenta uma implementação de MPLS para redes baseadas em microcomputadores, enfatizando suas capacidades de engenharia de tráfego e gerência da rede.

**Palavras-chave:** Qualidade de Serviço, MPLS, Engenharia de Tráfego, Roteamento Baseado em Restrições, Gerência de Redes.

---

\*Professor licenciado da Faculdade de Engenharia Elétrica da Universidade Federal de Goiás

# 1 Introdução

O roteamento convencional em redes envolve duas tarefas principais: a determinação de rotas ótimas e o encaminhamento dos pacotes de informação através de redes interconectadas. O protocolo de roteamento determina a rota ótima, baseado em algoritmos que são projetados para fornecer a rota de menor custo. Geralmente, tais custos são calculados como sendo o menor caminho ou o de maior largura de banda, sendo que nenhuma consideração é feita a outras métricas (como, por exemplo, congestionamento e atraso). Uma desvantagem do roteamento convencional é a sua dificuldade de suportar engenharia de tráfego.

Através de MPLS (*Multiprotocol Label Switching*), é possível influir-se no roteamento da rede adicionando-se novas rotas àquelas determinadas pelo roteamento IP padrão [4, 2, 10]. Estas rotas são denominadas Caminhos Comutados por Rótulos, ou LSPs (*Label Switched Paths*). MPLS permite aos operadores da rede um controle maior sobre o roteamento, uma característica fundamental para a engenharia de tráfego e, conseqüentemente, para a garantia de certos parâmetros de qualidade de serviço (QoS). A possibilidade de roteamento explícito, isto é, rotear fluxos de tráfego por caminhos diferentes daqueles estabelecidos pelo roteamento IP padrão, permite aos provedores de serviço utilizarem os recursos da rede (por exemplo, enlaces subutilizados) de forma mais eficiente. Desta forma, o tráfego pode ser roteado “driblando-se” enlaces interrompidos ou altamente congestionados ou mesmo gargalos da rede. Esta operação pode ser feita manualmente pelo operador da rede ou automaticamente através de, por exemplo, tecnologias de rede baseadas em políticas (*policy-based networks*).

Outra vantagem da utilização do roteamento explícito fornecido pelas redes MPLS é a diversificação dos fluxos de dados por vários caminhos dentro da rede, resultando em menor desperdício de largura de banda e possibilitando uma gerência da largura de banda mais efetiva. Além disto, os Provedores de Serviços na Internet (ISPs) podem evitar o super dimensionamento de circuitos, uma vez que a largura de banda é usada de forma mais eficiente do que nas redes IP tradicionais.

Cada fluxo ou grupo de fluxos associado a um LSP particular é uma classe de engenharia de tráfego que pode ser gerenciada através de uma política específica. A rede MPLS pode priorizar LSPs e estes podem fornecer esquemas de QoS que garantam que LSPs de alta prioridade apresentem latência e perda de pacotes mínimas.

Este artigo apresenta uma implementação de MPLS para redes baseadas em microcomputadores, enfatizando suas características para engenharia de tráfego, quais sejam, roteamento baseado em restrições (explícito) e reserva de recursos para LSPs. Duas características importantes desta implementação são: capacidade de integração com redes que suportam a Arquitetura de Serviços Diferenciados (DiffServ) e arquitetura de gerência baseada em CORBA. A seção 2 apresenta detalhes da implementação. A seção 3 apresenta a arquitetura de gerência. A seção 4 ilustra cenários possíveis de engenharia de tráfego. Finalmente, a seção 5 apresenta as conclusões do artigo.

## 2 L<sup>2</sup>SR: Uma Implementação de MPLS

Conforme apresentado em [12], desenvolvemos uma implementação de MPLS destinada a redes de baixo custo, com sistemas operacionais Linux e enlaces Ethernet. O software, denominado L<sup>2</sup>SR (*Linux Label Switch Router*), consiste da implementação do protocolo de sinalização LDP (*Label Distribution Protocol*) e extensões ao núcleo do sistema operacional Linux. A Fig. 1 ilustra a arquitetura desta implementação. Um *daemon* implementando o protocolo LDP (*ldpd*) executa no espaço do usuário e o plano de encaminhamento consiste de uma extensão do núcleo do sistema operacional Linux, visando adicionar capacidade de comutação por rótulo. Um agente de gerência disponibiliza uma interface CORBA para gerência da rede MPLS.

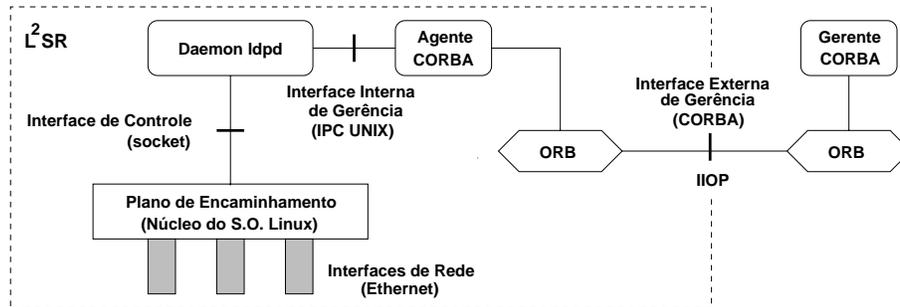


Figura 1: Arquitetura para uma nova implementação MPLS.

Duas novas extensões foram adicionadas à implementação original com o objetivo de incorporar roteamento baseado em restrições e reserva de recursos, visando qualidade de serviço. Estas novas funcionalidades foram alcançadas através da implementação do protocolo CR-LDP (*Constraint-based LDP*). Outro requisito de projeto foi prover suporte à integração com redes de Serviços Diferenciados (DiffServ), conforme especificado em [5].

O protocolo CR-LDP [1] estende a sinalização oferecida pelo LDP [9], incorporando novas funcionalidades, através da adição de novos componentes de mensagens (TLVs<sup>1</sup>). Estes novos TLVs impõem restrições durante a criação dos LSPs. Os tipos de restrições definidos pelo CR-LDP incluem o roteamento explícito de LSPs, a especificação das características de tráfego para LSPs, a fixação (pinagem) de LSPs, a preempção de LSPs através de prioridades de configuração e a definição de classes de recursos. Dentre estas extensões propostas pelo CR-LDP, escolhemos implementar as duas mais importantes sob o ponto de vista da engenharia de tráfego e da qualidade de serviço: o roteamento explícito e a especificação das características de tráfego, conforme descrito a seguir.

### 2.1 Roteamento Explícito

O roteamento explícito é uma função chave para se realizar engenharia de tráfego em redes MPLS (ET-MPLS). A fim de balancear o tráfego em redes, às vezes é necessário se antecipar ao roteamento do protocolo IP e determinar, baseado em métricas diferentes, a rota para um

<sup>1</sup>Type-Length-Value.

determinado fluxo de pacotes. Exemplo destas métricas incluem a disponibilidade de recursos, quesitos de segurança e restrições operacionais tais como SLAs (*Service Level Agreements*).

A engenharia de tráfego possibilita aos ISPs a capacidade de rotear o tráfego da rede para oferecer qualidade de serviço aos usuários, tal como conexões com banda assegurada ou com baixo atraso. ET-MPLS permite roteamento de LSPs baseado em restrições que levam em conta, por exemplo, capacidade dos enlaces e o volume de tráfego na rede. LSPs roteados com restrições (CR-LSPs) são estabelecidos introduzindo-se novos TLVs na mensagem de requisição de rótulos do protocolo LDP. Um destes é o TLV de rota explícita, que contém uma sequência de roteadores (ER-Hops), cada qual identificando um LSR (*Label Switch Router*) ou um LER (*Label Edge Router*) único ou um prefixo de rede. Assim, um CR-LSP de rota explícita atravessa todos os grupos de nós estipulados pelos ER-Hops na mesma ordem em que aparecem no TLV.

A primeira ação tomada por um LSR ao receber uma mensagem de requisição de rótulo com um TLV de rota explícita é determinar qual deve ser o próximo roteador no caminho. Esta decisão leva em consideração vários fatores, por exemplo, se a primeira entrada no TLV é um único nó ou um grupo de nós, se o LSR é topologicamente adjacente ao nó ou grupo de nós, e assim por diante. Após ter sido determinado o próximo roteador no caminho, o LSR deve continuar o processamento da mensagem de requisição de rótulos de acordo com o protocolo LDP. Terminado este processo, o LSP remove a primeira entrada no TLV e a encaminha para o próximo LSR. Quando a mensagem alcança o LER de egresso para a FEC<sup>2</sup>, este LER instala o rótulo e retorna uma mensagem de mapeamento de rótulos ao roteador acima (*upstream*) no caminho, permitindo que os demais LSRs instalem os rótulos correspondentes. Após completado este processo, um CR-LSP estará estabelecido.

É importante ressaltar o quão flexível é o esquema de roteamento explícito suportado pelo CR-LDP. O operador da rede pode declarar todos os LSRs que o LSP deve atravessar. Este esquema é denominado roteamento explícito rígido. Alternativamente, o operador pode empregar um esquema denominado roteamento explícito fraco, explicitando um subconjunto de LSRs, um conjunto de subredes<sup>3</sup> ou uma combinação destes. L<sup>2</sup>SR suporta estes dois esquemas de roteamento explícito.

## 2.2 Integração com Redes de Serviços Diferenciados

Atualmente, a gerência de tráfego é um importante requisito nas redes corporativas. Há um grande interesse na diferenciação de certas classes de tráfego, como aquelas geradas por aplicações multimídia.

A abordagem de Serviços Diferenciados (DiffServ) define três classes de serviços nas quais o tráfego gerado por uma aplicação pode ser mapeado. Estas três classes são denominadas *Expedited Forward* (EF), *Assured Forward* (AF) e *Best Effort* (BE). A classe AF é ainda subdividida em outras quatro subclasses, cada qual com três níveis de prioridades de descarte. A classe EF oferece um serviço de baixo atraso que emula um circuito de caminho comutado.

---

<sup>2</sup>*Forwarding Equivalent Class* - estipula a agregação de tráfego para um LSP na forma de um prefixo de rede ou endereço de nó.

<sup>3</sup>Dentro da subrede, o LSP é roteado *hop-by-hop*.

A classe BE corresponde ao serviço de melhor esforço. A classe AF oferece um serviço intermediário (melhor que o serviço de melhor esforço) com diferentes níveis de qualidade. Na presença de congestionamento, a rede privilegiará primeiro o tráfego EF, depois o tráfego AF e, por último, o tráfego BE.

Para se obter um comportamento comum e independente do roteador no encaminhamento de pacotes, a arquitetura DiffServ redefine a semântica do campo TOS (*Type Of Service*) no cabeçalho IP. A combinação dos seis bits de maior ordem do campo TOS define um código DiffServ (DSCP: *DiffServ Codepoint*) que identifica um *Per Hop Behaviour* (PHB). Um PHB descreve um nível de serviço particular em termos de disciplina de fila e decisões de descarte de pacotes. Em outras palavras, um dado PHB define o tratamento que um roteador deve dispensar à sua respectiva classe de serviço.

Um *Internet Draft* [5] estabelece algumas regras de interconexão entre redes MPLS e DiffServ. Estas regras determinam como se encaminhar pacotes marcados com um determinado PHB em uma rede MPLS. Dois modelos são propostos e diferem quanto ao posicionamento da informação referente ao PHB DiffServ no rótulo MPLS. No modelo l-LSP (*label-Label Switch Path*), uma certa quantidade ( $1 \leq n \leq 6$ ) dos bits usados para se encapsular o rótulo é utilizada para se codificar o PHB. Desta forma temos até  $2^n$  diferentes classes de serviço, cada qual servida por um LSP. Na Fig. 2 temos  $n = 2$  com 3 diferentes LSPs, um para cada classe de tráfego (EF-LSP, AF-LSP e BE-LSP) e com prioridade refletindo sua respectiva classe de tráfego. Neste exemplo, ao receber um datagrama com o campo DSCP marcado, o roteador de ingresso da rede MPLS determina o rótulo do pacote em função do valor do campo DSCP (2 bits do rótulo) e do endereço IP de destino (demais bits do rótulo).

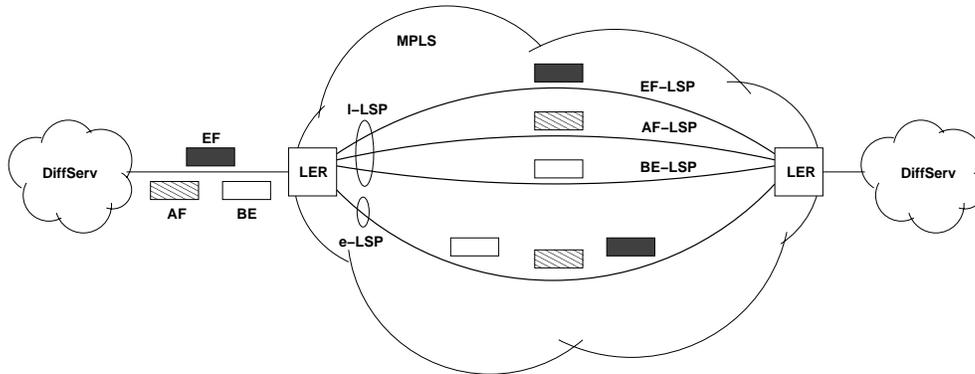


Figura 2: Integração DiffServ-MPLS através de l-LSPs e e-LSPs.

O segundo modelo, e-LSP (*exp-Label Switch Path*), codifica o PHB nos três bits reservados como experimentais (campo EXP) do cabeçalho adicional (*shim header*) introduzido no pacote. Como consequência, os e-LSPs suportam no máximo oito classes de serviços no interior de um mesmo LSP, dado que o campo EXP tem somente três bits (contra os seis bits do campo DSCP).

e-LSPs fazem a diferenciação do tráfego, utilizando um único LSP (ver Fig. 2). Neste caso, o roteador de ingresso determina o LSP baseado somente no endereço IP de destino do datagrama. A seguir, o campo DSCP é mapeado em um campo EXP adequado. Cada LSR, incluindo o roteador de ingresso (LER), encaminha o pacote baseado em seu campo EXP. Uma disciplina de filas deve assegurar a diferenciação do tráfego dentro de um mesmo LSP.

Por operar sobre redes Ethernet, L<sup>2</sup>SR utiliza *shim header* para a codificação do rótulo. Portanto, o campo EXP é sempre transportado em cada pacote IP e, por esta razão, adotou-se o modelo e-LSP para a integração MPLS-DiffServ.

## 2.3 Provisão de Qualidade de Serviço

Para prover qualidade de serviço e viabilizar uma sofisticada engenharia de tráfego, L<sup>2</sup>SR permite: (i) estabelecer características de tráfego para um dado CR-LSP; (ii) reservar recursos para CR-LSPs, visando honrar suas respectivas características de tráfego; e (iii) policiar o tráfego em CR-LSPs. A forma como QoS é implementada no L<sup>2</sup>SR é descrita a seguir.

### Especificação das Características de Tráfego

A especificação do protocolo CR-LDP estipula um TLV para codificação dos parâmetros de tráfego. Os parâmetros que descrevem as características de tráfego em relação a cada LSP são cinco:

1. PDR (*Peak Data Rate*) - taxa (bits/s) máxima gerada pela fonte de tráfego;
2. PBS (*Peak Burst Size*) - tamanho máximo de rajada (bits) gerada pela fonte de tráfego;
3. CDR (*Committed Data Rate*) - taxa assegurada de transferência;
4. CBS (*Committed Burst Size*) - tamanho assegurado de rajada;
5. EBS (*Excess Burst Size*) - excesso assegurado para rajada.

Os parâmetros PDR e PBS definem a taxa máxima na qual o tráfego pode ser enviado ao CR-LSP. Os parâmetros CDR e CBS definem a taxa que o domínio MPLS se compromete a disponibilizar ao CR-LSP. O último parâmetro (EBS), por sua vez, pode ser usado com o propósito de condicionamento de tráfego ou para se medir o quanto o tráfego enviado em um CR-LSP pode exceder sua taxa assegurada.

Estes parâmetros de tráfego vêm do modelo proposto para redes de Serviços Integrados (IntServ) [11]. Em relação à integração DiffServ-MPLS, os parâmetros de tráfego definidos para CR-LSPs são inadequados. A razão disto é a falta de informação para se mapear estes parâmetros em classes de serviços DiffServ. Por exemplo, a partir destes parâmetros é impossível se inferir a taxa média para cada classe de serviço DiffServ. Esta informação é importante para que o roteador de ingresso possa realizar o policiamento do tráfego.

A especificação referente a integração DiffServ-MPLS não determina como se empregar os parâmetros de tráfego do CR-LDP durante o estabelecimento de e-LSPs. Os autores visualizam pelo menos três abordagens para solucionar este problema:

1. Todos os LSRs policiam somente a taxa média total. Um mecanismo de policiamento de tráfego externo ao domínio MPLS assegura que o montante de tráfego para cada classe de serviço não será excedido.
2. O LER de ingresso é informado (através da gerência de rede, por exemplo) sobre as taxas médias para cada classe de serviço, quando o e-LSP estiver sendo estabelecido. O LER de ingresso, então, policia o tráfego para cada classe de serviço. Os roteadores de núcleo podem ou não policiar a taxa média total de acordo com os parâmetros de tráfego propagados durante a configuração do LSP.

3. Uma convenção pode ser estabelecida. Por exemplo, a taxa média de pico (PDR) informa o montante total de tráfego permitido para o e-LSP. O parâmetro CDR informa o montante de tráfego permitido para o tráfego EF. Com esta convenção, os tráfegos EF e AF podem ser policiados em todos os LSRs.

L<sup>2</sup>SR adota o segundo esquema com a opção de desabilitar o policiamento de tráfego nos roteadores de núcleo dado que, uma vez que o roteador de ingresso realiza o policiamento de tráfego para cada classe de serviço, o policiamento da largura de banda total no núcleo torna-se desnecessário.

### Reserva de Recursos e Policiamento de Tráfego

Desde sua versão 2.0, o núcleo do sistema operacional Linux incorporou várias facilidades para suportar controle de tráfego. Os elementos de controle de tráfego mais importantes são as disciplinas de filas, as classes e os filtros de pacotes [14]. Através da combinação destes elementos, uma ampla gama de comportamentos de policiamento e diferenciação de tráfego pode ser alcançada, incluindo o comportamento DiffServ.

A Fig. 3 ilustra um uso típico de filas, filtros e classes. Uma classe define um determinado tratamento dado aos pacotes que pertençam a ela. Para se selecionar a qual classe pertence um determinado pacote, são instalados filtros no caminho de processamento do pacote. Um filtro pode direcionar um pacote a uma classe ou fila, alterar seu cabeçalho, descartar um pacote ou, simplesmente, deixar o pacote passar, alcançando, eventualmente, um outro filtro.

A classe mais simples possível seria aquela que sustenta uma fila onde os pacotes são armazenados antes de serem transmitidos. Pode-se ter classes mais complexas que, recursivamente, possuam classes internas. As classes também podem definir disciplinas de filas a fim de estabelecer como as filas (ou classes interiores) são escalonadas. Esquemas como FIFO (*First-In, First-Out*), prioridade e CBQ (*Class-based Queuing*) são exemplos de disciplinas de filas.

No exemplo ilustrado na Fig. 3, duas classes são apresentadas. A primeira classe, destinada a tráfego de alta prioridade disponibiliza uma fila com filtro *Token Bucket* (TBF) capaz de limitar a taxa do tráfego para esta classe a um nível pré-estabelecido. A segunda classe, de menor prioridade, emprega uma disciplina FIFO sem nenhum policiamento de tráfego. As filas são processadas de acordo com a prioridade da classe, ou seja, pacotes selecionados para a classe de maior prioridade são enviados antes daqueles encaminhados através da classe de menor prioridade.

A Fig. 4 ilustra de forma simplificada a utilização de filas, filtros e classes para assegurar QoS aos CR-LSPs estabelecidos com parâmetros de tráfego no L<sup>2</sup>SR. Para cada CR-LSP, uma classe denominada “Controlada” é instalada no LER de ingresso (lado esquerdo da Fig. 4). A classe possui três filas com disciplinas *Token Bucket*, cada qual com uma prioridade (maior para a fila EF e menor para a fila BE), limitando o volume de tráfego, segundo os parâmetros estabelecidos para sua respectiva classe de tráfego. Um filtro inspecionando o campo EXP direciona o tráfego para a fila correspondente.

Ao ingressar na rede MPLS, um filtro determina, através do rótulo, se o pacote pertence a um CR-LSP ou a um LSP estabelecido sem reserva de recursos (isto é, estabelecido sem o TLV contendo parâmetros de tráfego). No primeiro caso, o filtro direciona o pacote para a classe “Controlada” associada ao LSP. No segundo caso, o filtro direciona o pacote para uma classe

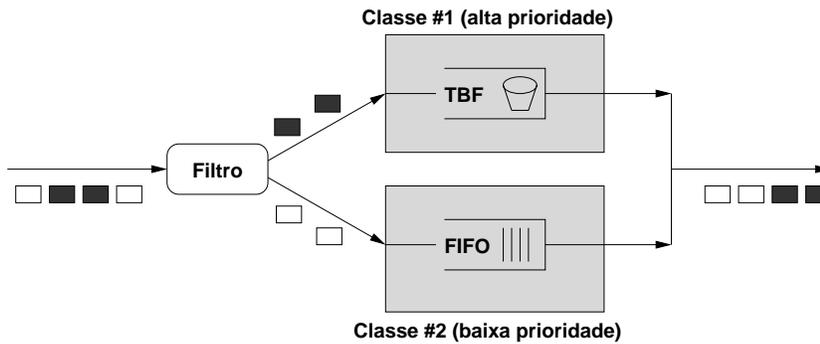


Figura 3: Classes, disciplina de filas e filtros.

denominada “Melhor Esforço”. Esta classe possui uma única fila FIFO, sendo compartilhada por todo o tráfego de melhor esforço.

Nos demais LSRs ao longo do CR-LSP, um filtro inspecionando o rótulo determina se o pacote trafega por um CR-LSP ou por um LSP convencional (lado direito da Fig. 4). No primeiro caso, o pacote é direcionado para uma classe denominada “Agregada”, compartilhada por todos os CR-LSPs, tendo este LSR como núcleo (isto é, os CR-LSPs não ingressando neste LSR). Esta classe possui três filas FIFO escalonadas por prioridade. No segundo caso, o pacote é direcionado para a classe “Melhor Esforço”.

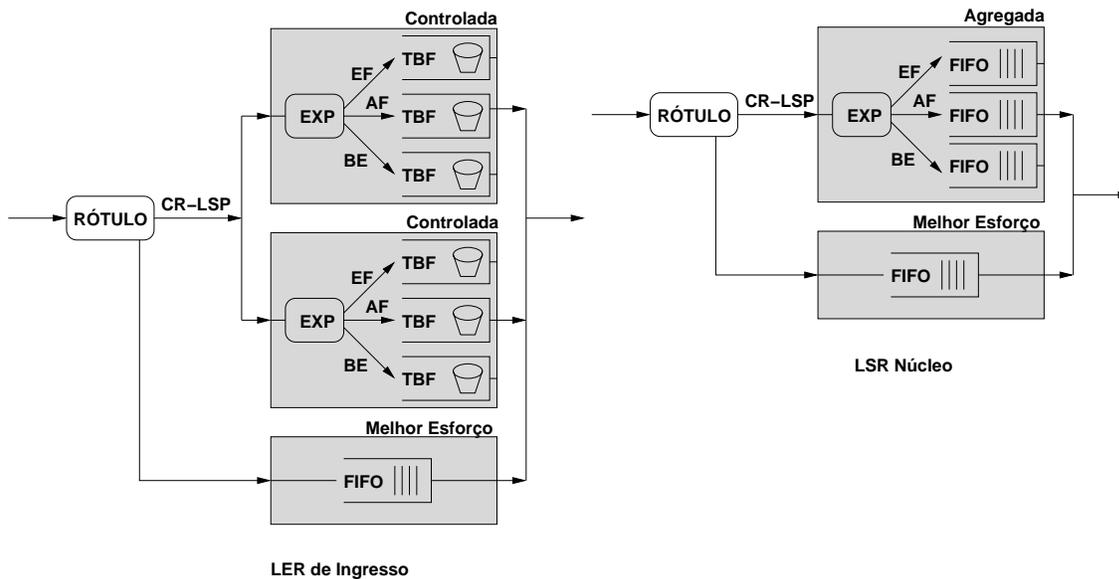


Figura 4: Mecanismo de QoS empregado no L<sup>2</sup>SR para CR-LSPs: no LER de ingresso (esquerda) e nos demais LSRs (direita).

L<sup>2</sup>SR emprega o utilitário  $\tau_c$  (*traffic controller*) para a instalação e desinstalação de classes, disciplinas de filas e filtros no núcleo. Este utilitário é executado no espaço do usuário e processa arquivos de *scripts* especificando os elementos a serem instalados, seus parâmetros, interconexões e assim por diante. Quando o protocolo de sinalização CR-LDP processa a mensagem de mapeamento de rótulo, o mesmo monta um *script* de entrada para o  $\tau_c$  de acordo com

os parâmetros passados na mensagem de requisição de rótulos correspondente (no caso de roteadores de núcleo) ou de acordo com a ação de gerência que gerou a mensagem de requisição de rótulos (no caso de roteadores de ingresso).

### Mapeamento de PHB para EXP

Existem duas formas para se mapear o PHB DiffServ para valores do campo EXP do MPLS: empregando-se um mapeamento estático (que seja consistente em todos os LSRs dentro do domínio) ou propagando-se um mapeamento via mensagens do protocolo de sinalização. L<sup>2</sup>SR emprega o mapeamento estático, por questão de simplicidade.

Não é estabelecido nenhum esquema de mapeamento nas especificações. Desta forma, adotamos o mapeamento entre os campos DSCP e EXP de acordo com a Tab. 1. Note que todas as quatro classes AF são agregadas em uma classe única no e-LSP.

DSCP	EXP	Comentário
101110	111	classe Expedited Forward (EF)
001xx0	100	classe Assured Forward (AF) 1
010xx0	100	classe Assured Forward (AF) 2
011xx0	100	classe Assured Forward (AF) 3
100xx0	100	classe Assured Forward (AF) 4
000000	000	classe Default (best effort)

Tabela 1: Mapeamento PHB – EXP.

## 3 Gerência da Rede MPLS

Conforme discutido em [12], decidimos projetar e implementar uma arquitetura de gerência baseada em CORBA para a implementação MPLS aqui descrita (L<sup>2</sup>SR).

Basicamente, a arquitetura proposta consiste de duas entidades: os Agentes CORBA (AC) e o Gerentes CORBA (GC). Os ACs executam em cada LSR da rede MPLS, enquanto os GCs executam em estações de trabalho gerentes. Desta maneira, o paradigma Gerente-Agente encontrado nos protocolos SNMP (*Simple Network Management Protocol*) e CMIP (*Common Management Information Protocol*) é preservado. No entanto, nesta arquitetura, a informação de gerência é transportada via IIOP (*Internet Inter-ORB Protocol*).

A Fig. 1 ilustra a arquitetura de gerência implementada. Como os ACs executam nos LSRs, os mesmos são objetos distribuídos (*servants*) que processam as requisições oriundas dos GCs (objetos clientes). Foram desenvolvidos duas versões do Gerente CORBA: uma escrita em C++, que executa como um processo Unix, e outra escrita em Java, que executa como um *applet* em um navegador *Web*.

As especificações das MIBs (*Management Information Bases*) para o LDP e para o LSR são apresentadas em [7] e [3], respectivamente. Entretanto, toda a estrutura da informação de gerência contida nestas especificações é descrita empregando-se ASN.1 (*Abstract Syntax Notation v.1*). Em um ambiente de gerência CORBA, tais estruturas devem ser descritas em IDL (*Interface Definition Language*).

Com base na especificação JIDM (*Joint Inter-Domain Management*) [13], mapeamos os modelos de informação ASN.1 para especificações CORBA IDL, preservando a estrutura da MIB.

## O Agente CORBA

Os Agentes CORBA (ACs) executam como *daemons* Linux e implementam os métodos de acesso definidos na interface IDL. A comunicação entre o *daemon* AC e o *daemon* LDP (`ldpd`) é realizada através de filas de mensagens providas pelo mecanismo de IPC (*Inter-Process Communication*) do Unix. O *daemon* `ldpd` possui uma *thread* separada para processamento destas mensagens IPC.

O agente CORBA foi escrito em C++ e emprega MICO<sup>4</sup> [8], uma implementação CORBA de domínio público. O AC mantém um arquivo de configuração XML (*eXtensible Markup Language*), armazenando informações tais como portas, limites, lista de interfaces que podem suportar controle de tráfego, largura de banda reservada para cada classe por interface e uma tabela de mapeamento do PHB para EXP. `ldpd` processa este arquivo durante sua inicialização. Um analisador XML, `libxml` [6], foi empregado para processar o arquivo de configuração.

## O Gerente CORBA

Foram implementadas duas versões de Gerente CORBA (GC) para a gerência de LSRs. A versão mais simples foi escrita em C++ e executa de forma autônoma em uma *shell* Unix. Esta versão oferece uma interface baseada em texto, através de menus, a partir da qual o usuário pode selecionar operações e fornecer os parâmetros correspondentes. Esta versão de GC emprega um ORB MICO para a comunicação com o Agente CORBA do LSR que se deseja realizar a operação. Este gerente permite um conjunto de operações para a gerência do LSR e do protocolo LDP. As ações de gerência do LSR incluem:

- inicialização e término do *daemon* LDP;
- gerência de interfaces: permite habilitar e desabilitar as interfaces do LSR para processamento MPLS e ajustar a largura de banda alocada para as classes de serviço nas interfaces com MPLS habilitado.

As ações de gerência do LDP incluem:

- informações operacionais: estado operacional, versão do protocolo, tamanho máximo da PDU (*Packet Data Unit*), modo de distribuição de rótulos e modo de retenção de rótulos;
- configuração: portas TCP/UDP, ajuste dos tempos de envio de mensagens para manutenção de sessões LDP e dos tempos de validade das sessões após o recebimento da última mensagem;
- visualização da lista de adjacências, sessões e LSPs mantidos pelo *daemon* LDP;
- criação e destruição de LSPs, com e sem roteamento explícito, especificando ou não parâmetros de tráfego.

---

<sup>4</sup><http://www.mico.org>

Uma versão mais elaborada de GC foi implementada como um *applet* Java e executa em navegadores Web. Este gerente CORBA orientado à Web fornece um conjunto de interfaces gráficas ao usuário as quais suportam as mesmas funções que o GC baseado em texto. A Fig. 6 apresenta uma das interfaces oferecidas por este GC.



Figura 5: Interface de gerência baseada em texto.



Figura 6: Interface de gerência baseada em Web.

## 4 Estudos de Caso

O suporte à engenharia de tráfego propiciado pelo  $L^2SR$  será ilustrado através de três experimentos. O primeiro visa mostrar uma das ações de engenharia de tráfego, a qual consiste em desviar um tráfego de um enlace congestionado para uma rota mais favorável. Isto se dá através do estabelecimento de um CR-LSP com roteamento explícito rígido e sem reserva de recursos. Um segundo experimento visa solucionar o mesmo problema, através de priorização de determinado fluxo de pacotes em detrimento de outro. Para tanto, estabelece-se um LSP com reserva de recursos para diferentes classes de tráfego. Por fim, um terceiro experimento ilustra a integração de dois ou mais domínios DiffServ por meio de uma rede MPLS. A topologia da rede utilizada durante os testes é apresentada na Fig. 7.

### 4.1 Experimento 1: Engenharia de Tráfego

O objetivo deste experimento é mostrar como uma ação de engenharia de tráfego pode melhorar a vazão dos pacotes através da rede, quando os fluxos envolvidos passam pelo mesmo enlace, conforme ditados pelo encaminhamento IP.

A fim de provocar um congestionamento nas redes 10.2.5.0/24 e 10.2.8.0/24, ilustradas na Fig. 7, foi estabelecido um fluxo de dados UDP entre as máquinas “mococa” e “aruana”, via programa MGEN<sup>5</sup>. O fluxo criado a partir da máquina “mococa” teve início no instante 0s (zero segundo) e término em 30s, com uma taxa constante de 8960 pacotes por segundo, cada qual com 1024 bytes, gerando uma taxa média de transmissão de aproximadamente 70Mbps.

<sup>5</sup><http://manimac.itd.navy.mil/MGEN/>

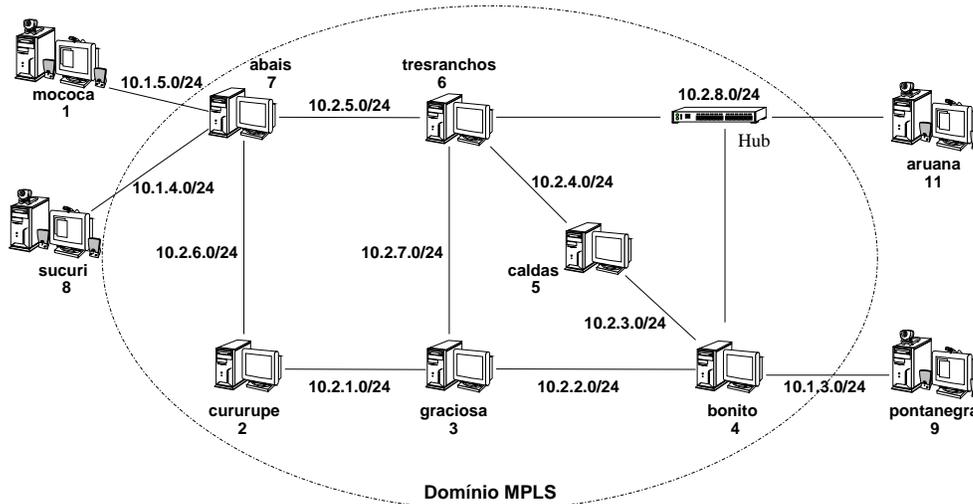


Figura 7: Topologia da rede utilizada nos estudos de caso.

Um segundo fluxo com estas mesmas características foi criado entre as máquinas “sucuri” e “pontanegra” a partir do instante 7s até o instante 22s. Este fluxo também tenta utilizar grande parte da largura de banda dos enlaces que atravessa (todos os enlaces são de 100Mbps). Através do roteamento IP, considerando que todas as rotas tenham o mesmo custo, é fácil perceber que este fluxo irá passar também pelas redes 10.2.5.0/24 e 10.2.8.0/24. As Figs. 8 e 9 mostram o resultado.

No roteador de egresso, os fluxos têm suas taxas médias de transmissão sensivelmente reduzidas, devido à concorrência entre eles em enlaces do *backbone* da rede, que se tornam gargalos. O aumento na taxa média de transmissão (aproximadamente 90Mbps) e o prolongamento além do instante 30s do fluxo da Fig. 8 se devem ao fato de que, face a um congestionamento, as interfaces de saída dos roteadores possuem a capacidade de armazenamento temporário (*bufferização*), descarregando-os assim que voltem a ter acesso ao meio.

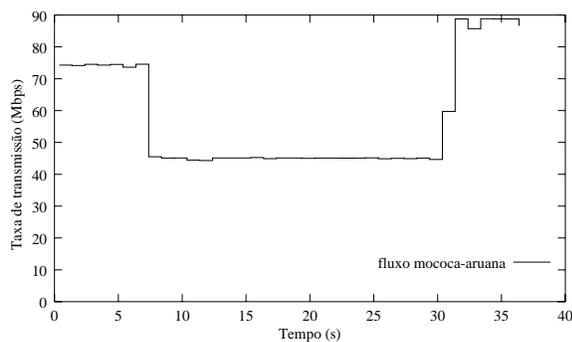


Figura 8: Fluxo na chegada da máquina “aruana” sem CR-LSP.

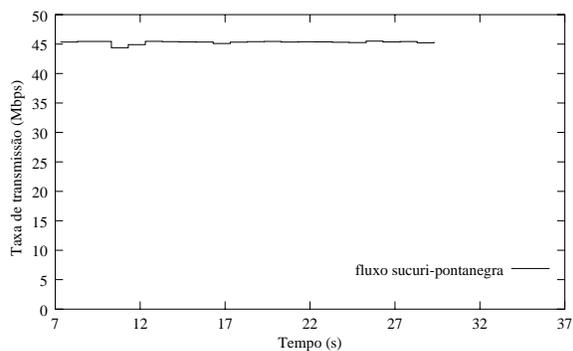


Figura 9: Fluxo na chegada da máquina “pontanegra” sem CR-LSP.

Uma forma de se contornar tal situação é realizar uma ação de engenharia de tráfego, através do emprego da funcionalidade de roteamento explícito propiciado pelo protocolo CR-LDP. Conforme pode-se notar na Fig. 7, é possível distribuir os fluxos na rede, com um pequeno aumento

no custo de uma das rotas, de forma a evitar o congestionamento. Para tanto, criamos um CR-LSP explícito entre o roteador de ingresso “abais” e o roteador de egresso “bonito”, passando pelos roteadores de núcleo “cururupe” e “graciosa”, nesta ordem. Desta forma, todo o fluxo com destino à FEC “pontanegra”, injetado pela máquina “sucuri”, é desviado da porção de rede com pequena largura de banda residual para o CR-LSP. O resultado do teste com fluxos de mesmas características do teste anterior e com a criação do CR-LSP é exibido nas Figs. 10 e 11.

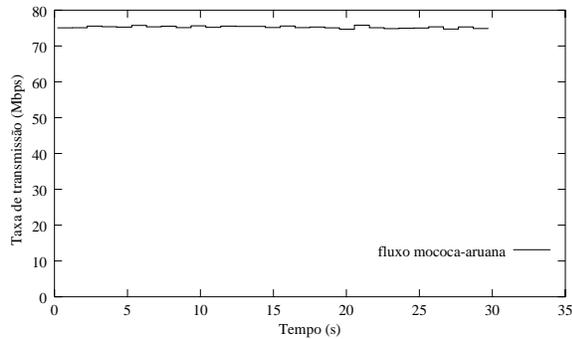


Figura 10: Fluxo na chegada da máquina “aruana” com CR-LSP estabelecido com rota explícita.

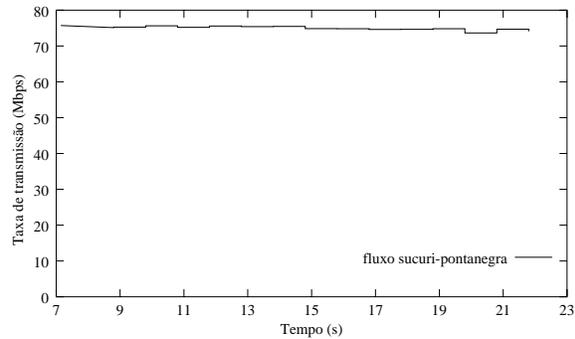


Figura 11: Fluxo na chegada da máquina “pontanegra” sem CR-LSP estabelecido com rota explícita.

## 4.2 Experimento 2: Reserva de Recursos

Uma segunda alternativa para solucionar o problema do congestionamento provocado pelos fluxos no caso ilustrado pelas Figs. 8 e 9 é através da criação de um CR-LSP com reserva de largura de banda. Por exemplo, suponha que o fluxo entre as máquinas “sucuri” e “pontanegra” deva ser garantido, enquanto aquele entre “mococa” e “aruana” possa ser tratado apenas como um fluxo de melhor esforço. É possível, então, criar-se um CR-LSP entre os roteadores “abais” e “bonito”, passando pelo LSR “tresranchos”, com reserva de largura de banda para um fluxo EF para contemplar o primeiro tráfego. As Figs. 12 e 13 mostram o resultado do teste, que permite honrar a precedência entre tráfegos de distintas prioridades.

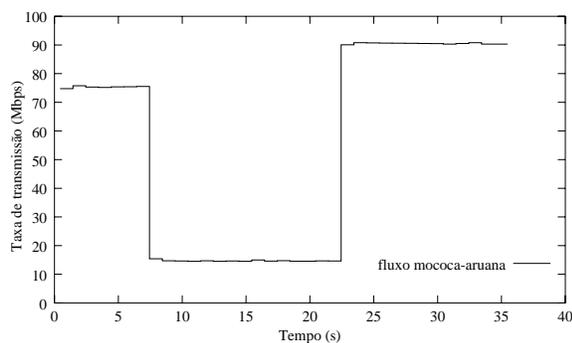


Figura 12: Fluxo na chegada da máquina “aruana” com CR-LSP estabelecido com reserva de largura de banda.

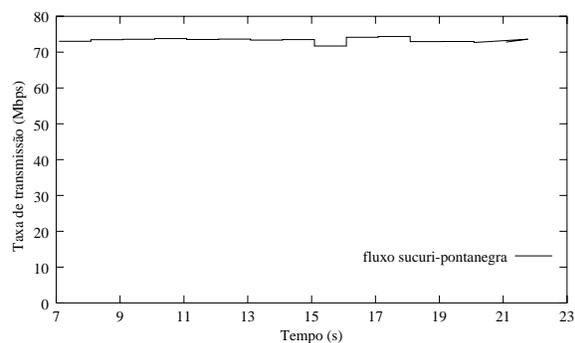


Figura 13: Fluxo na chegada da máquina “pontanegra” sem CR-LSP estabelecido com reserva de largura de banda.

### 4.3 Experimento 3: Integração DiffServ-MPLS

Além do roteamento explícito e da provisão de recursos de banda,  $L^2SR$  permite a Integração DiffServ-MPLS e o policiamento dos tráfegos de classes de serviço distintas dentro do domínio MPLS. Como exemplo, considerando a mesma rede de teste, estabelecemos um CR-LSP com TLV de parâmetro de tráfego entre os roteadores “abais” e “bonito”, para interconectar as máquinas “mococa” e “pontanegra”, representando dois domínios DiffServ.

A partir da informação sinalizada pelo protocolo CR-LDP, cada LSR pode realizar sua reserva de largura de banda para o LSP. No caso do LSR de ingresso, informamos, via interface de gerência, como a largura de banda total reservada deveria ser dividida entre as três classes de tráfego. Neste caso teste, 50Mbps foram reservados para o tráfego EF do domínio DiffServ, 30Mbps para o tráfego AF e os 20Mbps restantes do LSP para o tráfego BE. Desta forma, o LSR de ingresso pode realizar o controle de admissão e o posicionamento de tráfego por classe de serviço. Note que o TLV de parâmetro de tráfego não contém campos para propagar as taxas de transmissão por classes aos LSRs de núcleo. Assim, estes LSRs somente conhecem a largura de banda total que será compartilhada pelos fluxos dentro do LSP. Embora os LSRs de núcleo possam policiar a largura de banda total,  $L^2SR$  foi configurado para não realizar o policiamento no núcleo da rede.

Foram gerados três fluxos partindo da máquina “mococa”, cada um deles de 100Mbps. O primeiro, F1, iniciado no instante 0s e terminado no instante 20s. O segundo, F2, começou em 3s e terminou em 17s, enquanto, o fluxo F3 foi iniciado no instante 7s e terminado no instante 14s. Os resultados, quando não são definidos os valores do campo DSCP dos pacotes em cada fluxo, podem ser observados na Fig. 14. Como esperado, cada LSR concede o mesmo tratamento aos três fluxos, uma vez que não é possível diferenciá-los.

Com o campo DSCP devidamente marcado, os tráfegos dentro da rede MPLS são mapeados para distintos valores do campo EXP, o que diferencia as classes de serviço dentro da rede. Os resultados podem ser observados na Fig. 15. Neste caso, embora cada fluxo tente consumir toda a largura de banda disponível nas interfaces de saída dos roteadores do domínio MPLS, o controle de tráfego limita a largura de banda de cada um deles dentro dos valores especificados durante a configuração do e-LSP, honrando a diferenciação de serviços do domínio DiffServ.

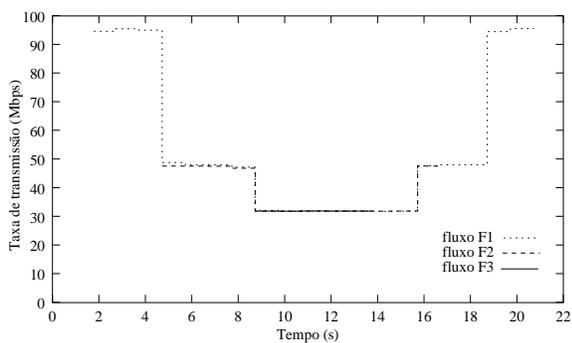


Figura 14: Fluxos de saída no roteador de egresso (“bonito”) sem a marcação DSCP.

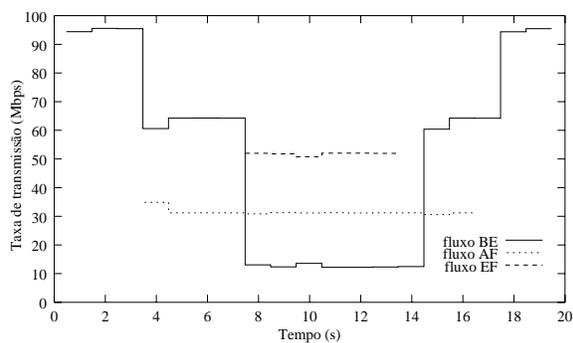


Figura 15: Fluxos de saída no roteador de egresso (“bonito”) com a marcação DSCP.

## 5 Conclusões

Engenharia de tráfego é um importante instrumento de operação e otimização das redes IP. MPLS é uma tecnologia de encaminhamento que suporta engenharia de tráfego através de roteamento baseado em restrições (por exemplo, restrições de qualidade de serviço). Este artigo apresentou uma implementação MPLS (L<sup>2</sup>SR) para redes constituídas de roteadores Linux e enlaces Ethernet que suporta roteamento baseado em restrições, além de capacidade de integração com redes DiffServ.

L<sup>2</sup>SR permite várias ações de engenharia de tráfego, algumas ilustradas na seção 4, tais como: estabelecer um LSP através de uma rota explícita; reservar recursos por classe de tráfego por LSP e combinar as duas estratégias anteriores.

As ações acima são iniciadas pelo operador da rede através de interfaces de gerência. Os autores estão investigando formas automatizadas de engenharia de tráfego onde a intervenção do operador é minimizada. São exemplos de ações automatizadas:

- estabelecer um LSP quando a qualidade de serviço associada a determinado fluxo roteado *hop-by-hop* degrada;
- rerrotear um LSP quando a qualidade de serviço oferecida ao seu tráfego degrada;
- ajustar a quantidade de recursos atribuída a um LSP de acordo com seu volume real de tráfego;
- utilizar a capacidade excedente da rede para o estabelecimento sob demanda de LSPs com duração limitada, por exemplo, para uma transmissão de vídeo.

Tais ações necessitam de uma supervisão e controle sofisticados para a rede. Estamos considerando técnicas de supervisão e controle baseadas em agentes móveis (na linha de redes ativas), inteligência computacional (por exemplo, programação genética) e sistemas dirigidos por eventos.

## Referências

- [1] B. Jamoussi et al. Constraint-Based LSP Setup using LDP. Internet Draft, MPLS Working Group, November 2001.
- [2] U. Black. *Multi-Protocol Label Switching*. Prentice Hall, December 2000.
- [3] C. Srinivasan, A. Viswanathan, T. Nadeau. Multiprotocol Label Switching (MPLS) Label Switch Router (LSR) Management Information Base. Technical report, IETF - Network Working Group, January 2002.
- [4] E. Rosen et al. Multiprotocol Label Switching Architecture. RFC-3031, January 2001.
- [5] F. Le Faucheur et al. MPLS Support of Differentiated Services. Internet Draft, Network Working Group, February 2001.
- [6] Gnome Project. The XML C library for Gnome.

- [7] J. Cucchiara, H. Sjostrand, J. Luciani. Definitions of Managed Objects for the Multi-protocol Label Switching, Label Distribution Protocol (LDP). Technical report, IETF - Network Working Group, August 2001.
- [8] K. Romer, A. Puder, F. Pilhofer. MICO: An Open Source CORBA Implementation. Alpha-Craze.com, 2000.
- [9] L. Andersson et al. Label Distribution Protocol Specification. RFC-3036, January 2001.
- [10] M. Magalhães, E. Cardozo. Comutação IP por Rótulos através de MPLS (Minicurso). In *Anais do 19º Simpósio Brasileiro de Redes de Computadores*, 2001.
- [11] S. Shenker, J. Wroclawski. General Characterization Parameters for Integrated Service Network Elements. RFC-2215, September 1997.
- [12] T. Badan, R. Prado, E. Zagari, E. Cardozo, M. Magalhães. Uma Implementação do MPLS para Redes Linux. In *Anais do 19º Simpósio Brasileiro de Redes de Computadores*, 2001.
- [13] The Open Group. Inter-Domain Management: Specification Translation and Interaction Translation.
- [14] W. Almesberger, J. Salim, A. Kuznetsov. Differentiated Services on Linux. <http://citeseer.nj.nec.com/almesberger99differentiated.html>, 1999.