UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
ESCOLA POLITÉCNICA
DEPARTAMENTO DE ELETRÔNICA E DE COMPUTAÇÃO

Implementação e Análise de Desempenho de um Ambiente de Testes de Qualidade de Serviço

Autor:

Marco Dias Dutra Bicudo

Orientador:

Prof. Otto Carlos Muniz Bandeira Duarte, Dr.Ing.

Examinadores:

Célio Vinicius Neves de Albuquerque, Ph.D.

Luís Henrique Maciel Kosmalski Costa, Dr.

Marcelo Gonçalves Rubinstein, D.Sc.

Marcelo Luiz Drumond Lanza, M.Sc.

DEL
Março de 2004

Agradecimentos

À minha família, principalmente meus pais por todo o amor, orientação e apoio ao longo da minha vida. Mesmo distante fisicamente em alguns momentos, sempre estiveram presentes e atuantes.

A toda a equipe do GTA, em particular ao professor Otto, meu orientador, pela orientação e conselhos. Ao professor Luis Henrique, pela amizade, pelas revisões de texto e dicas. Ao Doc, Pedro, e Bernardo pela boa vontade em emprestar seus PCs, e pela paciência nas semanas em que tinha que recompilar o *kernel*, tirar as medidas, e reiniciar o computador umas dezenas de vezes. Aos professores do Departamento de Eletrônica, principalmente aos professores Lanza, amigo presente de toda sexta feira, Rezende, Marcelo Campos, Joarez e Barúqui.

Aos professores Otto, Luís Henrique, Rubi, Lanza e Célio por aceitarem fazer parte da banca.

Resumo do Projeto Final apresentado ao DEL/UFRJ como parte dos requisitos necessários para a obtenção do grau de Engenheiro Eletrônico e de Computação.

Implementação e Análise de Desempenho de um Ambiente de Testes de Qualidade de Serviço

Marco Dias Dutra Bicudo

Março de 2004

Orientador: Otto Carlos Muniz Bandeira Duarte

Departamento: Engenharia Eletrônica e de Computação

O protocolo IP (*Internet Protocol*), desenvolvido para aplicações tolerantes ao atraso, satisfez, na época em que foi idealizado, os requisitos de aplicações onde a entrega do pacote seguindo o modelo do melhor esforço era suficiente. Porém, a popularização cada vez maior de aplicações multimídias, sensíveis a esses parâmetros, requer a diferenciação de serviços para garantir a Qualidade de Serviço (QoS). Como o protocolo IP não é capaz de fornecer garantia de QoS, outros protocolos e arquiteturas foram propostas para esta tarefa. Como por exemplo as arquiteturas de Serviços Diferenciados (*DiffServ*), de Serviços Integrados (*IntServ*) e o protocolo MPLS (*Multiprotocol Label Switching*).

Nesse trabalho é implementado um ambiente de testes com o objetivo de analisar o desempenho de uma rede com suporte a diferenciação de serviços através do MPLS, encaminhando fluxos prioritários para recursos reservados da rede. O MPLS também foi utilizado para realizar engenharia de tráfego tornando mais eficiente o uso da rede. O desempenho da rede foi monitorado com os protocolos de roteamento RIP e OSPF com a métrica banda passante para comparação do desempenho da rede com MPLS. Assim, é possível determinar o ganho na eficiencia com a utilização do MPLS.

Palavras Chave

MPLS

Internet

Engenharia de Tráfego

Qualidade de Serviço

Backbone

Lista de Acrônimos

ACK: *Acknowledge*;

AF: *Assured Forwarding*;

ARPA: *Advanced Research Projects Agency*;

ATM: *Asynchronous Transfer Mode*;

BA: *Behavior Aggregate*;

BE: *Best Effort*;

CBR: *Constant Bit Rate*;

CR-LDP: *Constraint Routing - LDP*;

DS: *DiffServ*;

DSCP: *DiffServ Code Point*;

EF: *Expedited Forwarding*;

EXP: *Experimental*;

FCFS: *First-come Firs-served*;

FEC: *Forwarding Equivalence Class*;

FTN: *FEC to Next Hop Label Forwarding Entry Map*;

HTTP: *HyperText Transfer Protocol*;

IETF: *Internet Engineering Task Force*;

ILM: *Incoming Label Map*;

IP: *Internet Protocol*;

ISO: *International Organization for Standardization*;

ITU-T: *International Telecommunication Union - Telecommunication*; *Standardization Section*;

LDP:	*Label Distribution Protocol*;
LSP:	*Label Switched Path*;
LSR:	*Label Switching Router*;
MF:	*Multi-Field*;
MPLS:	*Multiprotocol Label Switching*;
NHLFE:	*Next Hop Label Forwarding Entry*;
OSPF:	*Open Shortest Path First*;
PHB:	*Per Hop Behavior*;
QoS:	*Quality of Service*;
RIP:	*Routing Information Protocol*;
RSVP:	*Resource Reservation Protocol*;
RSVP-TE:	*RSVP - Traffic Engineering*;
TCP:	*Transfer Control Protocol*;
TOS:	*Type os Service*;
TTL:	*Time-To-Live*;
UDP:	*User Datagram Protocol*;
VBR:	*Variable Bit-Rate*;
WFQ:	*Weighted Fair Queue*;

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo I

Introdução

I.1 Motivação

As redes de computadores foram desenvolvidas e idealizadas para interconectar computadores em diferentes locais, com o objetivo de compartilhar dados de servidores para máquinas clientes. Devido ao desenvolvimento de novas aplicações com diferentes requisitos, surgiram novas exigências.

As novas aplicações que necessitam de recursos da rede são cada vez mais comuns, e a popularização da Internet cresce a cada dia. Transmissão de TV na Internet, conferências e jogos *on-line* são exemplos de aplicações que necessitam de requisitos específicos para um bom desempenho. Em um futuro próximo, o uso dessas aplicações multimídias pode ser mais freqüente e fazer parte do dia-a-dia. Para isso ser possível, é necessário que as tecnologias que interconectam as diferentes redes sejam capazes de diferenciar fluxos de dados mais prioritários dos menos prioritários. Além disso, essas tecnologias devem não só diferenciar, mas também tratar os fluxos conforme for especificado no contrato de serviço do cliente, mantendo o fluxo com os valores determinados de atraso, banda passante, taxa de erro, e qualquer outro parâmetro que se considere importante para uma determinada aplicação.

Com esse objetivo, foram propostas diferentes arquiteturas de Qualidade de Serviço (QoS) que tentam oferecer alguma garantia de desempenho às aplicações, e em conjunto também foram desenvolvidas técnicas de engenharia de tráfego, que tentam distribuir o tráfego na rede de forma a melhorar a utilização dos recursos disponíveis.

I.2 A História e a Arquitetura da Internet

O aparecimento da Internet se deve basicamente a duas necessidades que tinham que ser atendidas: a necessidade de compartilhar recursos entre computadores distantes, e a necessidade do governo americano em ter uma rede que fosse robusta frente a desastres naturais, guerras e cataclismas. A interconexão de redes sem administração central é robusta à destruição de algumas de suas partes, por isso a ARPAnet, rede que originou a Internet, teve seu projeto baseado na idéia de descentralização.

A ARPAnet, foi a rede desenvolvida pelo órgão americano do Departamento de Defesa, Agência de Projetos e Pesquisas Avançadas - ARPA, que tinha o objetivo inicial de compartilhar os altos custos das linhas telefônicas de longa distância. Para tornar possível esse compartilhamento, utilizou-se a comutação de pacotes desenvolvida por Baran em 1961, e que ficou esquecida por alguns anos. A ARPAnet foi pioneira no desenvolvimento de alguns dos padrões e técnicas de telecomunicações, como o controle de fluxo, o roteamento descentralizado, comutação de pacotes em redes de longa distância, [1]. Essas tecnologias, em conjunto com algumas aplicações da época como o `telnet` e o `rlogin`, serviram de base para a construção da Internet, e são utilizados até hoje.

A idéia de compartilhar os custos de linhas telefônicas de longa distância e a característica do tráfego de dados em rajadas levou ao uso da comutação de pacotes. Com a comutação de pacotes é possível se fazer a utilização de recursos sob demanda. Assim pode haver compartilhamento de recursos entre fluxos. Por outro lado, na comutação de circuito ou em uma linha dedicada, todo o recurso é dedicado para aquele fluxo, mesmo que este não o utilize. Toda a informação que a Internet transporta é através de pacotes. Os pacotes são formados basicamente de duas partes: o cabeçalho, que contém a informação de controle necessária para o encaminhamento dos pacotes, e os dados, que são as informações a serem transferidas. Como conseqüência da utilização de pacotes para transmissão de dados, o funcionamento da Internet segue o modelo denominado armazene-e-encaminhe (*store-and-forward*), ou seja, o pacote é recebido pelo roteador, armazenado em um *buffer*, e quando chegar sua vez de ser transmitido é retirado do *buffer* e enviado. Existem vantagens e desvantagens nesse modelo de encaminhamento. A vantagem é o compartilhamento dos recursos da rede, que podem ser utilizados de maneira mais eficiente, a desvantagem é a falta de garantia de uma qualidade, pois o desempenho se torna dependente da quantidade de usuários presentes e ativos na rede. Essa falta de garantia de QoS é problemática para as aplicações multimídias, que surgiram nos últimos anos.

Como a arquitetura da Internet se baseia na descentralização do controle, diferente da rede telefônica, os usuários da Internet não poderiam contar com a rede para obter confiabilidade na entrega dos dados. Para uma aplicação que demande a entrega confiável de seus dados, por exemplo, uma aplicação de e-mail, ou http, é necessário um protocolo de transporte que garanta essa confiabilidade, como é o caso do TCP. A filosofia dos desenvolvedores da Internet é fazer com que a complexidade de controle de fluxo e a garantia de integridade fossem passadas para os usuários finais. É importante notar que isso torna a rede simples e mais escalável.

No entanto, essa descentralização é problemática para a qualidade de serviço. A Internet foi projetada para entregar os pacotes através do modelo do melhor esforço, ou seja, o pacote será encaminhado quando possível. Isso é um problema para a transmissão de tráfegos de aplicações sensíveis aos parâmetros de QoS, e com restrições de valores de atraso, banda e taxa de perda.

A solução a esse problema é difícil de ser alcançada mantendo a escalabilidade e não indo contra a filosofia descentralizada da Internet. A comutação de pacotes é também outro complicador, já que esta não disponibiliza a reserva de recurso, ao contrário da comutação de circuito. Existem considerações a se fazer quanto à maneira como os tráfegos são priorizados. Isso deve ser feito levando em conta suas necessidades de QoS e da disponibilidade de recursos da rede. Mais adiante são apresentadas as técnicas de escalonamento de filas e gerenciamento ativo de filas, bem como o controle de admissão, e engenharia de tráfego, que farão a garantia dos recursos ao usuário final.

I.3 As Aplicações Multimídias

Em geral, a maioria das aplicações multimídias precisa de interatividade, e como conseqüência disso, necessita de atraso e variação de atraso (*jitter*) limitados a faixas de valores bem definidas, e dependendo da aplicação uma banda passante mínima também pode ser requisitado. Como a Internet utiliza o modelo do melhor esforço, esta não é a rede apropriada para as aplicações multimídias sem que modificações sejam realizadas. Esses parâmetros de desempenho são denominados parâmetros de Qualidade de Serviço, e o atual grande desafio da Internet é integrar a garantia desses parâmetros a arquitetura já existente da Internet.

Um roteador que faz o encaminhamento dos pacotes seguindo o modelo do "*store-and-forward*" armazena os pacotes de todos os usuários em um *buffer* compartilhado. Isso significa que quando um usuário enviar uma rajada de pacotes, o próximo usuário terá seu pacote enfileirado após essa rajada, o que acarretará um atraso acima do esperado. Caso o *buffer* do roteador tenha sido todo ocupado pela rajada do primeiro usuário, os pacotes subseqüentes serão perdidos. Este é um comportamento natural para o que se chama de um modelo de serviço FCFS (*first-come-first-served*). Em uma rede onde esteja sendo transmitida qualquer aplicação sensível a atraso, como conferências de vídeo ou voz, a qualidade dessas aplicações é degradada caso ocorram essas rajadas.

A Internet pode oferecer qualidade de serviço aos seus usuários se o *buffer* do roteador for compartilhado de maneira justa, ou seja, cada usuário tem a sua parcela de utilização. Para que isso seja realizado, os pacotes não podem ser transmitidos de acordo com a ordem de chegada, uma nova ordem deve ser efetuada pelos algoritmos de escalonamento de filas.

I.3 As Aplicações Multimídias

Para oferecer garantia de QoS, é necessária a sinalização para os roteadores do caminho que o fluxo passa. A sinalização é responsável por determinar os recursos que aquele fluxo obterá em cada roteador do caminho. A sinalização determina os recursos de cada fluxo através de classes de serviços pré-definidas. Para o caso de novos fluxos entrantes, é feita a classificação nessas classes de serviço. Antes da aceitação e reserva do recurso, esses fluxos serão admitidos pelo roteador, este é o papel do controle de admissão.

O funcionamento desse mecanismo, que inclui a sinalização, o controle de admissão e o escalonamento de filas, necessita de uma cooperação entre os roteadores que foge à filosofia original da Internet, de comutação de pacotes e de somente prover a conectividade de seus integrantes. Devido a isso, se torna complicado a introdução de mecanismos de QoS na Internet. Essa característica da Internet faz com que os pacotes sejam tratados de acordo com o modelo do melhor esforço (*best-effort*), por isso não há reserva de recursos como acontece na rede telefônica. Para oferecer alguma garantia de QoS é necessário oferecer um tratamento diferenciado para fluxos mais prioritários, reservando recursos para estes, se for o caso. São necessários mecanismos adicionais a Internet, pois o protocolo IP não possui mecanismos de garantia de QoS ou de reserva de recursos, para a diferenciação de serviços.

Além disso, a característica descentralizada da Internet, pode acarretar um uso ineficiente da rede, como será comentado nos próximos capitulos. As causas serão diagnosticadas e soluções para esse uso ineficiente serão apontadas no decorrer do trabalho. A solução mais indicada atualmente é a engenharia de tráfego, que consiste na distribuição dos fluxos entre os enlaces ou recursos da rede de maneira que estes não se tornem ociosos, acarretando uso ineficiente. O MPLS (*Multiprotocol Label Switching*) é um protocolo que possibilita a aplicação da engenharia de tráfego em uma rede devido, principalmente, a aplicação do conceito de comutação de circuitos em uma rede de comutação de pacotes. Ambos os conceitos serão explicados adiante. O objetivo desse trabalho é a construção de um ambiente de testes, onde serão realizados testes com o protocolo MPLS tornando possível a analise de desempenho da rede com e sem QoS e engenharia de tráfego, evidenciando quais são as vantágens da utilização do MPLS em uma rede.

Capítulo II

Qualidade de Serviço

Como a Internet é uma rede não orientada a conexão, os pacotes de um mesmo fluxo podem percorrer caminhos diferentes até a chegada ao destino. Um fluxo consiste em uma seqüência de pacotes de um computador de origem até um outro de destino. As características de cada fluxo podem ser representadas através de quatro parâmetros: o atraso, a variação do atraso - *jitter*, a vazão e a confiabilidade [2]. Esses são os parâmetros de QoS, que juntos determinam como um fluxo deve ser transportado pela rede para satisfazer à aplicação.

Algumas aplicações são muito exigentes em certos parâmetros de QoS. Cada fluxo, dependendo da sua aplicação, poderá por exemplo ser sensível ao atraso, e ao mesmo tempo ser tolerante à confiabilidade dos dados. Este seria o caso de uma aplicação multimídia interativa. Mas pode acontecer o oposto, o fluxo ser sensível à confiabilidade e ser tolerante ao atraso, como é o caso da transferência de e-mails e arquivos.

A Tabela II.1 mostra que as aplicações de `e-mail, ftp, www` e `rlogin` requerem confiabilidade dos dados, e são tolerantes ao atraso. Por esses motivos o TCP é utilizado na camada de transporte, caso haja algum erro na transmissão do fluxo, retransmissão é efetuada, não importando o atraso que isso acarrete para a transferência. Já em relação ao atraso, pode-se dizer que as aplicações de www e rlogin possuem uma tolerância ao atraso menor que as de e-mail e ftp. Essa característica se deve à interatividade com o

Tabela II.1: Requisitos das aplicações aos Parâmetros de QoS.

Aplicação	Confiab.	Atraso	*Jitter*	Banda
e-mail	alta	baixa	baixa	baixa
ftp	alta	média	baixa	média
www	alta	média	baixa	média
rlogin	alta	média	média	baixa
áudio sob demanda	baixa	baixa	alta	média
vídeo sob demanda	baixa	baixa	alta	alta
telefonia	baixa	alta	alta	baixa
videoconferência	baixa	alta	alta	alta

usuário necessária para a navegação na Web e para a execução de comandos remotamente. A telefonia e a videoconferência demandam um atraso de no máximo 150 ms, isso porque a interatividade da aplicação seria comprometida, como especificado no padrão ITU-T. Para o áudio e vídeo sob demanda o atraso não é crítico, mas a variação do atraso sim. No caso da transmissão de TV pela Internet, após a escolha do canal pelo usuário a aplicação pode demorar alguns segundos até começar, mas uma vez iniciado não pode haver quebra de continuidade (*gaps*) na transmissão. A banda por sua vez será requisitada para as aplicações que necessitem de grande volume de informação em pouco tempo. Esse será o caso da maioria das aplicações de vídeo, mas a banda passante também pode ser requisitada por aplicações de áudio, www e ftp.

II.1 Sinalização

Sinalização é o processo pelo qual o usuário final requisita à rede reserva, liberação ou renegociação de recursos. A sinalização também é responsável por definir como se comportará o fluxo dos usuários. A determinação do comportamento do fluxo é realizada através dos parâmetros de qualidade de serviço. Estes parâmetros são transmitidos para

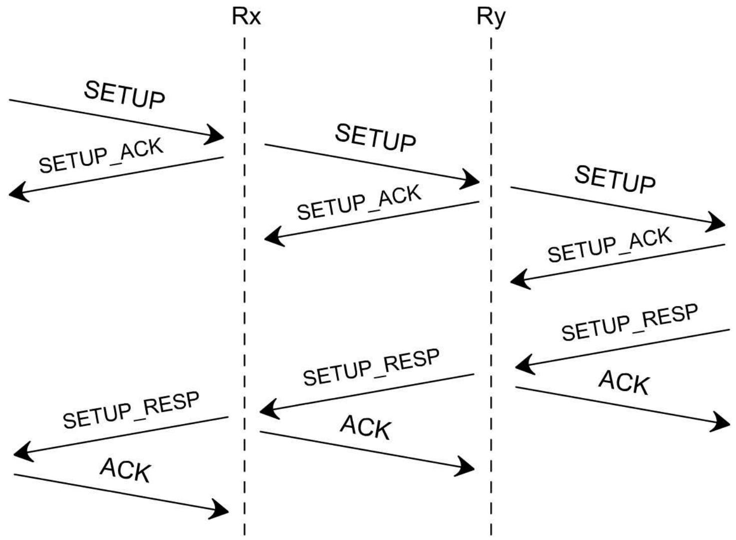

Figura II.1: Reserva de Recursos com RSVP.

todos os nós responsáveis pelo encaminhamento do fluxo, ou classe de fluxo. Com essas informações, o roteador pode: reservar *buffers* de transmissão nas interfaces de saída, reservar a banda que aquele fluxo utilizará, e definir a prioridade desse fluxo em relação aos outros.

O mecanismo básico de funcionamento da sinalização da reserva de recurso é o seguinte. A origem envia mensagens de SETUP para o primeiro roteador, caso este roteador seja capaz de atender aos recursos que estão sendo reservados é retornado um SETUP_ACK. Essa verificação dos recursos disponíveis é feita pelo Controle de Admissão de roteador em roteador até chegar ao destino, verificando se algum roteador não é capaz de atender os recursos especificados. Se a reserva chegar até o destino, este retorna um SETUP_RESPONSE para o último roteador que efetiva a reserva. Esse processo é repetido salto-a-salto até a origem do pedido da reserva, finalizando a reserva do recurso. A Figura II.1 é uma ilustração de um exemplo com dois roteadores com a seqüência de mensagens transferidas entre os roteadores e os clientes para que a reserva de recursos seja efetuada.

O IETF adotou o protocolo RSVP (*Resource Reservation Protocol*) como o padrão de sinalização de reserva de recursos, que foi definido em [3]. O RSVP foi desenvolvido e projetado sem ter pré-requisitos nem compatibilidades a cumprir do ponto de vista de sinalização, devido principalmente ao fato de a Internet não ser uma rede orientada a conexão, nem a reserva de recursos. O mecanismo de manutenção dos recursos também é adaptado às características da Internet. A dinâmica da rede e de seus enlaces fez com que o RSVP utilizasse o que se chama de estado volátil para a manutenção de seus recursos. O mecanismo consiste, basicamente, de mensagens de reservado receptor que serão enviadas periodicamente mantendo o estado do próximo roteador, este roteador fará o mesmo sucessivamente. Assim quando algum enlace da rede se quebrar, e os protocolos de roteamento se adaptarem à nova topologia, o RSVP fará uma migração da reserva para outros caminhos, já que as mensagens periódicas de RESV permitirão a reserva de recursos para esse caminho diferente.

II.2 Controle de Admissão

Como foi visto, a sinalização carrega os parâmetros que fazem a descrição do comportamento do tráfego. Mas para garantir que esse novo tráfego não prejudique a qualidade de serviço dos outros tráfegos com recursos já reservados, e que os recursos do novo tráfego estejam disponíveis para a reserva, é necessário o controle de admissão. O controle de admissão faz a decisão de aceitar, rejeitar, ou prorrogar a reserva do recurso. Essa decisão é feita baseada em um conjunto de parâmetros que faz a descrição do tráfego, e na quantidade de recursos disponíveis. A maneira como o controle de admissão vai fazer a verificação de disponibilidade do recurso depende da característica do tráfego. O controle de admissão de fluxos CBR (*Constant Bit Rate*) é simples, pois um fluxo com essas características pode ser descrito por apenas um único parâmetro. Para determinar se existem recursos suficientes para um fluxo CBR, basta saber a capacidade do enlace, e quanto deste enlace já está reservado por outros fluxos CBR já existentes.

O controle de admissão de fluxos VBR (*Variable Bit Rate*) não é simples como o controle de admissão e fluxos CBR. Isso se deve à sua característica de envio de dados em rajadas, o que torna a taxa de pico de envio diferente da taxa média. Ou seja, existem períodos de ociosidade, seguidos de momentos de intensa utilização. Porém, existe como simplificar esse modelo para que o algoritmo de controle de admissão seja factível. Estatisticamente, pode-se considerar a soma de vários fluxos VBR como sendo um CBR se a quantidade de fluxos VBR for grande o suficiente. A taxa de envio desse CBR que caracteriza a agregação de vários fluxos VBR será a soma das taxas médias dos fluxos VBR. A quantidade de fluxos VBR deve ser grande o suficiente para que a rajada de um fluxo seja compensada pela ociosidade de outro. Assim, basta usar a regra de alocação de recursos para fluxos CBR.

II.3 Condicionamento de Tráfego

A qualidade de serviço pode ser garantida através de diferentes mecanismos. Pode ser garantida através do superdimensionamento da capacidade dos enlaces da rede, através do Condicionamento de Tráfego (*Traffic Shaping*), ou através do Escalonamento de Pacotes (*Packet Scheduling*). As duas últimas são técnicas não inter-dependentes, e podem ser aplicadas ao mesmo tempo.

O superdimensionamento da rede é uma solução aparente. Essa solução pode até garantir a banda necessária às aplicações mais exigentes, mas pode não ser capaz de garantir atraso mínimo, e um *jitter* pequeno, restrições necessárias à maioria das aplicações multimídias. Essa solução não garante essas restrições porque a espera no *buffer* pode ainda existir, tornando o atraso imprevisível, e como conseqüência a variação do atraso. Além disso, o enlace é utilizado de maneira não eficiente.

O enfileiramento do tráfego no *buffer* pode ser uma solução satisfatória a princípio, pois pode diminuir o *jitter* controlando a cadência com que o tráfego chegará no receptor. Porém, para aplicações em tempo real, onde o atraso dos pacotes através da rede é crítico, essa solução não é adequada, pois atrasa a chegada de todos os pacotes. Essa técnica é muito utilizada em aplicações de vídeo e áudio sob demanda. Por exemplo, o Windows Media Player é uma aplicação que utiliza essa técnica para receber áudio e vídeo da

Internet.

A Figura II.2, mostra o comportamento do atraso de pacotes em um fluxo quando este passa por vários *buffers* de transmissão, que também estão sendo utilizados por outros fluxos. A soma dos atrasos em todos os *buffers* mostra que o atraso de pacotes não é associado somente à propagação. A espera em fila é na verdade a responsável por gerar essa variação do atraso, pois a fila é um recurso compartilhado, e portanto, seu desempenho é dependente do comportamento dos outros fluxos.

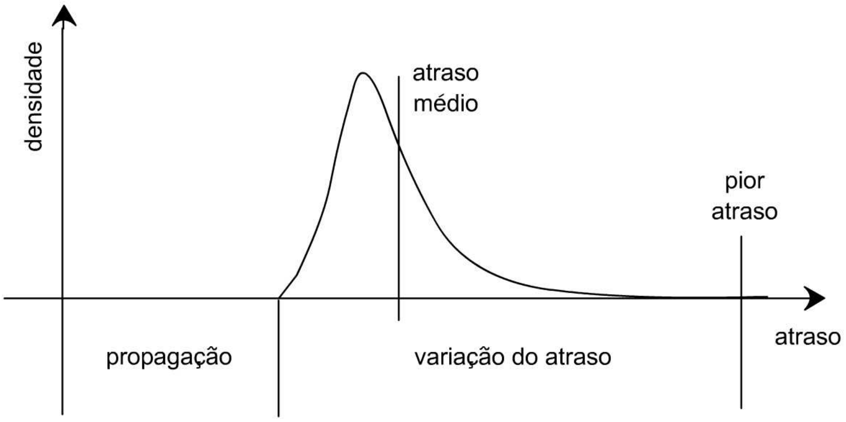

Figura II.2: Densidade da variação do atraso.

O Condicionamento de Tráfego é responsável por regular a taxa média de envio de dados. É geralmente utilizado para garantir que um usuário não exceda a vazão especificada no contrato de serviço. Após concordar em carregar um tráfego com as características especificadas, a rede deve ser capaz de monitorar o tráfego do usuário e verificar se ele está seguindo as características especificadas. Caso o usuário não esteja, a rede deve saber que ação será executada. Esse monitoramento do tráfego do usuário é conhecido como Policiamento de Tráfego (*Traffic Policing*).

O Balde Furado (*Leaky Bucket*) e o Balde de Fichas (*Token Bucket*) são algoritmos de condicionamento de tráfego utilizados para limitar a banda de usuários. O funcionamento do algoritmo é semelhante a um balde com um furo no fundo, daí o nome. Há um balde que acumula pacotes. Uma taxa constante de pacotes vazam pelo furo. Esta taxa independe do fluxo de pacotes que esteja entrando no balde. Portanto, caso o fluxo de pacotes entrante seja maior que o fluxo de pacotes que vaza, haverá acumulação de pacotes no balde. A vazão do furo equivale a taxa de transmissão máxima, a capacidade de armazenamento de pacotes do balde equivale ao tamanho do *buffer* de transmissão, e a vazão de pacotes que entra equivale a taxa de transmissão do emissor que será controlada. Então, se os pacotes que entram tiver a vazão muito maior que a vazão do furo o *buffer* irá transbordar, obrigando o roteador a descartar os pacotes subseqüentes. A função desse algoritmo é diminuir os picos de intensa atividade seguidos de momentos de silêncio, característicos de alguns fluxos, além de limitar a banda utilizada pelo usuário. Isso será feito através do enfileiramento nos *buffers* dos pacotes excedentes. Esse condicionamento do tráfego torna a utilização dos enlaces mais eficiente, pois os momentos de ociosidade serão menos freqüentes [4].

O algoritmo do Balde de Fichas tem o funcionamento semelhante ao funcionamento do Balde Furado. A vantagem dele é a flexibilidade de configuração, pois possibilita que o tráfego controlado possa transmitir pequenas rajadas, desde que não ultrapasse um limite estabelecido, o que não é permitido no Balde Furado. Esse algoritmo consiste em um balde cheio de fichas, onde cada ficha corresponde ao direito de enviar um pacote. Cada pacote enviado, retira uma ficha do balde. As fichas caem no balde a uma taxa especificada pelo administrador, que vai depender da característica do fluxo e do serviço contratado [4]. Então a principal diferença com relação ao Balde Furado é a possibilidade de ter uma rajada com uma taxa máxima maior. Porém se essa rajada é monitorada pelo algoritmo Balde de Fichas, a duração e a vazão máximas são especificadas pelo administrador, assim não há possibilidade de um usuário mal comportado prejudique o desempenho da rede.

II.4 Arquiteturas IntServ e DiffServ

Em meados da década de 90, o IETF investiu no desenvolvimento de uma arquitetura que tornasse possível a transmissão multimídia na Internet. Desta necessidade, surgiu o IntServ, a abreviação de *Integrated Services* [5] . O principal objetivo a cumprir era atender às especificidades de cada aplicação, respeitando as necessidades de QoS de cada fluxo. No IntServ existe a necessidade de sinalizar para cada os roteadores reservarem os recursos necessários a cada fluxo. Para fazer essa sinalização o IETF desenvolveu o RSVP, mencionado na Seção II.1.

Como o IntServ guarda informação sobre cada fluxo e seus respectivos requisitos, a quantidade de estados em cada roteador é proporcional a quantidade de fluxos que passam no roteador. O custo computacional para a manutenção dos estados torna inviável tecnicamente a implementação dessa arquitetura na Internet. Concluiu-se que essa arquitetura não é escalável para aplicar em uma rede dessa magnitude.

As falhas do IntServ para a Internet impulsionaram a criação de outra arquitetura que permitisse a transmissão multimídia. Para ter uma arquitetura de garantia de QoS que fosse possível aplicar na Internet era necessário fazer com que essa nova arquitetura fosse mais escalável. Isso significa criar uma arquitetura onde um roteador fosse capaz de garantir a QoS de tantos fluxos quantos existirem nos principais roteadores da Internet. Essa nova abordagem levou a uma solução baseada na agregação de fluxos.

O *Differentiated Services*, ou DiffServ, é um serviço oferecido por um conjunto de roteadores que formam um mesmo domínio administrativo, o domínio DiffServ ou DS [6]. O administrador da rede define as classes de serviços, as agregações de fluxo, e as regras de encaminhamento associadas a essas classes. A identificação dessas agregações é feita através do campo TOS (*Type of Service*) do cabeçalho IP, mais especificamente no subcampo DSCP (*DiffServ Code Point*), definido em [7], o qual é responsável por fazer o mapeamento das agregações em um Comportamento por Nó (*Per Hop Behavior* - PHB). Os roteadores de ingresso do domínio DS devem ser capazes de identificar um fluxo individualmente, definindo a agregação de fluxos a qual ele pertence. Essa classificação será feita com base em informações de um ou mais campos dos cabeçalhos IP, TCP e UDP dos pacotes. Devido a essa característica esses classificadores são chamados de classificadores Multi-campo (*Multifield* - MF). Dentro do Domínio DS os pacotes são classificados em classes de serviço de acordo com o código do campo DSCP, esses classificadores são chamados de Comportamento Agregado (*Behavior Aggregate* - BA). Nessa arquitetura, quando um usuário assinante do serviço enviar pacotes, seus pacotes podem carregar um TOS que indique uma agregação mais prioritária dependendo do serviço que foi especificado na contratação.

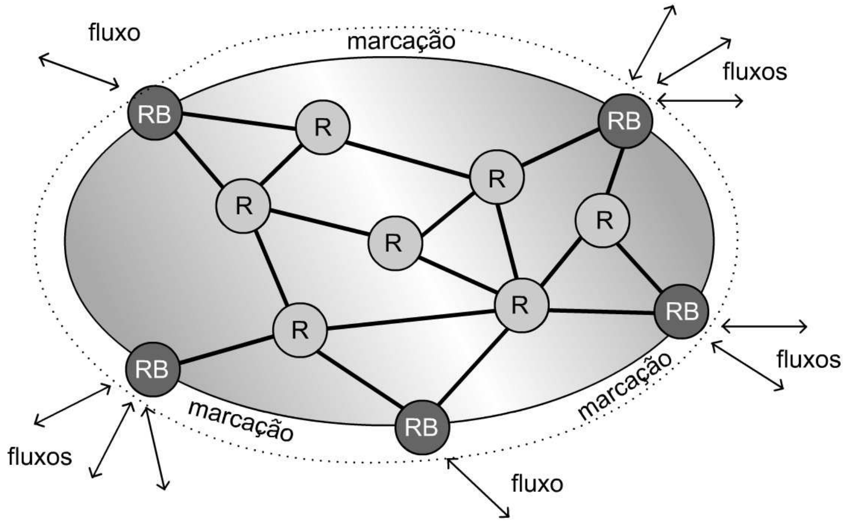

Figura II.3: Domínio DiffServ.

O tráfego associado a uma classe pode ser condicionado para manter conformidade com o serviço contratado. Isso pode ser feito utilizando-se os algoritmos Balde Furado ou Balde de Fichas, e tem o objetivo de diferenciar as agregações de classes de serviço. É importante ressaltar que esse esquema de controle do usuário não requer configurações avançadas, reserva de recursos e negociação para cada fluxo, como no IntServ. Isso faz do DiffServ uma arquitetura mais simples de implementar, menos custosa computacionalmente e, portanto, escalável, sendo mais apropriada à Internet.

Analisemos o caso da telefonia na Internet para mostrar as diferenças entre a qualidade de serviço baseada em fluxo e a baseada em agregações de fluxos. Na QoS baseada em fluxo, cada transmissão de voz reserva seus recursos, e faz os roteadores terem um estado para cada conexão. Já na baseada por agregação, todos os fluxos telefônicos pertencem a uma mesma classe, que têm os parâmetros de QoS específicos, necessários a um tráfego de voz. A um fluxo de transferência de arquivos não é possível utilizar os recursos alocados aos fluxos de telefonia. Porém as ligações não têm recursos reservados individualmente, assim uma ligação pode interferir em outras, ou seja, caso o número de usuários passar de certo limite podem ocorrer degradações na qualidade das ligações. Mas por outro lado a manutenção de estados é reduzida, tornando possível a aplicação na Internet.

A escolha das classes de serviços que existem em um domínio DS depende somente da administração da rede. A escolha de quais são os parâmetros de QoS alocados para cada classe, qual a granularidade das classes, e qual é o mapeamento dos fluxos nas diferentes classes não segue um padrão. Essa falta de padronização é um problema para alguns domínios DS que podem ser constituídos de mais de um domínio administrativo. Para solucionar essa deficiência, o IETF definiu algumas classes de serviço padrão para que as classes pudessem ter equivalência em outras redes que não fossem a rede pela qual o fluxo entrou no domínio DS. Elas são o EF (*Expedicted Forwarding*) [8] e o AS (*Assured Forwarding*) [9].

O princípio básico do EF é simples. Existem dois tipos de serviço, a classe expressa EF, que será composta dos fluxos de aplicações críticas em relação ao atraso, como as aplicações em tempo real, e a classe regular BE (*Best Effort*), que seriam os fluxos de www, transferência de arquivo, e-mail etc.

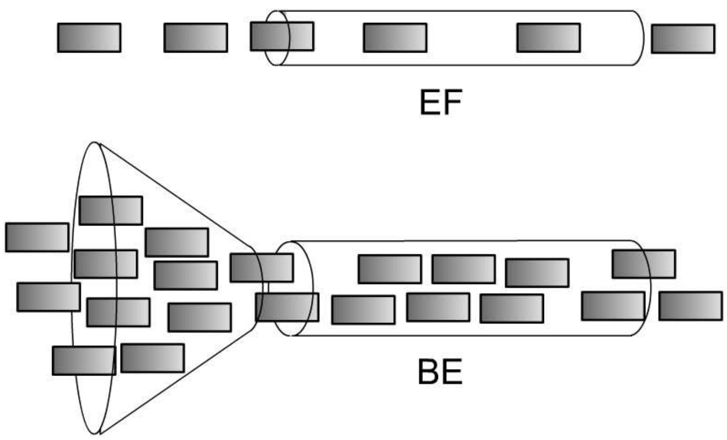

Figura II.4: Filas EF e BE.

O condicionamento de tráfego da classe de serviço EF pode ser implementado de várias maneiras. A fila de saída do roteador pode ser dividida em duas, uma é destinada aos fluxos EF, e a outra aos fluxos BE. Uma maneira de tratar os fluxos EF de maneira prioritária é aplicando um escalonamento de filas de simples prioridade, ou seja, o fluxo

EF tem prioridade sobre o fluxo BE. Desta maneira o fluxo EF não é atrasado pela espera dos pacotes a serem transmitidos. Em contrapartida, existe a possibilidade do fluxo EF monopolizar a utilização dos recursos já que enquanto existirem pacotes na fila EF, pacotes da fila BE não são transmitidos. A outra solução, seria o uso do WFQ (*Weighted Fair Queueing*), que faz uma atribuição ponderada da capacidade de transmissão do enlace de saída de acordo com as filas EF e BE. Para garantir a QoS dos fluxos EF nessa atribuição de recursos ponderada, é reservado à classe EF uma quantidade a mais de recursos que ela precisa. Desta forma a classe EF não satura seus recursos, e conseqüentemente consegue garantir QoS dos fluxos. Por exemplo, caso os fluxos EF resultem em 10% da capacidade do enlace, 20% deste pode ser reservado para a fila EF e o restante seria associado a fila BE. Fazendo assim, é oferecido aos fluxos EF o dobro dos recursos necessários para alcançar um baixo atraso. Assim, para esses fluxos a rede esta sempre descongestionada, mesmo quando isso não seja verdade em outras classes de serviço.

A classe de serviço AF é um esquema mais elaborado de gerenciar as diferentes classes de serviço. Dentro da classe AF existem outras quatro classes de prioridade, cada uma com seus parâmetros de QoS, e suas reservas de recursos. Seu modelo de serviço, diferente do EF que fornecia uma garantia estrita, fornece uma expectativa de serviço obtida por um determinado tráfego quando existem momentos de congestionamento. A proposta do modelo de serviço da classe AF define que a entrega do pacote é sempre realizada com alta probabilidade, desde que o tráfego não exceda a taxa contratada. Caso o fluxo exceda a taxa contratada a entrega de seus pacotes não é mais garantida com a mesma alta probabilidade.

II.5 MPLS

Na arquitetura DiffServ, os pacotes são marcados na borda do domínio DS, através do campo DSCP. Dentro do domínio, o tratamento dos pacotes é feito de acordo com o campo DSCP.

A tecnologia MPLS (*Multiprotocol Label Switching*) é uma tentativa de solucionar o problema do excesso de processamento necessário para o roteamento de pacotes. A comutação dos pacotes foi a solução encontrada. Para fazer a comutação é inserido no datagrama IP um rótulo antes do cabeçalho IP que fará a indexação da tabela de encaminhamento MPLS. Um mesmo rótulo pode ser inserido em diferentes fluxos, fazendo a agregação de fluxos. Nas redes MPLS, a qualidade de serviço é oferecida de maneira não muito diferente. Isso porque o rótulo MPLS tem o mesmo comportamento do campo DSCP, definir o comportamento da agregação de fluxo. Como em um domínio MPLS o cabeçalho IP não é mais consultado para o encaminhamento, o campo TOS , que está contido no cabeçalho IP, também não será consultado para definir a precedência do pacote. Entretanto, o rótulo MPLS tem um campo EXP (experimental) que foi destinado para representar a precedência do pacote, tendo a mesma função que o campo DSCP do protocolo IP.

Outra maneira de oferecer qualidade de serviço em redes MPLS é através do uso de "túneis" MPLS, os LSPs (*Label Switched Path*), para cada classe de serviço. A conseqüência é a divisão do enlace em vários outros enlaces virtuais, um para cada classe. Cada túnel desse pode ter diferentes topologias e utilizar recursos específicos para cada classe de serviço associada. As diferentes topologias dos túneis dão uma flexibilidade e versatilidade administrativa que possibilita um uso mais eficiente da rede, devido a engenharia de tráfego. Além disso, os túneis de diferentes topologias, que passam por diferentes enlaces e nós, podem direcionar um fluxo mais prioritário para um enlace menos congestionado, ou até mesmo reservado para essa classe de serviço [10]. Ao mesmo tempo, a capacidade de poder definir todos os nós da origem ao destino possibilita a Engenharia de Tráfego. Tanto a Engenharia de Tráfego quanto o MPLS serão mais extensamente abordados nos próximos capítulos.

Capítulo III

Multiprotocol Label Switching - MPLS

III.1 Visão Geral

Os roteadores em uma rede não-orientada a conexão fazem o encaminhamento dos pacotes analisando o cabeçalho IP. Cada roteador toma uma decisão individualmente sobre qual deve ser o próximo salto do pacote baseado na análise do cabeçalho feita pelo roteamento IP. Cada roteador verifica o endereço destino do pacote IP e decide qual o próximo roteador a ser encaminhado.

A função de encaminhamento pode considerar mais parâmetros do que o endereço destino.de pacotes. A tecnologia MPLS (*Multi-protocol Label Switching*) permite encaminhar pacotes de uma forma eficiente e, ao mesmo tempo, considerar outros parâmetros além do endereço destino. Esta tecnologia permite classificar os fluxos de pacotes em classes (*Forwarding Equivalence Class* - FEC) e encaminhar os pacotes segundo sua classe. O MPLS é uma solução que começou como propostas proprietárias de empresas pesquisando uma solução de como fazer o roteamento, ou encaminhamento, dos pacotes de maneira mais eficiente [4]. As propostas iniciais visavam apenas diminuir o tempo de processamento no encaminhamento dos pacotes substituindo a função de roteamento pela função de comutação de pacotes. O MPLS começou a ser padronizado pelo IETF após uma pressão de empresas que estavam desenvolvendo suas próprias pesquisas na área [11].

A substituição do roteamento pela comutação possui vantagens. O processo de roteamento pode ser definido como a procura em uma tabela pelo endereço de destino para descobrir para onde enviá-lo. Por sua vez, a comutação é definida como o processo que usa um rótulo para indexar uma tabela de comutação que determinará o destino do pacote. Por essa característica a comutação é dita como mais eficiente que o roteamento pois requer menos processamento. A busca na tabela necessária ao mecanismo de roteamento IP é complexa porque existe a necessidade de se obter o encaminhamento para o próximo roteador que tem como endereço o maior prefixo de rede. Esta característica é denominada de "*Best Prefix Matching*". Na comutação por rótulos a busca se efetua com um único acesso à memória tendo como entrada o rótulo e como saída a interface relativa ao próximo salto do encaminhamento. Para que isto seja possível é necessário acrescentar um rótulo ao pacote que é denominado de cabeçalho MPLS. Este cabeçalho MPLS, ou rótulo MPLS (*MPLS label*), envelopa o pacote IP e, portanto, se refere a uma camada abaixo da camada IP. Assim, define-se um domínio MPLS como um domínio habilitado a comutar rótulos MPLS. O pacote IP recebe um rótulo na entrada do domínio, no roteador de entrada, envelopando o pacote IP. Este rótulo só é retirado ao sair do domínio MPLS pelo roteador de saída. Todos os roteadores do domínio MPLS são denominados *Label Switch Router* (LSR). Dentro do domínio MPLS os pacotes são comutados ao invés de serem roteados. Assim, um LSR recebendo um pacote com rótulo consulta uma tabela de rótulos que tem como saída as informações sobre o próximo salto, e o valor do próximo rótulo, como nas rede ATM (*Asynchronos Transfer Mode*). Assim, o cabeçalho IP só é analisado (função de roteamento) na entrada e na saída do domínio MPLS e no interior do domínio ocorre a comutação de rótulos. Esse encaminhamento tem algumas vantagens sobre o método tradicional de encaminhamento de datagramas IP:

- o encaminhamento pode ser feito por comutadores, implementados em hardware tornando o processo mais rápido;

- a atribuição de rótulos pode levar em conta outras informações para determinar a classe (FEC) do pacote sem onerar demais o processamento, pois isto será realizado uma única vez;

- a atribuição de classes permite forçar pacotes a seguirem caminhos específicos, possibilitando encaminhamento por políticas ou simplesmente engenharia de tráfego;

III.2 O Rótulo

O rótulo MPLS é um identificador com significado local somente, isso significa que a informação carregada por ele tem validade somente no enlace que o pacote se encontra. Isso se deve porque o rótulo é modificado a cada roteador por qual o pacote passa. Essa troca é feita através do NHLFE (*Next Hop Label Forwarding Entry*), que é o conteúdo da entrada das tabelas FTN (*FEC-to-NHLFE*) e ILM (*Incoming Label Map*). O conjunto de NHLFEs e suas associações mútuas constituem a própria tabela de comutação dos LSRs.

O rótulo MPLS, definido em [12], é constituído de 32 bits, que são divididos em 20 primeiros bits para a identificação da FEC, os 3 bits seguintes são os do campo *EXP*, mais 1 bit que indica o fim da pilha de rótulos, será explicado adiante, e os 8 restantes indicam o TTL (*Time To Live*) do pacote, que devem ser explicitados no rótulo MPLS, já que o TTL do cabeçalho IP não será consultado e decrementado.

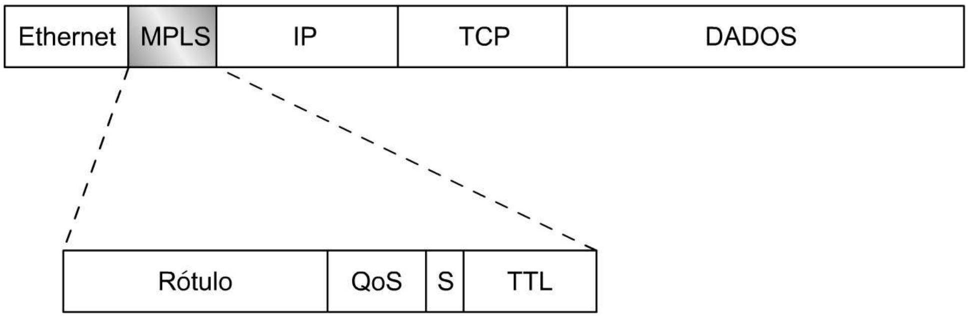

Figura III.1: Rótulo MPLS.

Os três bits de *EXP* representam a prioridade do pacote. O bit de fim de pilha de rótulos existe para que se possa fazer o que se chama de empilhamento de rótulos, ou seja, quando um pacote é recebido por um LSR, este pode conter mais de um rótulo. A função do bit S de fim de pilha é indicar qual é o último rótulo presente no pacote, e está relacionado com as operações que podem ser efetuadas com os rótulos.

Ao receber um pacote o roteador opta por uma das três possibilidades de tratamento de um rótulo: "*pop*", "*push*"e "*swap*". O comando *pop*, determina que o roteador retire o rótulo, e após isso analise o pacote sem aquele rótulo, tendo o pacote outro rótulo ou não. O comando *push*, insere um rótulo no pacote, geralmente esse comando é dado na entrada do domínio MPLS. O comando *swap*, troca o rótulo existente por outro, esse tipo de ação será geralmente encontrada no interior do domínio MPLS, onde a passagem do pacote entre roteadores acarreta em trocas de rótulos a cada enlace.

O comando *push* pode ser dado várias vezes antes que seja dado um *pop*, isso fará com que a pilha de rótulos tenha mais de um nível. Desta maneira é possível, que dentro do domínio MPLS, exista um túnel que agregue FECs com um mesmo destino, ou pelo menos com destinos próximos, reduzindo o número de entradas nas tabelas MPLS em uma rede que contém muitos túneis.

III.3 FTN e ILM

Quando o LSR recebe um pacote com o rótulo MPLS, a tabela ILM (*Incoming Label Map*) é acessada para busca de qual deve ser o tratamento tanto do pacote quanto do rótulo. O rótulo é extraído do cabeçalho MPLS e indexará a ILM em busca da entrada referente a esse rótulo. A ILM pode ter escopo local a uma interface somente, ou pode ter escopo global a todo o LSR, depende somente da configuração e das necessidades do domínio.

A entrada da ILM indica o comando (*pop, push* e *swap*) que será executado no rótulo. A entrada da tabela ILM vai poder estar associada a uma entrada da tabela FTN (*FEC-to-NHLFE*), que determina qual o próximo salto, interface de saída, e novo rótulo do quadro, se este for o caso.

III.4 O Domínio MPLS

O domínio MPLS é o conjunto de LSR controlados por um mesmo administrador, que pode estar oferecendo serviço de *backbone*, ou de interligação de redes.

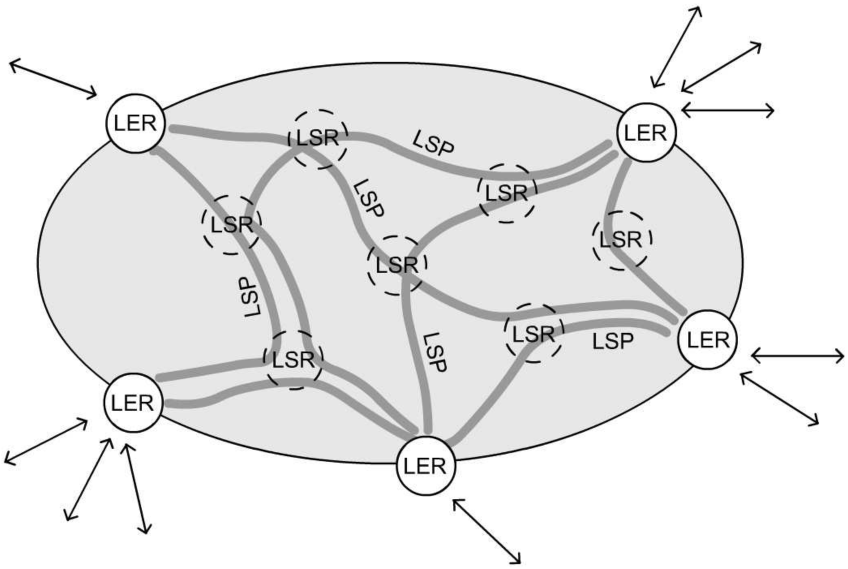

Figura III.2: Domínio MPLS e LSPs.

Em um domínio MPLS, as FECs se diferenciam de um circuito virtual em uma rede de comutação de circuitos, como a rede ATM, pela agregação. Apesar de uma FEC poder ser associada a somente um fluxo, ou seja, um fluxo ter seu próprio conjunto de rótulos no domínio, isso não é geralmente usado, já que uma das vantagens do MPLS é a agregação de vários fluxos em apenas uma FEC, podendo assim reduzir a quantidade de estados nos roteadores internos do domínio MPLS. A definição de fluxo é bastante flexível nesse caso, dependerá somente de como o LSR da borda do domínio determina o que será um fluxo. É regra geral que o MPLS deve ser usado para agregar fluxos provenientes de uma mesma subrede, ou não só de uma mesma subrede, mas que compartilham uma mesma classe de serviço. Essa agregação de múltiplos fluxos vai ser encaminhada através da rede de acordo com os rótulos, e constituirão uma mesma FEC. Essa FEC não determina somente por onde esses fluxos são encaminhados, mas também determina a que classe de serviço pertencem esses fluxos.

Nos métodos tradicionais de circuito virtual, como nas redes telefônicas ou como nas redes ATM, a agregação não é possível, já que a agregação na origem não adiciona informação para que os fluxos sejam identificados e demultiplexados na saída da rede. O MPLS, por outro lado, não enfrenta esse problema, pois os fluxos estão identificados pelo protocolo de camada 3, que na maioria das vezes é o IP.

III.5 *Label Switched Path* - LSP

O LSP é o conjunto de rótulos reservados nos roteadores ao longo do domínio MPLS que definirão qual caminho será percorrido, e pelo qual o roteador identificará a que FEC pertence o pacote. A diferença conceitual entre FEC e LSP é difícil de ser definida, FEC representa o comportamento do pacote quando esse chega ao roteador, o LSP é a combinação de todas as FECs dos roteadores.

Através dos LSPs o administrador do domínio poderá fazer a diferenciação de serviço entre seus clientes. A determinação da classe de serviço pode ser feita por outra maneira além daquela que utiliza FECs com reservas de recursos feitas em cada roteador, ou seja, de maneira semelhante ao *DiffServ*. Essa diferenciação de serviço e priorização de fluxos não será feita através de reserva de recursos nos roteadores, ela será feita através do encaminhamento de fluxos para outros enlaces da rede. Esses enlaces podem ser exclusivos para fluxos prioritários, ou podem ser enlaces que são sub-utilizados com tráfego de controle e gerenciamento da rede, e conseqüentemente com excedente de recursos para ser usado.

A determinação de qual LSP seguirá um tráfego é feita na entrada do domínio, a decisão pode ser baseada em um conjunto de informações presentes no pacote, como cabeçalho do protocolo de rede IP, ou pode até levar em conta a interface em que o roteador recebeu o pacote. No caso de utilizar informações do cabeçalho IP, o endereço de destino pode ser usado para priorizar o tráfego que vai para uma máquina específica, como por exemplo, um servidor HTTP. De maneira semelhante, o IP de origem pode

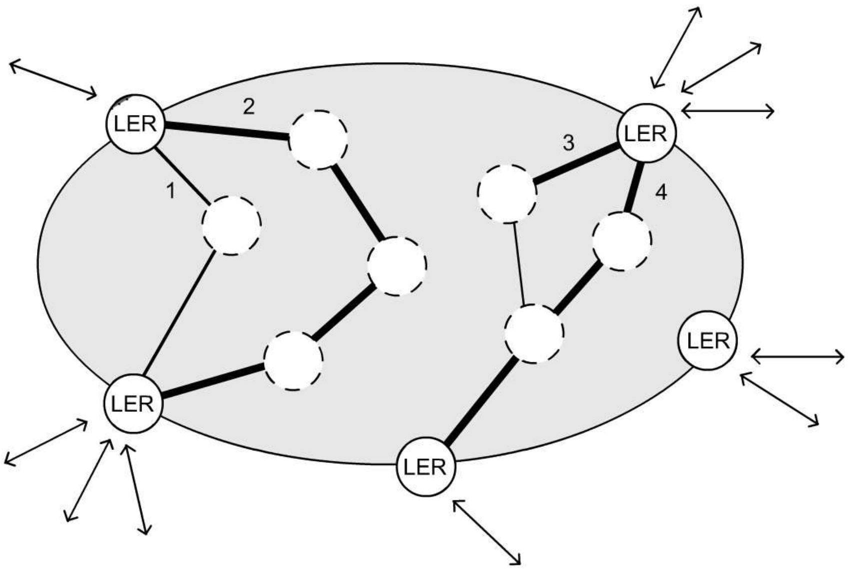

Figura III.3: Domínio MPLS e Engenharia de Tráfego.

ser usado para classificar e priorizar o tráfego originário do mesmo servidor, assim o tráfego de ida e volta serão encaminhados para recursos especiais ou dedicados. Mas pode-se ir além desses campos do cabeçalho IP, e classificar o fluxo baseado em campos do cabeçalho do protocolo de transporte, como o UDP e o TCP. O campo de porta desses protocolos significa, na maioria das vezes, uma aplicação específica, como http, e-mail, ou videoconferência. Então pode ser desejável que somente os tráfegos de uma certa aplicação utilizem um enlace específico, seja por questões de engenharia de tráfego ou pelas próprias características do meio físico de transmissão desse enlace, que pode oferecer baixo atraso, mesmo que tenha pouca banda, isso porque para certas aplicações o atraso é crítico. Veja esse exemplo na Figura III.4, o enlace de maior banda, é mais utilizado por aplicações de transferência de dados, como http, e-mail, ftp e *peer-to-peer*, mas como passa por vários roteadores seu atraso é insatisfatório para as aplicações multimídias. Já o enlace de 1 Mbps, não oferece uma banda passante grande, mais por não ser utilizado pelos outros fluxos e não ter vários saltos, é preferível para a transmissão de videoconferências e voz, pois o atraso será menor.

Porém antes de utilizar os LSPs, estes devem ser configurados, isso é feito através distribuição dos rótulos na rede. A distribuição dos rótulos consiste em configurar os

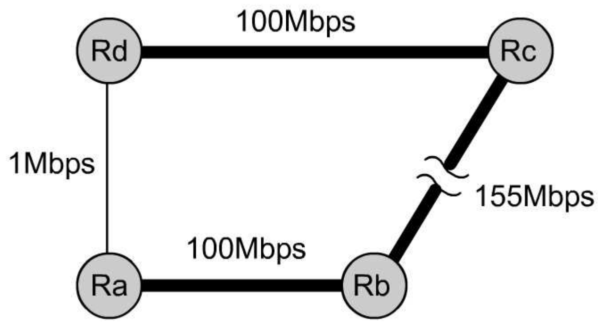

Figura III.4: Engenharia de Tráfego.

roteadores de maneira que esses sejam capazes de identificar um pacote pelo seu rótulo na interface de entrada, associá-lo a uma FEC/LSP e determinar a interface e o rótulo de saída. Essa configuração tem que ser feita em todos os roteadores pertencentes ao domínio pelos quais o LSP passará. A configuração dos LSPs pode ser feita de maneira estática, pelo administrador do domínio, desta forma o administrador tem que configurar os rótulos de entrada e saída em cada roteador por onde o LSP passará. Caberá a ele verificar que não aconteçam conflitos na distribuição de rótulos, ou seja, não permitir que duas FECs/LSPs tenham um mesmo rótulo em um mesmo enlace. A maneira alternativa é através dos protocolos de distribuição de rótulos, o LDP (*Label Distribuition Protocol*) [13] é o mais conhecido nas redes MPLS. O RSVP, que adicionou funcionalidades de engenharia de tráfego e de compatibilidade com os rótulos MPLS, se tornou outro protocolo de distribuição de rótulos chamado RSVP-TE, onde o TE significa *Traffic Engineering*. O próprio LDP também foi modificado para ter a funcionalidade de engenharia de tráfego, que resultou no CR-LDP [14], onde o CR (*Constraints Routing*) significa roteamento baseado em restrições.

III.6 Aplicação do MPLS

Nesta seção discute-se a aplicabilidade do MPLS em um ambiente de produção, e tenta-se mostrar suas vantagens e desvantagens, fazendo uma ligação entre a teoria e os experimentos práticos feitos.

O MPLS é bastante útil para tornar o encaminhamento dos pacotes mais eficiente em uma rede com muitos nós. Em uma rede, onde o atraso e a *jitter* são críticos, o MPLS é um bom candidato a solução. Além do atraso, é provável que a capacidade de encaminhamento de cada roteador aumente já que um pacote será roteado com menos custo computacional. Assim, caso o gargalo da rede seja a falta de capacidade computacional dos roteadores, o desempenho da rede poderá obter uma melhora. Mas, caso o gargalo seja a falta de capacidade dos enlaces, e não dos roteadores, então o desempenho possivelmente não sofrerá alterações consideráveis.

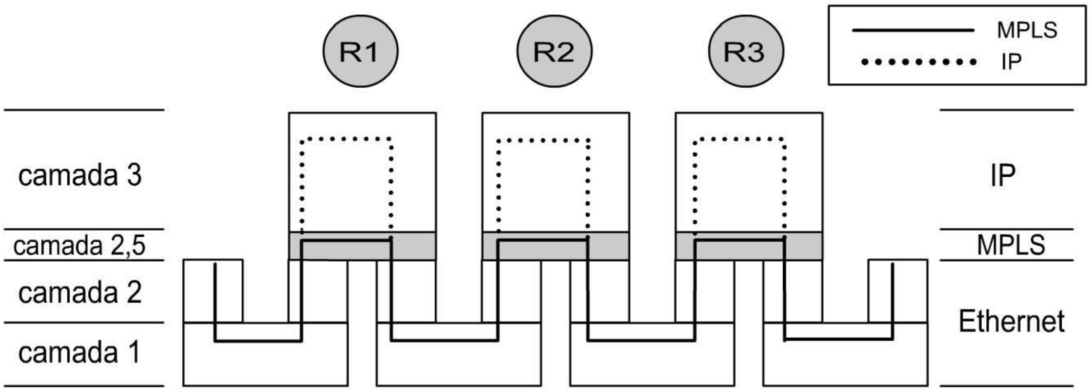

Figura III.5: Encaminhamento do pacote MPLS.

Apesar das vantagens citadas, o MPLS atualmente não é mais utilizado somente como um mecanismo de aumentar a capacidade agregada de uma rede. Sua versatilidade quanto a sua flexibilidade de configuração são características que pesam a seu favor, pois um administrador de redes terá uma rede mais eficiente. Uma rede mais versátil significa poder ter ao mesmo tempo tráfegos de voz, vídeo e dados, sem que haja interferência de um fluxo no outro. E ao mesmo tempo, esses fluxos serão distribuídos melhor, usando mais eficientemente os recursos da rede. Existe Engenharia de Tráfego, pois a distribuição dos fluxos pode ser controlada, trazendo para o administrador o controle de onde os fluxos devem passar, e ao mesmo tempo dar a possibilidade de distribuir de maneira mais adequada para suas políticas administrativas. Existe também Qualidade de Serviço, pois será possível que o administrador queira que somente os clientes prioritários que pagam mais, utilizem um certo caminho da rede, recurso ou enlace, enquanto os outros utilizem os recursos padrões.

III.6 Aplicação do MPLS

A engenharia de tráfego pode ser implementada com protocolos de roteamento, tanto de estado de enlace quanto de vetor distância, pois o conceito trata-se apenas de uma maneira mais eficiente de utilizar os recursos da rede, geralmente feito através de múltiplos caminhos e balanceamento de carga. O problema de fazer isso com um protocolo de roteamento é a complexidade das regras inseridas nas tabelas de roteamento, além da necessidade de inserí-las em todos os roteadores da rede. O processamento de definir a direção do fluxo seria realizado a cada salto, diferente do MPLS, que quando um fluxo é associado a um LSP seu destino é automaticamente definido. Como o MPLS traz para as bordas do domínio toda a complexidade das decisões de encaminhamento, a rede fica aliviada computacionalmente em seu núcleo, podendo aumentar sua capacidade agregada de encaminhar pacotes.

Capítulo IV

A Implementação do Ambiente de Testes

Neste capítulo é descrita a metodologia utilizada na implementação. Também é explicado o motivo da escolha de topologia usada, suas vantagens e desvantagens. Foi implementado um ambiente de testes formado por roteadores PCs, executando o sistema operacional Linux 2.4 da distribuição RedHat.

Para se testar a engenharia de tráfego é necessário uma topologia de rede onde os roteadores tivessem pelo menos dois caminhos possíveis até o destino. Cada um desses caminhos teria características diferentes, por exemplo, um teria baixo atraso, enquanto outro teria grande banda passante, baixo custo ou confiabilidade. Através dessas características da rede, que são comentadas adiante na seção da Topologia, é possível a utilização de caminhos distintos que favoreçam parâmetros de QoS diferentes, o que permite a diferenciação de serviço. Além disso, os múltiplos caminhos possibilitam a engenharia de tráfego.

Para a implementação do Ambiente de Testes utilizaram-se ferramentas livres e de código aberto. O sistema operacional utilizado foi o Linux 2.4 [15] e a distribuição foi o RedHat versões 9.0 e 7.3. A escolha do Linux se deve porque este é o sistema operacional livre e aberto mais utilizado, e também devido ao amplo suporte a mecanismos de controle de tráfego. Esses mecanismos poderiam ser eventualmente utilizados no projeto para fazer reservas de recursos para um LSP, ou simplesmente reduzir a banda disponível para uma classe de serviço, podendo ser relevante nos testes de QoS e diferenciação de serviços na topologia.

IV.1 Suporte MPLS em Linux

O software que dá suporte a MPLS no Linux é o `mpls-linux` [16], baseado no código da Nortel Networks mas tem seu desenvolvimento independente hospedado sob o domínio da SourceForge [17], autorizado pela Nortel desde que o desenvolvimento se mantenha portável. A SourceForge é um sítio de apoio ao desenvolvimento de software livre para a plataforma Linux. O sítio está disponível para a divulgação e funciona como um ponto de encontro de usuários e desenvolvedores interessados em encontrar, para usar ou ajudar a desenvolver, softwares livres. Para se manter portável, o código da Nortel foi separado em dois, o `mpls-linux` é o suporte MPLS para o Linux, inclui *patch* do *kernel* e ferramentas administrativas, e o `ldp-portable` é o protocolo de distribuição de rótulos portável.

Para transformar um Roteador Linux em um Roteador com suporte a MPLS, ou um LSR (*Label Switch Router*), é necessário que o código fonte do *kernel* do Linux seja modificado através de um *patch*, que funciona como um "remendo" do sistema operacional. Em seguida o kernel deve ser recompilado e posto operacional. Este "novo *kernel*" terá uma função que recebe o pacote antes da função do roteamento IP. Esta função determinará se o quadro Ethernet recebido deve ser encaminhado pelo MPLS, através das tabelas de comutação que serão criadas, ou se o quadro deve ser encaminhado para a função do roteamento IP, que normalmente seria feito sem o suporte ao MPLS.

Após esse primeiro passo, o *kernel* do roteador é capaz de identificar um pacote que esteja com o rótulo MPLS, mas isso não é suficiente para que o *kernel* identifique e encaminhe o pacote. É também necessário que as tabelas de comutação sejam preenchidas corretamente para que o *kernel* desse LSR identifique e encaminhe pacotes MPLS. Essas tabelas, a ILM e a FTN, definem qual será a direção dos pacotes.

A ILM e FTN podem ser preenchidas estática ou dinamicamente. A maneira dinâmica é feita sob demanda pelo protocolo de distribuição de rótulos que pode ser o LDP, CR-LDP ou RSVP-TE. A maneira estática é feita manualmente pelo administrador da rede através de uma interface de administração MPLS. O software linux-mpls possui uma ferramenta para adicionar essas entradas estáticas nas tabelas MPLS do *kernel*. Essa ferramenta é o `mplsadm2`, que através da linha de comando possibilita que entradas sejam adicionadas modificadas e removidas das tabelas ILM e FTN. Além disso, a ferramenta faz a associação entre as entradas das tabelas, que é necessária para o encaminhamento dos pacotes MPLS. Também existe um *patch* para o *daemon* de roteamento zebra [18] que implementa o LDP, para fazer as entradas dinâmicas. O zebra é um software livre, sob a licença GPL (*GNU Public License*), que implementa diversos protocolos de roteamento.

O `mplsadm2` não é suficiente para poder implementar um domínio MPLS, já que este modifica somente as tabelas MPLS do *kernel*. Isso é suficiente para os LSR do núcleo do domínio, que vão receber pacotes já rotulados. Para os LSR da borda do domínio é necessário algum mecanismo que possibilite a associação de datagramas IP com as entradas da tabela FTN que determinam a interface, o próximo salto e o rótulo. Esse conjunto de informações vai determinar a qual FEC/LSP esse pacote deve ser associado, e qual o próximo LSR. Para isso os desenvolvedores do `mpls-linux` fizeram também *patchs* para as ferramentas `iptables` [19] e `IPRoute2` [20].

Através do `iptables` é possível adicionar linhas na lista de regras de pré-roteamento para que os pacotes que estiverem de acordo com o padrão determinado sejam encaminhados utilizando o LSP especificado pelo administrador. Já a ferramenta `IPRoute2`, contém dois programas, o `ip` e o `tc`. O `tc` vem de *Traffic Control*, e devido a sua aplicabilidade não será utilizado, pois o tratamento diferenciado não será efetuado através da reserva de recursos. O `ip` é uma ferramenta completa para configuração de quase tudo necessário para a camada de rede no Linux, desde a interface de rede até a manutenção da tabela de roteamento. Para a associação do datagrama IP a um LSP foi utilizado justamente a parte referente à manutenção da tabela de roteamento do comando `ip`. A diferença da ferramenta `iptables` para ferramenta `IPRoute2` está na atuação das estruturas do *kernel*. O `iptables` tem a sua cadeia de regras de pré-roteamento antes do pacote ser analisado pela função de encaminhamento do *kernel*. Assim, a vantagem de usar o `iptables` é poder analisar todos os campos do cabeçalho IP, TCP e UDP, ser for o caso, proporcionando uma maior flexibilidade de configuração. Por outro lado, o comando `ip` do `IPRoute2` é mais limitado nas possibilidades de opções e campo de atuação, já que suas regras terão que ficar na tabela de roteamento e somente o endereço IP, de origem ou destino, são analisados. Sua grande vantagem é poder ficar na tabela de roteamento, assim, as regras serão tratadas pela função de roteamento do *kernel*, alterando a estrutura de dados do *kernel* minimamente, otimizando o desempenho do encaminhamento do pacote.

Portanto, devido às características diferentes de cada ferramenta, cada uma delas foi utilizada para uma determinada finalidade. O comando `ip` é usado para colocar os LSPs padrão pelos quais os pacotes devem ser encaminhados, isso faz com que os LSPs atuem como túneis que levam a uma única rede. Assim para cada rede do *backbone*, é identificada por um LSP. O comando `ip` é usado para fazer balanceamento de carga pelo endereço IP de destino. Isso é realizado colocando um LSP diferente para faixas de endereços IPs de uma mesma rede. Assim os pacotes para uma mesma rede poderiam seguir caminhos distintos dependendo do destino do fluxo.

No caso da diferenciação de serviço, onde o que determina um LSP é uma informação mais específica, como a porta de origem ou de destino do protocolo de transporte (TCP e UDP), o `iptables` foi usado.

IV.1.1 *Patch* do *kernel*

O *kernel* é o núcleo do sistema operacional Linux. O *kernel* é responsável pelo interfaceamento do hardware com os softwares de nível de privilégio de usuário (*user space*). Além disso, faz o roteamento de pacotes quando o Linux é configurado para ser um roteador. O Linux como roteador é uma opção cada vez mais comum, devido ao baixo custo de implementação, facilidade de se encontrar documentação e constante atualização do

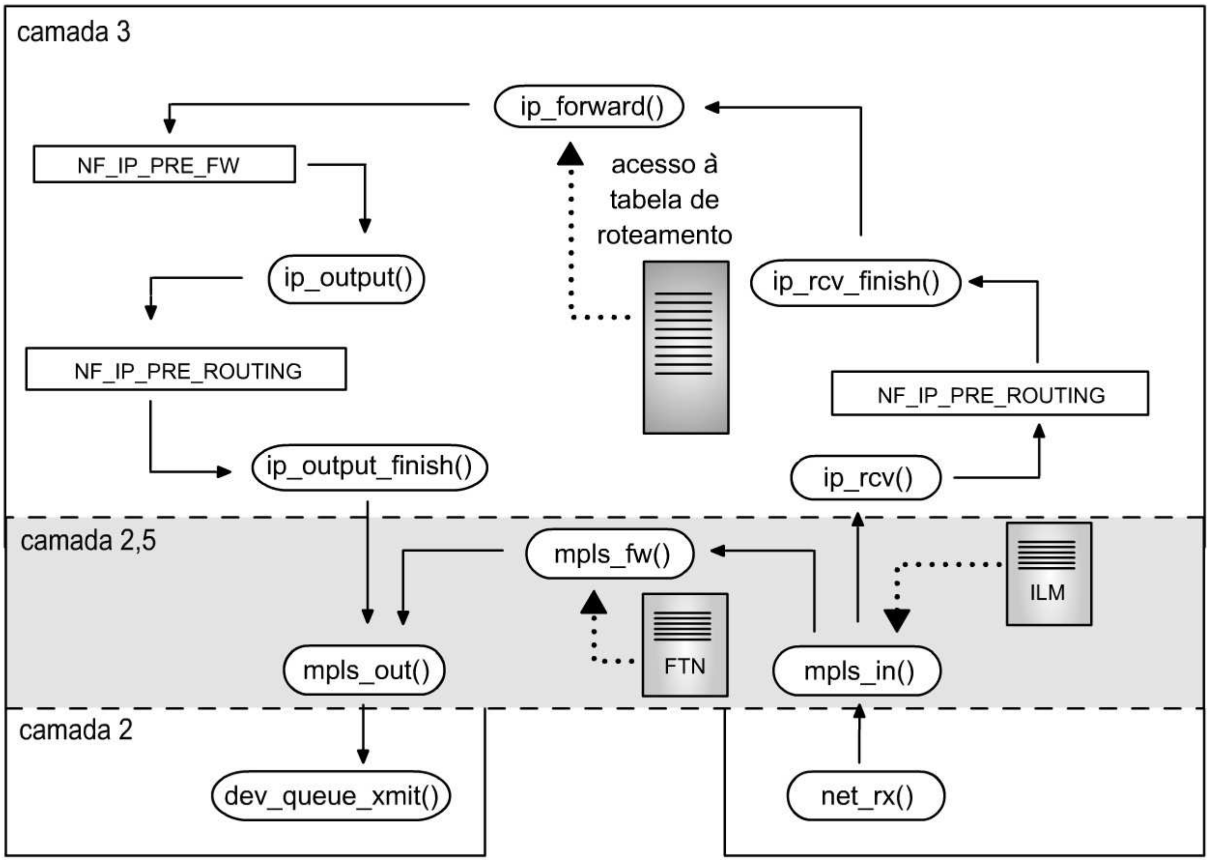

Figura IV.1: Funcionamento do *kernel*.

código fonte, possibilitando a atualidade com novos protocolos e novas funcionalidades.

O *patch* do *kernel* consiste em um arquivo, com parâmetros que indicam aonde e como devem ser feitas as modificações nos arquivos fontes do *kernel*. Como esse arquivo necessita de referências para localização das modificações, algum trecho de código, é necessário que a versão do *kernel* da qual o *patch* foi originado seja utilizada. Caso contrário, existe a possibilidade dessas referências não existirem, o que acarretará em modificações errôneas com conseqüências imprevisíveis.

Para aplicar o *patch* é necessário o fonte do *kernel*. São também necessários as bibliotecas `glibc` e o compilador `gcc` para compilar o novo *kernel* com o suporte a MPLS. Os fontes do *kernel* e do `mpls-linux`, são instalados nos diretórios `/usr/src/linux` e `/usr/src/mpls-linux`, respectivamente. O *patch* é instalado no kernel. Segue abaixo a seqüência de comandos para instalação dos softwares.

```
INSTALANDO A PATCH:
  root@lsr1#cd /usr/src/linux
  root@lsr1#patch -p1 < ../mpls-linux-1.1/patches/linux-mpls.diff
  root@lsr1#make menuconfig
CONFIGURANDO O KERNEL:
  a. Sob "[Code maturity level options]" selecione:
          -  "[Prompt for development and/or incomplete code/drivers]"
   b. Sob "Networking Options" selecione:
           -  "Kernel/User netlink socket"
           -  "  Routing messages"
           -  "Multi-Protocol Label Switching"


COMPILANDO:
      root# make dep; make clean; make bzImage;
      root# make modules; make modules\_install
      root# cp arch/i386/boot/bzImage /boot/
      root# vi /etc/grub.conf
VERIFICANDO:
      root# dmesg | grep MPLS
      MPLS version 0.993 06/05/2001 jleu@mindspring.com
      MPLS Tunnel interface
      root#
```

Após a compilação e instalação, o novo *kernel* do Linux está habilitado para comutar pacotes MPLS.Como a comutação se realiza através das tabelas MPLS do *kernel*, deve-se ter uma ferramenta que possibilite o administrador do domínio modificar estas tabelas, ou então algum protocolo de distribuição de rótulos para fazer essas modificações automaticamente.

IV.1.2 Ferramenta de administração `mplsadm2`

O `mplsadm2` é um software distribuído em conjunto com o *patch* do *kernel* do Linux [16]. Esse programa faz a interface do administrador com as tabelas MPLS do *kernel*. Através do `mplsadm2` pode-se configurar entradas MPLS estáticas, ou seja, definidas pelo administrador da rede. Isso significa que todos os parâmetros serão introduzidos pelo administrador manualmente. Além disso, o administrador terá que configurar as entradas em todos os roteadores do domínio por onde o "túnel" LSP passará.

Como o `mplsadm2` é um programa de linha de comando, os comandos se tornam extensos, já que a quantidade de parâmetros é grande. Com ele é possível inserir, modificar, apagar e fazer associações entre as entradas das tabelas MPLS do *kernel*. A tabela FTN é responsável por definir a interface e rótulo de saída dos pacotes, e a ILM define qual rótulo deve pertencer o pacote e de qual interface este deve ter sido originário para ser encaminhado para a função de encaminhamento MPLS. Essa associação entre as entradas da tabela FTN de saída, e as entradas da tabela ILM de entrada, é a responsável pela comutação dos pacotes MPLS dentro do roteador.

Os comandos abaixo inserem uma entrada na tabela FTN. Essa tabela é responsável por determinar o rótulo, a interface de saída e o próximo salto do pacote.

```
#adiciona o NHLFE ao FTN
(1) KEY=‘mplsadm2 -A -O 0 | awk ’\{print $2\}’‘
(2) mplsadm2 -O $KEY -o push:gen:200:set:eth2:ipv4:192.168.12.37
```

O primeiro comando simplesmente insere uma entrada na FTN. A opção "-A" define que se trata de uma adição e a "-O" define que a adição é na tabela FTN de saída. O segundo comando define o que será executado na entrada da tabela FTN, identificada pela chave $KEY. Essa chave recebeu a saída do primeiro comando, e contém o identificador da entrada inserida no comando anterior. Continuando a análise do comando, o "push", determina que o rótulo desta entrada deve ser inserido no pacote, independente de haver ou não outro rótulo no pacote. O "gen", de genérico, significa que o rótulo é genérico, ou seja, terá um cabeçalho MPLS [12]. Diferente do Ethernet, existem tecnologias de rede que não precisam desse cabeçalho, como o ATM. O valor "200" é o valor em decimal do rótulo. A parte "set:eth2:ipv4:192.168.12.37" define que o próximo salto será o o roteador com o endereço IP 192.168.12.37 e tem como interface de saída a interface `eth2`.

Somente com essa entrada na tabela FTN de saída configura-se o roteador LSR da borda do domínio. Para configurar os roteadores de núcleo é necessário que além dessa, uma outra entrada na tabela ILM seja inserida e associada com a entrada da tabela FTN de saída. Como mostrado abaixo:

```
#habilita a interface a ouvir pacotes mpls
(1) mplsadm2 -L eth3:0
#adiciona o NHLFE da FTN
(2) KEY=‘mplsadm2 -A -O 0 | awk '{print $2}'‘
(3) mplsadm2 -O $KEY -o push:gen:202:set:eth1:ipv4:192.168.22.6
#adiciona um NHLFE ao ILM e o associa ao NHLFE do FTN
(4) mplsadm2  -A -I gen:201:0
(5) mplsadm2  -B -I gen:201:0 -O $KEY
```

O primeiro comando habilita o MPLS na interface `eth3` e associa ao "*label space*" zero. O *label space* determina o escopo ao qual essa interface é associada, determinando qual tabela ILM é acessada pelas funções de comutação MPLS. O segundo e terceiro comando já foram explicados, e somente inserem uma entrada na tabela FTN de saída. No quarto comando a opção "-A" define que a operação é uma adição. A opção "-I gen:100:0" determina que a adição é feita na tabela ILM e que o rótulo que esta deve se associar é 100, no "label space" zero. No quinto comando a opção "-B", junto com "-O $KEY", associa a entrada inserida na ILM, através do quarto comando, à entrada inserida pelo terceiro comando na FTN, através do segundo e terceiro comando.

IV.1.3 As Ferramentas `ip`, `tc` e `iptables`

A ferramenta `ip` atua no *kernel*, modificando a maneira como os pacotes são encaminhados através da tabela de roteamento. A adição de entradas nas tabelas MPLS do *kernel* não é suficiente para que os roteadores comecem a comutar os pacotes MPLS. A associação de fluxos aos LSPs criados é necessária nas bordas do domínio, pois sem isso, não existirão pacotes MPLS para serem comutados no núcleo. Essa associação entre fluxos e LSP faz a rotulação dos pacotes IP e permite que os roteadores do núcleo do domínio utilizem as tabelas MPLS.

```
#adiciona a rota para apontar para o NHLFE
ip route add 192.168.0.0/24 via 192.168.11.37 lsp \$KEY
```

O comando acima é executado logo após os comandos da ferramenta `mplsadm2` modificarem as tabelas MPLS. Ele é o responsável por fazer a associação de todos os fluxos que se destinam à rede 192.168/16 para o LSP identificado pela chave $KEY.

O `iptables` possibilita que a associação de fluxos seja feita considerando outras informações além do endereço IP, como, por exemplo, os fluxos que tenham no protocolo TCP porta de origem ou destino 22. Fazendo assim as sessões interativas de SSH serão os alvos da regra.

A ferramenta `tc` é executada no *kernel* antes da função de transmissão que envia os pacotes para a interface de rede. Através desta ferramenta é possível limitar a capacidade de transmissão de uma interface e criar uma interface com uma banda virtual reduzida. Através dessa ferramenta também é possível fazer reserva de recurso para fluxos específicos, isso é feito através de filtros que utilizam as informações do cabeçalho do protocolo de rede e de transporte.

É importante observar que todas as três ferramentas foram baixadas da Internet com o *patch* para suporte ao MPLS já aplicado. Possibilitando que esses comandos fossem aplicados a fluxos de LSPs.

IV.2 As Ferramentas de Medição

A ferramenta `iperf` foi usada para medir banda disponível nos pares cliente-cliente, cliente-roteador, roteador-cliente e roteador-roteador. Ela também foi utilizada como ferramenta geradora de tráfego para inundar a rede de tráfego e torná-la congestionada quando necessário. A medida do atraso foi feita através de *ICMP echo request/reply*, com o software `ping`, disponível em qualquer distribuição Linux.

IV.2.1 `iperf`

Para medir a banda disponível utilizou-se o programa `iperf` [21] que tem como uma de suas medidas a vazão efetiva do TCP. Desta maneira, obtém-se a banda disponível entre as máquinas, já que o tamanho da janela do TCP é ajustado automaticamente para utilizar o máximo de recursos disponíveis do enlace, ou até da rede, caso o fluxo utilize um caminho com vários enlaces.

Ao mesmo tempo em que o `iperf` era usado como ferramenta de medição, ele também foi usado como ferramenta geradora de tráfego UDP CBR (*Constant Bit Rate*). Apesar dessa característica de tráfego ser um modelo simplista para a maioria dos tráfegos da Internet atualmente, pode ser considerado um bom modelo para tráfegos de voz não comprimida ou não codificada, e atinge o seu objetivo do tráfego de fundo, que é encher o canal.

Para calcular o *jitter* da transmissão, também usa-se a ferramenta `iperf`. O *jitter* é calculado automaticamente pelo servidor `iperf` como especificado em [22], e o valor é posto na saída do console do programa.

O `iperf` consiste em um cliente e um servidor. Através de opções de linha de comando determina se um novo processo será um cliente ou um servidor. As opções de linha de comando determinam também se o processo cliente, ou servidor, será relativo a um tráfego TCP, ou UDP. A seguir estão os principais comandos utilizados para as medidas e geração de tráfego UDP.

```
# servidor de fluxo TCP
iperf -s
# cliente de fluxo TCP para o servidor r1
iperf -c r1
#servidor de fluxo UDP
iperf -s -u
# cliente de fluxo UDP para o servidor r1 com banda 5 Mbits/s
iperf -c r1 -b 5m
```

É importante notar que o nome do roteador "r1" deve estar registrado no arquivo `/etc/hosts/`, mapeando esse nome no endereço IP do roteador R1, caso contrário o endereço IP deve ser utilizado no lugar do "r1".

A seguir temos a saída do console de comando do servidor que contém os dados referentes à perda, ao *jitter* e à percentagem de pacotes UDP entregues. A perda de pacotes é detectada através da falta de seqüenciamento dos pacotes recebidos do fluxo UDP. Ocorrem perdas quando há saltos na numeração dos pacotes recebidos.

```
root@r1# iperf -s -u -i 1
Server listening on UDP port 5001
Receiving 32768 byte datagrams
UDP buffer size:  128 KByte
[ 3] local r1 port 5001 connected with c15 port 11303
[ID] Interval     Transfer Bandwidth   Jitter   Lost/Total Datagrams
[ 3] 0.0- 1.0 s  1.3MB   10.0Mbits/s 0.430 ms 0/41 (0%)
[ 3] 1.0- 2.0 s  1.1MB    8.5Mbits/s 5.996 ms 6/40 (15%)
[ 3] 2.0- 3.0 s  1.2MB    9.7Mbits/s 0.796 ms 1/40 (2.5%)
...
[ 3] 9.0-10.0 s  1.2MB   10.0Mbits/s 0.407 ms 0/40 (0%)
[ 3] 0.0-10.0 s 12.3MB    9.8Mbits/s 0.407 ms 7/401 (1.7%)
```

IV.3 A Topologia

Para se realizar os testes é necessário uma topologia de rede que permita múltiplos caminhos entre a origem e o destino e que os caminhos apresentem características diferentes. Nestas condições é possível fazer engenharia de tráfego, realizando balanceamento de carga nos múltiplos caminhos. Desta maneira os enlaces de pequena banda, porém de baixa latência, também serão utilizados, fato que não seria possível tão facilmente sem a engenharia de tráfego disponível em uma rede MPLS, como é comentado adiante. Esses enlaces de menor banda serão utilizados para aplicações que não requerem muita banda, mas que tenham necessidades de um atraso e *jitter* com valores dentro de uma faixa determinada.

É importante explicar porque o enlace de baixa banda é o que possui menor atraso. Isso é característica da topologia, existem caminhos de poucos saltos que propositalmente tem pouca banda. A topologia está ilustrada na Figura IV.2. Existem caminhos entre pares origem-destinos que otimizam um parâmetro de QoS. Assim, dependendo do que se pretende favorecer, seja atraso ou banda passante, o caminho escolhido será diferente.

Nas redes clientes, todos os computadores que funcionam como emissor e receptor de tráfego estão conectados com o seu respectivo roteador através de um comutador, ou de um cabo "*cross-over*". Isso foi feito para eliminar a hipótese do gargalo das medidas ser o enlace entre o PC cliente e o seu roteador, já que desta maneira esse enlace estará em modo "*full-duplex*", possibilitando o envio simultâneo de dados bidirecional.

IV.3.1 A configuração lógica

A faixa de endereços IP alocada para cada enlace é a `192.168.xx/24` onde o terceiro byte define qual é o enlace. Como todos os roteadores fazem parte da rede do laboratório do GTA, todos têm um endereço IP da rede GTA. O último byte do endereço da rede GTA constituirá também o último byte dos endereços das redes da rede de testes. Assim, a máquina grumari que tem o IP `xxx.xxx.xxx.6` na rede GTA terá os endereços

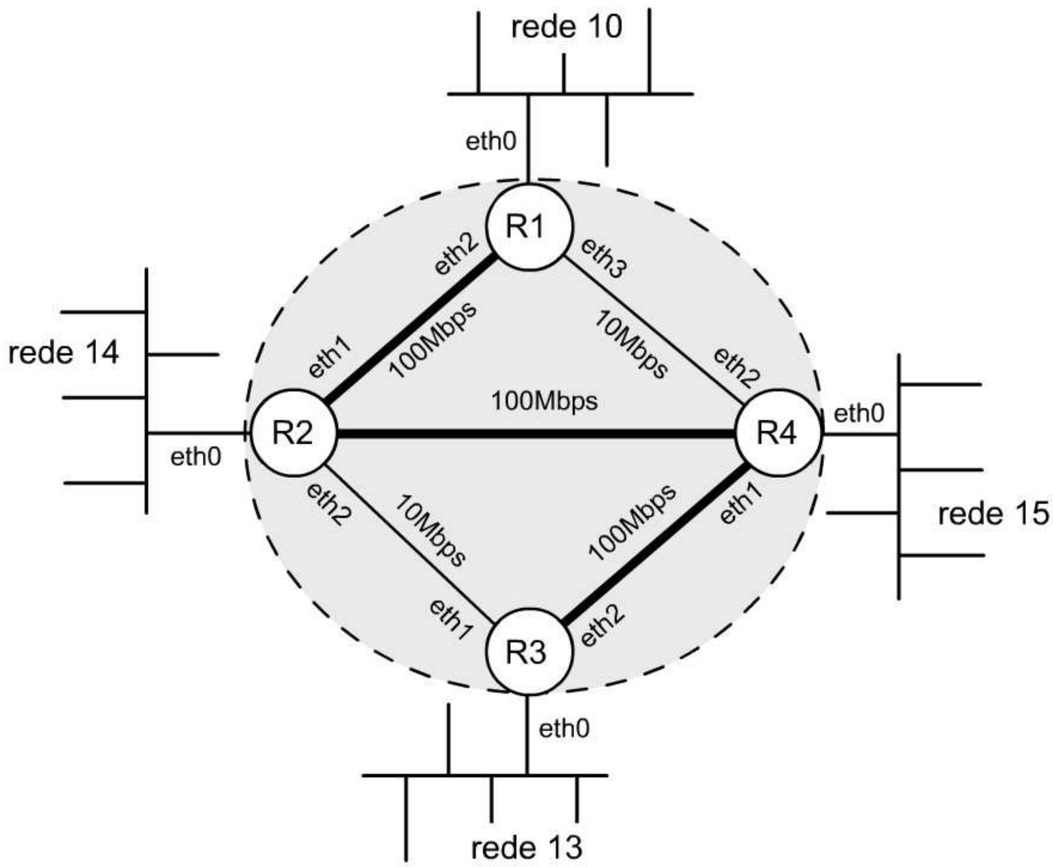

Figura IV.2: Ambiente de Teste.

`192.168.13.6`, `192.168.12.6` e `192.168.21.6` nas interfaces `eth0`, `eth1` e `eth2`, que estão conectadas as redes 13, 12 e 21, respectivamente. Na Tabela IV.1 estão as faixas das redes do ambiente de testes.

Além de funcionar como um comutador MPLS, um LSR, os roteadores também estão habilitados a funcionar como um roteador IP normal. Assim rotas ainda serão acessadas para o roteamento de pacotes que não sejam rotulados com o MPLS na borda do domínio. Os fluxos não marcados com o rótulo MPLS vão seguir os caminhos determinados pelas tabelas de roteamento IP.

O domínio MPLS é constituído pelos roteadores R1, R2, R3 e R4, que a linha tracejada contorna, veja Figura IV.2. Para construir um domínio de diferenciação de serviços, foram utilizadas algumas das funcionalidades do MPLS para tornar mais eficiente o uso dos

Tabela IV.1: Endereçamento IP.

Roteador	Endereço da Rede
R1-R2	192.248.11/24
R2-R3	192.248.12/24
R3-R4	192.248.22/24
R4-R2	192.248.30/24
R4-R1	192.248.21/24
rede10	192.248.10/24
rede13	192.248.13/24
rede14	192.248.14/24
rede15	192.248.15/24

recursos da rede ao mesmo tempo em que garantia a qualidade de serviço fornecida ao usuário. Em um domínio de diferenciação de serviços DS, a diferenciação propriamente dita se faz através da marcação do datagrama IP com um identificador que determinará, nos próximos saltos, quais recursos estão disponíveis para esse pacote marcado. Na rede de testes criada não existirá a marcação de datagramas IP como em um domínio DS. A diferenciação e a garantia de QoS será feita através dos LSPs. Na borda do domínio MPLS, o tráfego será encaminhado para um dos LSPs. Dependendo do LSP para a qual o fluxo foi encaminhado, enlaces diferentes serão utilizados e recursos reservados diferentes serão consumidos. Desta maneira, pode-se fazer engenharia de tráfego e diferenciação de serviço ao mesmo tempo, sem realmente usar a arquitetura *DiffServ*.

Visando a diferenciação de serviço os LSPs foram configurados de maneira a disponibilizar os roteadores de borda vários caminhos até um destino. Tomando como exemplo o roteador R3 como origem e a rede 10 como destino. Esse roteador deve ter pelo menos dois LSPs para a rede 10. Um LSP deve percorrer o caminho que disponibilize mais banda. Este LSP será utilizado para passar todo o fluxo não prioritário. O outro LSP deve se preocupar em passar pelo caminho que proporcione o menor atraso possível. Para isso deve-se otimizar enlaces pouco utilizados, mesmo que de pouca banda, e escolher um caminho que passe por um menor número de saltos. Esse segundo LSP será o responsável por encaminhar os fluxos críticos no tempo e prioritários, desde que não necessitem de muita banda passante.

Através desses LSPs restritos a usuários prioritários, clientes serão favorecidos e terão garantia de QoS. Isso será feito através desses túneis específicos, que utilizarão recursos da rede que não estão disponíveis para o usuário comum.

Capítulo V

Resultados Obtidos e Análise de Desempenho

Neste capítulo são analisados os resultados obtidos nos testes de Qualidade de Serviço. É analisado como a implementação dos mecanismos de QoS aumenta a eficiência da rede devido a possibilidade da Engenharia de Tráfego. Também é descrito nesse capítulo como os testes de desempenho foram realizados, e como as ferramentas foram usadas para alcançar a diferenciação de serviços e a engenharia de tráfego.

V.1 Os Testes de Desempenho

Os testes de têm como objetivo a comparação do desempenho da rede sem o suporte ao MPLS, considerando dois casos: o roteamento dos pacotes através do cabeçalho IP e da rede com suporte MPLS, permitindo engenharia de tráfego e QoS. No roteamento IP as rotas são aprendidas através do protocolo de roteamento vetor distância RIP [23] e estado de enlace OSPF [24]. Também é analisado o desempenho da rede com esses protocolos, comparando com o desempenho do MPLS.

Os parâmetros de QoS são monitorados na rede com e sem o MPLS. Isso é realizado para verificar o impacto e a efetividade da diferenciação de serviços efetuada. Em relação à engenharia de tráfego, o MPLS é uma ferramenta mais flexível para o administrador na implementação de políticas de alocação de recursos tornando a rede mais eficiente. Esse aumento da eficiência caracteriza a Engenharia de Tráfego, e tem como conseqüência o aumento da capacidade agregada da rede.

V.2 Capacidade Agregada do Roteador

A capacidade agregada de um roteador é uma medida em Mbps (Megabits por segundo). Ela é a quantificação de quanto o roteador é capaz de encaminhar pacotes entre as suas interfaces. Geralmente, os fatores que limitam esse índice de desempenho de um roteador são associados a características do hardware. A capacidade de processamento do roteador, geralmente relacionado a sua freqüência de operação do processador, é um possível gargalo. Mas a freqüência do processador pode não ser determinante dependendo da arquitetura do hardware e de qual seja a capacidade de processamento das interfaces de rede. A capacidade do elemento comutador, que faz o encaminhamento dos pacotes de uma interface a outra é outro gargalo. Essa capacidade pode ser aumentada se a arquitetura permitir paralelismo na comutação de pacotes entre as interfaces. Para esse paralelismo a arquitetura não pode ser um barramento, obviamente, uma matriz de comutação é usada nesse caso. Essa é uma preocupação para roteadores de alto desempenho e de arquiteturas proprietárias [25].

Os roteadores utilizados são PCs, e portanto não possuem características de hardware aperfeiçoado para o roteamento de pacotes. O processamento de todos os pacotes é centralizado no processador da máquina que é um potencial gargalo. Todas as placas de rede se comunicam com o processador através de um barramento PCI, que é outro gargalo em potencial. A definição de qual o fator limitador da capacidade agregada do roteador e as justificativas para tal, não são abordados nesse trabalho, e não fazem parte deste estudo. Esse limite é quantificado para embasamento dos argumentos referentes aos testes e análises de desempenho da rede. Essa medição é feita através de várias conexões

TCP que passam pelo roteador que se quer medir a capacidade agregada. Para ter certeza que o gargalo não é a capacidade das interfaces, os fluxos são distribuídos entre todas as interfaces do roteador. Esse cuidado na medição é importante, pois é muito provável que caso os testes fossem realizados utilizando somente uma placa de 10 Mbps, com certeza o roteador não atingiria sua máxima capacidade de encaminhar pacotes antes da saturação da interface, tornando a medida falsa. Devido ao controle de fluxo do TCP, conforme se adiciona fluxos, a capacidade de rotear é igualmente distribuída entre os fluxos e, conseqüentemente, entre as interfaces de rede já que esses fluxos tem origens e destino escolhidos de maneira a utilizarem interfaces distintas, eliminando a possibilidade do gargalo da interface. Quando a capacidade agregada do roteador for atingida, a soma das taxas de transmissão dos fluxos TCP não será mais alterada, independente da adição de mais fluxos. A soma de todas as taxas de transferências de todos os fluxos é a capacidade agregada desse roteador. A Tabela V.2 mostra os valores adquiridos nas medidas em cada roteador da topologia. As taxas foram medidas através do comando `iperf`.

Tabela V.1: Especificação Técnica dos Roteadores.

Dispositivo	R1	R2	R3	R4
eth0	-	dlink	dlink	3com
eth1	dlink	dlink	pcnet	3com
eth2	dlink	pcnet	3com	pcnet
eth3	pcnet	realtek	-	realtek
processador	P4 1.8GHz	AMD XP2k	AMD 1.2GHz	P4 1.8GHz
kernel	2.4.20	2.4.20	2.4.20	2.4.20
Mem. RAM	256MB	512MB	256MB	512MB

Tabela V.2: Capacidade Agregada dos Roteadores.

Roteador	Capacidade Agregada
R1	102.0 Mbps
R2	128.8 Mbps
R3	149.2 Mbps
R4	194.6 Mbps

V.3 Os Testes de Qualidade de Serviço

Através da diferenciação de serviços é implementado QoS no domínio MPLS. A diferenciação é efetuada através de LSPs, passando por caminhos distintos e conseqüentemente usando diferentes recursos, ou através da reserva de recursos nos roteadores. Na segunda opção tem-se o problema de disputa de recursos, que somente o escalonador do roteador consegue solucionar. Porém essa solução tem um custo computacional considerável. Optou-se neste trabalho por utilizar uma topologia que possibilite a diferenciação de serviços através da utilização de múltiplos caminhos, pois assim também se atende os requisitos de engenharia de tráfego.

Para testar a qualidade de serviço um exemplo trivial é utilizado. O mecanismo de QoS favorece um fluxo utilizando recursos reservados da rede. Enquanto isso, o outro fluxo tem um tratamento normal, utilizando os recursos padrão.

A situação criada para os testes de QoS consiste em dois PCs na rede cliente 10, Figura IV.2. O PC "videoserver" é o cliente preferencial, e usa recursos reservados, enquanto o PC "tx" usa os recursos padrão, que qualquer outro fluxo também está usando. Pela topologia da rede, e pelas necessidades de banda das redes clientes, mostrada na matriz de demanda de tráfego da Tabela V.3, o enlace R2-R4 fica constantemente congestionado, devido a banda requerida de R2 para R4, e de R4 para R2.

Essa matriz de tráfego, Tabela V.3, é inventada com valores apropriadamente escolhidos para que o ambiente seja favorável a implementação do MPLS. Essa matriz de tráfego não é totalmente impossível de ser aplicada a realidade, pela capacidade e disposição na topologia dos enlaces podemos imaginar essa rede como uma rede corporativa de alguma empresa com sede em cidades brasileiras. Os roteadores R2 e R4, por exemplo, seriam as sedes das cidades do Rio de Janeiro e São Paulo. Isso pode ser justificado devido ao número de enlaces pertencentes a esses roteadores, o que disponibiliza um alcance de um salto para qualquer destino na rede. Os roteadores R1 e R3, por outro lado, poderiam ser cidades do Sul e do Nordeste, que são ligadas diretamente com São Paulo e Rio, mas não são ligadas entre si, primeiro pela distância geográfica, pois uma está no Nordeste enquanto a outra está no Sul, e segundo pela falta de demanda, já que as atividades econômicas, comerciais e industriais estão concentradas no Rio e em São Paulo. Pelo mesmo motivo as demandas de tráfego são concentradas nas redes clientes dos roteadores R2 e R4, Rio e São Paulo.

Tabela V.3: Matriz de Demanda de Tráfego.

	R1	R2	R3	R4
R1	-	4 Mbps	3 Mbps	4 Mbps
R2	4 Mbps	-	4 Mbps	60 Mbps
R3	3 Mbps	4 Mbps	-	4 Mbps
R4	4 Mbps	60 Mbps	4 Mbps	-
Total	11 Mbps	68 Mbps	11 Mbps	68Mbps

É importante observar que essa é a matriz de demanda, isso não implica necessariamente que esses fluxos terão essas taxas de transferência. A configuração da rede define qual é a capacidade da rede, e conseqüentemente a capacidade de atender à demanda de cada um dos fluxos.

Devido a essa alta demanda de banda entre os roteadores R2 e R4, a comunicação entre os roteadores R1 e R3 se degrada. A degradação ocorre, em grande parte, devido à espera na fila de transmissão nos roteadores R2 e R4, o que acarreta atraso e *jitter* grandes. Outro possível motivo para a degradação é a quantidade de processamento nos roteadores R2 e R4, que estão sobrecarregados encaminhando o tráfego das redes clientes 14 e 15. Para aplicações multimídias, como sessões de terminal remoto, voz e vídeo, o atraso e o *jitter* são parâmetros de QoS muito importantes.

Para diminuir os valores do atraso e do *jitter* a valores controlados, existe um LSP específico para o tráfego sensível. Os LSPs com esse propósito passam por caminhos na rede que evitem gargalos e congestionamentos. A Tabela V.4 mostra o *jitter* nas diferentes configurações de rede do fluxo do videoserver até o PC c13, o PC cliente da rede 13.

Na Tabela V.4, a "Topologia IP" consiste na rede usando o protocolo de roteamento IP, que no caso é o OSPF com a banda passante como métrica. O congestionamento em ambos os casos consiste na inundação do enlace entre R2 e R4, pois é o enlace crítico para o OSPF. A topologia MPLS não exclui o roteamento IP, somente são adicionados os LSPs de forma que os fluxos prioritários não utilizem o enlace congestionado R2-R4.

Tabela V.4: *Jitter*.

Topologia	*Jitter*	Ping
IP	0.029 ms	0.547 ms
IP-congest.	0.265 ms	2.984 ms
MPLS	0.031 ms	0.600 ms
MPLS-congest.	0.150 ms	3.743 ms

Pode-se observar que o *jitter* do fluxo encaminhado para o LSP que passa pela seqüência de nós R3-R2-R1 é consideravelmente menor que quando é roteado pelo protocolo IP, passando por R3-R4-R2-R1. Isso se deve aos enlaces utilizados pelo roteamento IP. O protocolo de roteamento OSPF escolhe sempre o caminho de maior banda, gerando uma concentração dos fluxos da rede no enlace R2-R4, ocasionando o congestionamento. Esse congestionamento acarreta maiores tempos de espera em filas nos roteadores, e conseqüentemente maior atraso e maior *jitter*. Por outro lado, se todo o tráfego do cliente preferencial atravessar o enlace reservado para as aplicações sensíveis, as conseqüências serão indesejáveis. O congestionamento do enlace reservado certamente ocorrerá, pois ele é de pouca capacidade. E a falta de banda para aplicações do cliente redirecionado para o enlace reservado pode gerar insatisfação. Os enlaces com maior capacidade, mesmo que com muito tráfego, ainda vão oferecer uma taxa de transferência maior que o de baixa capacidade e com pouca utilização. Nesse sentido, a diferenciação de serviços deve ser bem planejada. Uma associação mal realizada entre um fluxo e recursos da rede acarreta na degradação de outros fluxos e até mesmo no não atendimento dos requisitos desse novo fluxo. Por esse motivo, todos os fluxos do cliente prioritário não são encaminhados para os recursos reservados da rede. O administrador deve ser cauteloso nas associações que faz.

Esse cuidado e verificação de recursos disponíveis é o Controle de Admissão. Como na rede de teste os recursos são associados manualmente, o Controle de Admissão é manual também. O Controle de Admissão será automático se a associação de fluxo com os recursos da rede for realizada através de algum protocolo de distribuição de rótulos, como o LDP. Porém não pode ser qualquer protocolo devido às restrições que a Engenharia de Tráfego requer. O protocolo deve ser habilitado com Engenharia de Tráfego, como os protocolos CR-LDP e RSVP-TE. A implementação desses protocolos ainda está em fase de desenvolvimento, então não foi possível fazer comparações práticas com esses protocolos, por enquanto.

V.4 Os Testes da Capacidade Agregada da Rede

A distribuição do tráfego entre os enlaces da rede, que tem o objetivo de tratar diferenciadamente alguns fluxos é também uma forma de Engenharia de Tráfego. Aliviar o tráfego de enlaces congestionados e redirecionando-os para enlaces livres torna a rede mais eficiente. Como conseqüência, sua capacidade agregada aumenta.

As rotas escolhidas pelo protocolo de roteamento estado de enlace usam os enlaces que tem maior banda nominal. Diz-se banda "nominal", porque a banda usada no cálculo não é necessariamente a banda disponível. Na verdade, é muito improvável que a banda nominal do enlace esteja disponível, já que para isso é necessário a inexistência de outros fluxos.

A rotas escolhidas pelo protocolo de roteamento vetor distância utilizam nessa topologia os enlaces de menor capacidade, porém com menos saltos. Desta maneira o enlace de maior capacidade não fica sobrecarregado, em contrapartida fluxos são limitados pelos enlaces de menor capacidade, gerando sobrecarga agora nesses enlaces.

Para fazer as medições da capacidade agregada da rede, usam-se fluxos UDP para inundar a rede. A banda dos fluxos é definida no emissor, e no receptor mede-se o quanto se recebe. Isso é feito através do software iperf. O objetivo do uso de fluxos UDP é para inundar a rede e obter o uso máximo dos recursos da rede, já que o UDP não tem controle de fluxo, além de ser possível definir a vazão dos fluxos, o que o TCP não permite.

Como o protocolo UDP não tem confirmação de recepção, as medidas do percentual da taxa recebida são medidas no iperf receptor. O iperf através do seqüenciamento dos pacotes vai contar quantos pacotes foram descartados na rede, devido ao enchimento das filas dos roteadores, e medir a percentagem de pacotes entregues.

Tabela V.5: Matriz de Tráfego RIP.

	R1	R2	R3	R4
R1	-	3.9 Mbps	2.5 Mbps	3.1 Mbps
R2	2.3 Mbps	-	**0 Mbps**	39.6 Mbps
R3	3.5 Mbps	3.5 Mbps	-	4.0 Mbps
R4	**1.4 Mbps**	41.3 Mbps	4.0 Mbps	-
Total	7.2 Mbps	48.7 Mbps	6.5 Mbps	46.7 Mbps

A Tabela V.5 mostra a matriz de tráfego efetivo dos fluxos entre os roteadores quando a rede é configurada com o protocolo de roteamento RIP. Verifica-se que os enlaces R3-R2 e R4-R1 ficam congestionados, e devido a isso os valores de transferência de R1 para R4

(1.4 Mbps) e de R3 para R2 (0 Mbps) ficam prejudicados.

Tabela V.6: Matriz de Tráfego OSPF.

	R1	R2	R3	R4
R1	-	3.9 Mbps	2.9 Mbps	3.9 Mbps
R2	3.9 Mbps	-	3.9 Mbps	**38.0 Mbps**
R3	3.0 Mbps	3.9 Mbps	-	4.0 Mbps
R4	3.9 Mbps	**33.8 Mbps**	3.9 Mbps	-
Total	10.8 Mbps	41.6 Mbps	10.7 Mbps	45.9 Mbps

A Tabela V.6 mostra a matriz de tráfego efetivo dos fluxos quando a rede é configurada pelo protocolo de roteamento OSPF. Observe que agora a transferência de R1 para R4, e de R3 para R2 não ficam degradas. O problema agora é a maneira como o OSPF concentra todos os fluxos no enlace R2-R4, degradando os fluxos de transferência de R2 para R4 (33.8 Mbps) e vice-versa (38 Mbps).

Tabela V.7: Matriz de Tráfego MPLS.

	R1	R2	R3	R4
R1	-	3.9 Mbps	2.7 Mbps	4.0 Mbps
R2	3.9 Mbps	-	4.0 Mbps	40.3 Mbps
R3	3.0 Mbps	4.0 Mbps	-	4.0 Mbps
R4	4.0 Mbps	40.7 Mbps	4.0 Mbps	-
Total	10.9 Mbps	48.6 Mbps	10.7 Mbps	48.3 Mbps

O MPLS tem a flexibilidade de configuração que permite exatamente o que nenhum dos dois protocolos de roteamento consegue, encontrar o meio termo. Enquanto o protocolo OSPF que se serve do mecanismo de estado de enlace congestiona a rota de maior capacidade, o RIP que usa o vetor distância congestiona a rota de menos saltos. Através do MPLS pode-se optar por uma diferenciação de serviço, solucionando o problema da priorização, e ao mesmo tempo solucionar o problema do congestionamento, com a engenharia de tráfego. O resultado é um alívio do enlace R2-R4, que possibilita o aumento da taxa de transferência entre os roteadores R2 e R4. E ao mesmo tempo não congestiona os enlaces de baixa capacidade R2-R3 e R1-R4, possibilitando que os fluxos entre esses roteadores também não sejam prejudicados. As capacidades agregadas de cada configuração da rede são mostradas na Tabela V.8:

Tabela V.8: Soma das capacidades agregadas.

Topologia	Capacidade da Rede
RIP	109.1 Mbps
OSPF	109.0 Mbps
MPLS	118.5 Mbps

É importante comentar os valores da vazão medida dos fluxos entre os roteadores R2 e R4. Apesar da demanda ter sido configurada para 60 Mbps, o software iperf do emissor não consegue transmitir a essa taxa. Já que o iperf receptor constata uma percentagem baixa de perda. Uma possível causa é a limitação da capacidade de processamento do emissor, não possibilitando a transmissão de dados a essa taxa. Mas como isso está associado a um problema de computação e não é derivado de alguma incapacidade da rede e nem de nenhum de seus componentes diretos, não foram realizados testes profundos e exaustivos visando o diagnóstico do problema.

Capítulo VI

Conclusão

O objetivo desse trabalho foi comparar o desempenho da comutação de pacotes MPLS com o roteamento IP. Para atingir este objetivo, foi implementada uma rede de testes que consistiu de quatro roteadores. A configuração lógica da rede, incluindo o protocolo de roteamento utilizado, foi modificada para poder comparar os diversos paradigmas de encaminhamento de pacotes.

O MPLS pode ser utilizado para fazer diferenciação de serviços, e não necessariamente precisa estar de acordo com a arquitetura *DiffServ*. Essa diferenciação é essencial para o trânsito de fluxos multimídias na Internet e para que a ela seja no futuro a rede que transportará todas as aplicações: telefonia, videoconferência, transferência de dados, TV digital, etc.

Para atingir esse objetivo o MPLS se mostra bastante eficaz, pois através dele é possível fazer diferenciação de serviços e balanceamento de carga simultaneamente. O redirecionamento dos fluxos mais prioritários para enlaces da rede que estão menos congestionados, ou até mesmo reservados, possibilitou o uso mais eficiente dos recursos da rede, aumentando sua capacidade agregada. Ao mesmo tempo, esse tratamento diferenciado possibilitou a garantia da qualidade de serviço, já que esses recursos são usados por um número restrito de usuários, devido ao Controle de Admissão. É importante comentar que não é recomendado que todos os fluxos desses usuários prioritários sejam encaminhados para esses recursos reservados, pois pode acarretar em degradação da QoS de outros fluxos.

Teoricamente, a implementação do MPLS tornaria o encaminhamento do pacote mais eficiente e menos custoso computacionalmente para o roteador. A prática não mostra isso no ambiente de testes desenvolvido. A implementação do mecanismo de encaminhamento do MPLS no kernel ainda não otimizada, segundo o próprio desenvolvedor do mecanismo inserido no kernel [26], a pouca quantidade de roteadores de núcleo do domínio MPLS do ambiente de teste feito, e a arquitetura de hardware dos roteadores PC que não são aperfeiçoadas para o roteamento de pacotes são possíveis causas dessa ineficiência. Porém é importante frisar que estas conclusões se aplicam somente a esses resultados particulares do ambiente de testes criado que é bem particular. Não podemos tirar nenhuma conclusão abrangente ou generalizada sobre o desempenho do MPLS em redes com outras topologias, que sejam constituidas de roteadores com outras arquiteturas de hardware.

O pequeno número de roteadores de núcleo do domínio MPLS acarreta em valores de atraso e *jitter* grandes devido à sobrecarga computacional da borda do domínio que não é compensada no núcleo. A vantagem do MPLS sobre o IP se dá em uma topologia com muitos saltos no núcleo do domínio, o que não ocorre na topologia proposta, assim, o MPLS não obtém desempenho superior ao roteamento IP nessa topologia, o que pode ser observado nos valores de atraso, medidos com o `ping`.

As medidas do *jitter* mostraram que mesmo com a rede congestionada, o MPLS é capaz de mantê-lo em valores inferiores aos valores do roteamento IP. Isso se deve, pois no roteamento IP os enlaces escolhidos ficam congestionados, e com o MPLS a diferenciação é realizada escolhendo caminhos alternativos.

Nas medidas da capacidade agregada da rede, o MPLS satisfaz os requisitos de engenharia de tráfego, que o RIP e o OSPF não são capazes de satisfazer. O RIP exagera na quantidade de fluxos desviados do enlace de maior capacidade, tornando os enlaces secundários congestionados. O OSPF faz o oposto, encaminha os fluxos para o enlace de maior capacidade, congestionando este. O MPLS devido a sua flexibilidade de configuração possibilita que somente parte do fluxo de um mesmo par origem-destino seja encaminhado através de um caminho alternativo. Assim, é possível utilizar recursos ociosos, sem congestioná-los, e ao mesmo tempo aliviar a carga do enlace principal possibilitando que outros fluxos o utilizem.

Os resultados obtidos neste projeto indicam que o MPLS é uma alternativa viável e eficaz para controle de tráfego de uma rede, justificando o seu uso nas redes óticas de altíssima velocidade que deverão ser usadas nos próximos anos.

Referências Bibliográficas

[1] D. D. Clark, "The Design Philosophy of the DARPA Internet Protocols", *ACM Sigcomm*, 1988.

[2] A. S. Tanenbaum, *Computer Networks.* Prentice Hall Inc, fourth ed., 2002.

[3] A. Mankin, E. F. Baker, B. Braden, S. Bradner, M. O'Dell, A. Romanow, A. Weinrib e Zhang, "Resource ReSerVation Protocol (RSVP) – Version 1 Applicability Statement Some Guidelines on Deployment", *Internet RFC 2208*, setembro de 1997.

[4] S. Keshav, *An Engineering Approach to Computer Networking: ATM Networks, the Internet, and the Telephone Network.* Addison-Wesley, first ed., 1997.

[5] J. Wroclawski, "The Use of RSVP with IETF Integrated Services", *Internet RFC 2210*, setembro de 1997.

[6] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang e W. Weiss, "An Architecture for Differentiated Service", *Internet RFC 2475*, dezembro de 1998.

[7] K. Nichols, S. Blake, F. Baker e D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", *Internet RFC 2474*, dezembro de 1998.

[8] B. Davie, A. Charny, J. Bennet, K. Benson, J. L. Boudec, W. Courtney, S. Davari, V. Firoiu e D. Stiliadis, "An Expedited Forwarding PHB (Per-Hop Behavior)", *Internet RFC 3246*, março de 2002.

[9] J. Heinanen, F. Baker, W. Weiss e J. Wroclawski, "Assured Forwarding PHB Group", *Internet RFC 2597*, junho de 1999.

[10] B. Bhaniramk, P. Sun e R. Jain, "Quality of Service using traffic engineering over MPLS: an analysis", *IEEE Conference on Local Computer Networks*, vol. 1, no. 25, pp. 238–241, novembro de 2000.

[11] E. Rosen, A. Viswanathan e R. Callon, "Multiprotocol Label Switching Architecture", *Internet RFC 3031*, janeiro de 2001.

[12] E. Rosen, D. Tappan, G. Fedorkow, Y. Rekhter, D. Farinacci, T. Li e A. Conta, "MPLS Label Stack Encoding", *Internet RFC 3032*, janeiro de 2001.

[13] L. Andersson, P. Doolan, N. Feldman, A. Fredette e B. Thomas, "LDP Specification", *Internet RFC 3026*, janeiro de 2001.

[14] B. Jamoussi, E. L. Andersson, R. Callon, R. D. L. Wu, P. Doolan, T. Worster, N. Feldman, A. Fredette, M. Girish, E. Gray, J. Heinanen, T. Kilty e A. Malis, "Constraint-Based LSP Setup using LDP", *Internet RFC 3012*, janeiro de 2002.

[15] *Linux Kernel*. http://www.kernel.org/, 2003.

[16] *MPLS-Linux Project*. http://mpls-linux.sourceforge.net/, 2003.

[17] *Source Forge*. http://sourceforge.net/, 2003.

[18] *Zebra Daemon*. http://www.zebra.org/, 2003.

[19] *Iptables*. http://netfilter.samba.org/, 2003.

[20] *IPRoute2*. http://www.opencontent.org/openpub/, 2003.

[21] *IPerf*. http://dast.nlanr.net/Projects/Iperf/, 2003.

[22] H. Schulzrinne, S. Casner, R. Frederick e V. Jacobson, "RTP: A Transport Protocol for Real-Time Application", *Internet RFC 1889*, janeiro de 1996.

[23] G. Malkin, "RIP Version 2", *Internet RFC 2453*, novembro de 1998.

[24] J. Moy, "OSPF Version 2", *Internet RFC 2328*, abril de 1998.

[25] S. Keshav e R. Sharma, "Issues and trends in router design", *IEEE Communications Magazine*, no. 1, pp. 144–151, maio de 1998.

[26] *MPLS Linux List*. http://sourceforge.net/mailarchive/forum.php?forum=mpls-linux-general, 2003.