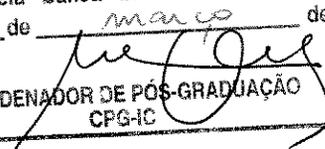


Este exemplar corresponde à redação final da
Tese/Dissertação devidamente corrigida e defendida
por: Gustavo Bittencourt
Figueiredo
e aprovada pela Banca Examinadora.
Campinas, 30 de março de 04

COORDENADOR DE PÓS-GRADUAÇÃO
CPG-IC

49646,000

**Algoritmos de Roteamento com Interferência
Mínima**
Gustavo Bittencourt Figueiredo
Dissertação de Mestrado

UNICAMP
BIBLIOTECA CENTRAL
SEÇÃO CIRCULANTE

Algoritmos de Roteamento com Interferência Mínima

Gustavo Bittencourt Figueiredo

Dezembro de 2003

Banca Examinadora:

- Prof. Dr. Nelson Luís Saldanha da Fonseca (Orientador)
- Prof. Dr. Maurício F. Magalhães
FEEC/UNICAMP
- Prof. Dr. Edmundo R. M. Madeira
IC/UNICAMP
- Prof. Dr. João C. Setubal
IC/UNICAMP

UNIDADE BC
Nº CHAMADA T UNICAMP
F 469a
V _____ EX _____
TOMBO BC/ 58115
PROC 16. 117-04
C _____ D λ
PREÇO R\$ 11,00
DATA 08/05/04
Nº CPD _____

CM00197B36-3

BIB ID 316772

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP**

Figueiredo, Gustavo Bittencourt

F469a Algoritmos de roteamento com interferência mínima/Gustavo
Bittencourt Figueiredo -- Campinas, [S.P. :s.n.], 2003.

Orientador : Nelson Luis Saldanha da Fonseca.

Co-orientador: José Augusto Suruagy Monteiro.

Dissertação (Mestrado) - Universidade Estadual de Campinas,
Instituto de Computação.

1.Redes de computação - Protocolos. 2.Internet(Redes de
computação) - Tecnologia. 3.Telecomunicações - Tráfego. I. Fonseca,
Nelson Luis Saldanha da. II. Monteiro, José Augusto Suruagy. III.
Universidade Estadual de Campinas. Instituto de Computação. IV.
Título.

Algoritmos de Roteamento com Interferência Mínima

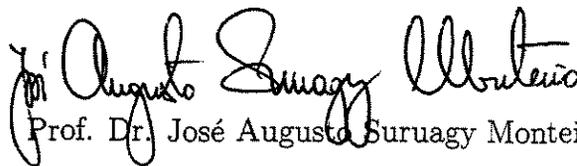
Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Gustavo Bittencourt Figueiredo e aprovada pela Banca Examinadora.

Campinas, 15 de dezembro de 2003.



Prof. Dr. Nelson Luís Saldanha da Fonseca

(Orientador)



Prof. Dr. José Augusto Suruagy Monteiro

(Co-orientador)

Dissertação apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

Substitua pela folha com a assinatura da banca

© Gustavo Bittencourt Figueiredo, 2003.

Todos os direitos reservados.

Resumo

Um dos principais desafios relacionados à Engenharia de Tráfego em redes MPLS (*MultiProtocol Label Switching*) é a definição das rotas para os *Label Switched Paths* - LSPs. Algoritmos de roteamento baseados na computação do caminho mais curto tendem a concentrar o tráfego nestes caminhos, ocasionando congestionamentos e conseqüente degradação dos serviços oferecidos pela rede. Assim sendo, algoritmos de roteamento com interferência mínima vêm sendo propostos para superar estas deficiências.

A idéia central dos algoritmos de roteamento com interferência mínima é que a redução da probabilidade de bloqueio de solicitações de estabelecimento de LSPs é conseqüência da minimização da redução do fluxo máximo total da rede. Desta forma, ao rotear uma conexão entre um par origem-destino, os algoritmos de interferência mínima tentam minimizar a redução do fluxo máximo entre os outros pares.

Nesta dissertação, um novo algoritmo de roteamento com interferência mínima, denominado *Light Minimum Interference Routing* - LMIR, é proposto. O algoritmo LMIR possui menor custo computacional e precisão similar quando comparado a outros algoritmos de interferência mínima propostos na literatura, conforme verificado através de experimentos de simulações realizados utilizando redes de pequeno e grande porte.

1

Abstract

Routing is one of the major challenges in MPLS network traffic engineering. Shortest path based routing algorithms may cause unbalanced utilization, generating congestion and service degradation. Therefore, minimum interference routing algorithms have been proposed to overcome such deficiency.

The central idea behind these algorithms is that minimizing the maximum network flow reduction a smaller blocking probability value is obtained. Thus, when routing a connection request between a source-destination pair, the minimum interference routing algorithm is trying to minimize the maximum network flow reduction between the other pairs.

In this dissertation, a new minimum interference routing algorithm, called *Light Minimum Interference Routing* (LMIR), is proposed. The algorithm has a smaller computational cost and similar accuracy to the other existing algorithms as verified in simulations using both small and large networks.

Agradecimentos

Em primeiro lugar eu gostaria de agradecer a Deus, que é o detentor de todas as honras. Gostaria de agradecer por todas as oportunidades e realizações que me tem concedido e pela conclusão desta dissertação.

Gostaria de agradecer também à minha mãe, “Dona” Zane e ao meu pai, “Seu” Fernando, por toda a dedicação e pelo amor irrestrito que sempre me dedicaram. Aos meus irmãos, Messias, Mércia, Nando, Sandra, Cal, Tiago e Rafa pelo apoio e carinho. Pela ótima convivência que sempre tivemos e pela força em todos os momentos de necessidade (Vcs sabem o quanto são importantes, não?? :-)).

À Cristina que tem sido minha grande companheira em todos esses anos e ao nosso filhinho, Gutinho, que é um presente precioso!!

Ainda na seção família, agradeço à tia Bebel e meu tio Juca e os primos Rogério, Nem, Nadir, Gisele, Neuri, Kati etc...pela família que conseguimos construir!

Gostaria de deixar um agradecimento especial ao meu tio Fio e minha vó Ju!!

Gostaria de agradecer também aos meus orientadores Nelson e Suruagy pela amizade e franqueza de todas as horas e pelas suas tantas contribuições ao meu desenvolvimento profissional!!!

Aos amigos da república dos pistolinhas: Power, o Guido (Dawson), Triste Toy Boy (Alceu Valença) e Zeh pela convivência harmoniosa e pelas discussões acaloradas por horas a fio.

Aos amigos do IC, Bazinho, Borin, Bartho, César, Chenca, Cléo, Eduardo (Pará), Flavim, Fernando, Fábio, Glauber, Juliana, Juliana, Lasaro, Luiz Gomes, Marília, Mi-

chele, Rogério, Salgadinho, Silvana, Wesley, que tornaram a estadia em Campinas mais divertida.

Aos amigos do fim-de-linha Tito, Alex, Paulinho, Alexsandro e todos os outros de C. Branco. Os que não consegui lembrar neste momento, por favor, me perdoem...e apareçam mais..assim na próxima não ficarão de fora!! :-)

Por fim, agradeço ao CNPq pelo apoio financeiro.

Sumário

Resumo	ix
Abstract	xi
Agradecimentos	xiii
1 Introdução	1
2 Arquitetura do MPLS	9
2.1 Histórico	9
2.2 Conceitos Básicos	12
2.2.1 Componentes das redes MPLS	13
2.3 Rótulos	14
2.3.1 Codificação dos Rótulos	14
2.3.2 Rótulos inválidos	16
2.4 LSP's	17
2.4.1 Túneis LSP	18
2.4.2 Roteamento de LSPs	20
2.5 Protocolos para Distribuição de Rótulos	20
2.6 Considerações Finais	22
3 Engenharia de Tráfego	23
3.1 Introdução	23

3.2	Modelo do processo de Engenharia de Tráfego	24
3.3	Taxonomia	28
3.4	Políticas de controle de congestionamento	30
3.5	MPLS e Engenharia de Tráfego em redes IP	32
3.6	Considerações Finais	36
4	Roteamento com interferência mínima	37
4.1	Fluxos em Redes	38
4.1.1	O método de Ford-Fulkerson	39
4.1.2	Redes residuais	40
4.1.3	Caminhos Aumentantes	40
4.1.4	Cortes de fluxo em redes	40
4.2	Algoritmos de roteamento com interferência mínima	41
4.3	Notação e definição do problema	42
4.4	O algoritmo <i>MIRA</i>	43
4.5	O algoritmo de Wang, Su, e Chen (WSC)	45
4.6	O algoritmo <i>Light Minimum Interference Routing</i>	48
4.7	Limite superior para K	53
5	Comparação Entre os Algoritmos de Roteamento com Interferência Mínima	59
5.1	Validação através da comparação com os Resultados Encontrados na Lite- ratura	60
5.2	Resultados para redes de grande porte	65
5.2.1	Redes densas de 30 vértices	67
5.2.2	Redes esparsas de 30 vértices	69
5.2.3	Redes densas com 40 vértices	71
5.2.4	Redes esparsas de 40 vértices	74
5.2.5	Redes densas com 50 vértices	75
5.2.6	Redes esparsas com 50 vértices	78

5.3	Limitante superior de K nos exemplos numéricos	79
5.4	Considerações finais	84
6	Um mecanismo para dimensionamento dinâmico de LSPs	85
6.1	Banda Passante Efetiva	85
6.2	Redimensionamento dinâmico de LSPs	87
6.3	Experimentos de simulação	90
6.4	Considerações Finais	92
7	Conclusões	97
7.1	Trabalhos Futuros	98
	Referências Bibliográficas	99

Lista de Tabelas

5.1	LMIR: tempo de execução para 8,000 requisições.	64
5.2	Parâmetros usados na geração aleatória dos grafos.	66
5.3	Limitante superior $K(\theta)$	81
5.4	Tempo de execução para 30,000 requisições	82
5.5	Ganho relativo no tempo de execução para 30,000 requisições	82
5.6	Tempo médio de execução para conexões individuais	83
5.7	Ganho relativo no tempo de execução para conexões individuais	83
6.1	Valores dos limiares	92

1

Lista de Figuras

2.1	Componentes de uma rede MPLS	13
2.2	Rótulos	14
2.3	Exemplo de problema causado por rótulo inválido	17
2.4	Túneis LSP	19
3.1	Modelo do processo de Engenharia de Tráfego, adaptado de [Awd99]	26
3.2	Congestionamento causado pela concentração de tráfego no caminho mais curto.	33
4.1	Interferência: exemplo	42
4.2	Rede com vértice concentrador	46
4.3	Quantidade de caminhos disjuntos menor que a quantidade de arestas no corte mínimo	55
4.4	Transformação na rede original	55
4.5	Rede da Figura 4.3 transformada	55
5.1	Topologia usada nas simulações	60
5.2	Largura de banda total entre todos os pares origem-destino da rede.	62
5.3	Número de rejeições.	62
5.4	Interferência sofrida pelo par S1-D1.	63
5.5	Influência do parâmetro K no número de conexões rejeitadas.	65
5.6	Largura de banda total (redes densas de 30 vértices).	67

5.7	Largura de banda total (Vista ampliada).	68
5.8	Número de requisições rejeitadas (redes densas de 30 vértices).	68
5.9	Interferência sofrida pelo par S1-D1 (redes densas de 30 vértices).	69
5.10	Número de perdas em função de K (redes densas de 30 vértices).	69
5.11	Largura de banda total (redes esparsas de 30 vértices).	70
5.12	Largura de banda total (redes esparsas de 30 vértices).	70
5.13	Número de requisições rejeitadas (redes esparsas de 30 vértices).	71
5.14	Interferência sofrida pelo par S1-D1 (redes esparsas de 30 vértices).	71
5.15	Número de perdas em função de K (redes esparsas de 30 vértices).	72
5.16	Largura de banda total (redes densas de 40 vértices).	72
5.17	Número de requisições rejeitadas (redes densas de 40 vértices).	73
5.18	Interferência sofrida pelo par S1-D1 (redes densas de 40 vértices).	73
5.19	Número de perdas em função de K (redes densas de 40 vértices).	74
5.20	Largura de banda total (redes esparsas de 40 vértices).	74
5.21	Número de requisições rejeitadas (redes esparsas de 40 vértices).	75
5.22	Interferência sofrida pelo par S1-D1 (redes esparsas de 40 vértices).	76
5.23	Número de perdas em função de K (redes esparsas de 40 vértices).	76
5.24	Largura de banda total (redes densas de 50 vértices).	77
5.25	Número de requisições rejeitadas (redes densas de 50 vértices).	77
5.26	Interferência sofrida pelo par S1-D1 (redes densas de 50 vértices).	78
5.27	Número de perdas em função de K (redes densas de 50 vértices).	78
5.28	Largura de banda total (redes esparsas de 50 vértices).	79
5.29	Número de requisições rejeitadas (redes esparsas de 50 vértices).	80
5.30	Interferência sofrida pelo par S1-D1 (redes esparsas de 50 vértices).	80
5.31	Número de perdas em função de K (redes esparsas de 50 vértices).	81
6.1	Mecanismo de redimensionamento dinâmico de LSPs	89
6.2	Modelo utilizado na simulação	90
6.3	Quantidade total de perdas	93

6.4	Taxa de perdas experimentada pelo LSP	93
6.5	Comprimento médio do buffer	94
6.6	Utilização média	94

Capítulo 1

Introdução

Há muito, percebe-se a necessidade de uma rede integrada que ofereça serviço de transmissão de voz, vídeo e dados numa mesma estrutura física. A mais importante iniciativa nesse sentido foi a ATM (*Asynchronous Transfer Mode*)[SLC95][Tan97]. Ela foi criada, ao final dos anos 80, com a finalidade de ser a tecnologia de transmissão das Redes Digitais de Serviços Integrados de Faixa Larga (RDSI-FL) e desde sua concepção já previa a incorporação de mecanismos capazes de atender aos requisitos de Qualidade de Serviço (QoS) de aplicações dos mais diversos tipos de mídia.

Apesar de ser uma rede que opera a altas taxas de transmissão e ter todo seu projeto baseado no suporte a aplicações sensíveis a perdas ou a atrasos, além de possuir mecanismos eficientes de controle de admissão e controle de congestionamento, o ATM nunca teve uma utilização em larga escala. A ausência de esquemas eficientes que compatibilizassem o ATM com a pilha de protocolos TCP/IP, ou seja, com a Internet, foi responsável pela sua não popularização.

Por outro lado, a Internet experimentou, nos últimos anos, um crescimento vertiginoso, tornando-se parte permanente do cotidiano da sociedade. O crescimento do número de usuários e a facilidade de comunicação propiciada pela Internet, motivou a utilização de aplicações pertinentes a várias áreas do conhecimento humano. Surge, assim, uma nova classe de aplicações que oferece novas oportunidades e formas de comunicação, novas

formas de cooperação de dados e idéias.

Tais aplicações podem envolver um grande número de mídias, podem ser interativas e manipularem, em tempo real, grandes quantidades de dados. Aplicações de *Data Mining*, por exemplo, podem requerer troca de grandes quantidades de dados entre *sites* remotos. Vídeo Conferência e TV de alta definição (*HDTV*) que são aplicações de alta qualidade de visualização requerem largura de banda garantida e possuem requisitos temporais estritos para a entrega de pacotes. Outras aplicações avançadas requerem altas taxas de transmissão e parâmetros limitados de *jitter*, atraso e perdas.

Entretanto, o serviço oferecido na Internet atualmente é do tipo melhor-esforço, o que significa que o tráfego do usuário será encaminhado sem nenhuma garantia mínima de largura de banda, taxa de perdas, atraso, *jitter* e, obviamente, sem nenhum tipo de prioridade. A transformação da Internet em uma rede capaz de suportar satisfatoriamente tráfego sensível a perdas e atrasos passa, necessariamente, pela adoção de mecanismos de provisionamento de Qualidade de Serviço (QoS) e pela utilização balanceada dos recursos da rede. Esta última provoca a eliminação dos pontos críticos, onde a ocorrência da degradação do serviço é mais provável, e propicia a acomodação de um número maior de usuários cujas necessidades de tráfego podem ser atendidas.

Diferentes esquemas têm sido propostos para o provisionamento de QoS na Internet. Estes esquemas seguem em duas vertentes principais, a dos serviços integrados *IntServ* [BCS94] e a dos serviços diferenciados *DiffServ* [A198]. Na primeira, a proposta é introduzir os princípios das redes baseadas em comutação de circuitos virtuais ao modelo sem conexão das redes IP, estabelecendo “canais” para fluxos individuais.

O *IntServ* foi a primeira proposta para garantir QoS na Internet. Ele adiciona classes de serviços ao modelo de melhor-esforço da Internet para atender às necessidades de QoS das aplicações. Duas classes foram propostas: serviço garantido (*Guaranteed Service*) e serviço de carga controlada (*Controlled Load Service*). Dependendo da aplicação, essas classes de serviço propiciam limites mais estreitos ou tolerantes nos controles de QoS.

O serviço garantido foi criado para atender a aplicações com requisitos explícitos de

transmissão, como garantias de banda passante, limites para atraso fim-a-fim e ausência de perdas nas filas para fluxos que estejam em conformidade com as especificações deste fluxo.

O serviço de carga controlada não provê garantias tão estritas de QoS como o serviço garantido. Ao contrário, ele é um serviço para aplicações que necessitam de um serviço mais confiável e eficiente que o melhor-esforço e que suportam pequenos atrasos e perdas.

Na arquitetura *IntServ*, as aplicações devem fazer reserva dos recursos através de um protocolo de sinalização, o RSVP. Os roteadores fazem controle do tráfego para garantir às aplicações que suas necessidades serão atendidas.

Diferentemente da arquitetura *IntServ*, o *DiffServ* não requer que as aplicações façam reservas de recursos. As garantias de Qualidade de Serviço nos roteadores são estáticas e são representadas por classes de serviço. Os fluxos são agregados de acordo com os seus requisitos de QoS e cada agregado recebe tratamento diferenciado de acordo com a classe de serviço a que pertence.

O *DiffServ* foi desenvolvido para ser escalável. Por basear-se na agregação de fluxos em um número fixo de classes de serviço para provisão de diferentes níveis de QoS, o *DiffServ* não precisa manter informações individuais de estado para cada fluxo. Assim, o sistema não se degrada com o aumento do número de fluxos na rede.

Os serviços definidos inicialmente para a arquitetura *DiffServ* são o serviço assegurado (*AF - Assured Forwarding*) e o serviço de encaminhamento expresso (*EF - Expedited Forwarding*). O serviço AF não dá garantias de banda passante, mas oferece garantias de que o tráfego será encaminhado com confiabilidade e alta prioridade, mesmo na presença de congestionamento. Ele é composto por classes distintas de serviço, as quais possuem sub-classes com diferentes níveis de precedência em relação ao descarte de pacotes.

O serviço de encaminhamento expresso é indicado às aplicações de tempo-real. Ele oferece um serviço de rede com baixa perda, pequeno atraso, baixo *jitter* e banda passante garantida.

A utilização de mecanismos de provisionamento de QoS na Internet não é condição

suficiente para o perfeito funcionamento das aplicações. Congestionamentos e a consequente perda de pacotes ainda são possíveis de existir se houver uma má distribuição de tráfego nos enlaces da rede. Ademais, atualmente está em voga a utilização de fibras ópticas como meios de transmissão de enlaces, associadas a comutadores também ópticos em conjunto com técnicas de multiplexação WDM (*Wavelength Division Multiplexing*) ou DWDM (*Dense Wavelength Division Multiplexing*). Isto sugere a possibilidade de que, no futuro, a banda-passante será tão abundante e barata que a QoS será provida adicionando mais banda aos enlaces.

Entretanto, mesmo com a abundância de banda-passante, a má distribuição de tráfego pode levar ao comprometimento da QoS. Este fato indica que o provisionamento efetivo de QoS na Internet só poderá ser alcançado se acompanhado de técnicas de Engenharia de Tráfego.

A Engenharia de Tráfego (TE) está ligada à otimização do desempenho da rede e a utilização eficiente dos seus recursos. Segundo Awduche et.al. [Awd99] [Al.02a], a Engenharia de Tráfego na Internet engloba tecnologia e aplicação de princípios científicos para medir, modelar, caracterizar e controlar o tráfego Internet, de forma a facilitar a operacionalidade e resultar em uma utilização balanceada dos recursos da rede.

Como pode-se perceber, a ocorrência de congestionamentos é um problema crucial das redes, especialmente quando se pretende oferecer serviços de rede com garantias de QoS. A utilização balanceada dos recursos, além da utilização de mecanismos usuais de provisionamento de QoS, é um passo fundamental nesta direção. É a Engenharia de Tráfego que é responsável pelo mapeamento adequado dos fluxos na estrutura física da rede.

O MPLS (*MultiProtocol Label Switching*) tem sido apontado como uma ferramenta de suma importância para a Engenharia de Tráfego na Internet. A principal motivação para a utilização de MPLS na Engenharia de Tráfego vem das próprias limitações dos protocolos utilizados na Internet. Os protocolos de roteamento utilizados fundamentam-se em algoritmos de menor caminho (*Shortest Path First*) sem restrições. Esses algoritmos

utilizam como métrica o número de saltos. Dessa forma, não havendo mudança de topologia, todo o tráfego entre um par de vértices será encaminhado pelo mesmo caminho, independente das condições dos enlaces ou restrições administrativas.

Além disso, o modelo de encaminhamento de datagramas empregado pelo protocolo IP é baseado apenas no endereço de destino contido nos pacotes, sem indicação de precedência dos mesmos, ou mesmo sem nenhum tipo de prioridade. Em redes com aplicações com requisitos distintos de QoS, é desejável que os pacotes indiquem estas tais características sem nenhum *overhead* adicional. A criação de LSPs (*Label Switched Paths*) em redes MPLS entre determinados pares de roteadores pode solucionar este problema associando cada LSP a uma classe de serviço e a cada classe de serviço uma prioridade.

Entre os principais desafios da Engenharia de Tráfego usando MPLS estão o roteamento baseado em restrições, dimensionamento dos LSPs, particionamento do tráfego em múltiplos LSPs, definição da topologia inicial da rede e políticas de gerência que integrem os diversos mecanismos citados anteriormente [OSAU02][A1.02b].

Uma vez estabelecidos os LSPs, às vezes, o particionamento do tráfego entre múltiplos LSPs entre o mesmo par de roteadores é uma maneira conveniente de diminuir a carga ou atraso em um deles sem o custo de sinalização adicional, já que a única tarefa envolvida é a troca dos rótulos dos pacotes. Porém, o particionamento por pacotes pode levar a uma reordenação de pacotes muito grande no roteador destino. Assim, mecanismos de particionamento por fluxos muitas vezes são soluções alternativas para o problema.

O dimensionamento adequado dos LSPs é necessário para reduzir a quantidade dos mesmos na rede, e conseqüentemente o custo de sinalização envolvido. Além disso, oferecimento de QoS e utilização ótima dos recursos da rede só podem ser alcançados se os LSPs tiverem o dimensionamento correto. Caso contrário, podem ocorrer perda de pacotes (se os LSPs tiverem capacidade inferior à demandada) ou requisições por estabelecimento de LSPs podem ser rejeitadas caso alguns LSPs utilizem recursos acima do necessário.

A utilização das técnicas de Engenharia de Tráfego citadas anteriormente, muitas vezes envolve um custo de sinalização alto, sendo, porém, mais apropriadas ao estado

corrente da rede. Logo, a adoção de uma política “inteligente” que indique o tipo de técnica mais adequada a ser empregada como uma função dos custos de sinalização, custos operacionais ou outra restrição é, sem dúvida, de suma importância para que os objetivos da Engenharia de Tráfego sejam alcançados.

O roteamento baseado em restrições (*Constrained Based Routing*), por sua vez, é responsável pela seleção dos caminhos por onde os LSPs serão estabelecidos. Estes caminhos são considerados ótimos se satisfazem um conjunto de restrições e requisitos. As métricas utilizadas na escolha dos caminhos podem ser desde custo financeiro até o atraso fim-a-fim experimentado pelos pacotes.

Uma classe particular de algoritmos tem ganho importância no contexto de roteamento baseado em restrições. São os algoritmos de roteamento com interferência mínima. Estes algoritmos utilizam como restrição a diminuição do fluxo máximo entre outros roteadores selecionados como pontos de entrada e saída de tráfego da rede. Dessa forma, espera-se que quanto menor for a diminuição do fluxo máximo provocada, menor será a probabilidade de bloqueio.

No entanto, um problema desses algoritmos é o elevado custo computacional. Isto acontece pois os mesmos realizam, a cada requisição de conexão, a identificação de arestas que devem ser evitadas através da execução de algoritmos para cálculo do fluxo máximo, o que pode consumir um tempo proibitivo.

Nesta dissertação é apresentado um novo algoritmo de interferência mínima, denominado *Light Minimum Interference Routing - LMIR*, que realiza a identificação de arestas que devem ser evitadas ao se estabelecer um LSP, através de uma versão modificada do algoritmo de *Dijkstra*. Dessa forma, o algoritmo LMIR alcança precisão similar aos demais, porém com um custo computacional mais baixo. Além do algoritmo de roteamento com interferência mínima, é apresentado também um mecanismo de dimensionamento dinâmico dos LSPs, capaz de identificar o instante em que os LSPs já não correspondem à demanda de tráfego para os mesmos.

Nesta dissertação serão apresentados os conceitos básicos da arquitetura do MPLS

bem como da Engenharia de Tráfego. Será ainda apresentado um algoritmo de roteamento com interferência mínima com complexidade computacional menor que a dos outros algoritmos de roteamento com interferência mínima. Resultados obtidos através de simulações mostram que o algoritmo apresenta precisão similar aos demais algoritmos e tempo de execução inferior. É introduzido também um mecanismo de dimensionamento dinâmico de LSPs cuja idéia central é fornecer uma indicação de quando deve ser realizado o processo de atualização de capacidades dos LSPs, se necessário.

Este trabalho está organizado como segue: o capítulo 2 apresenta os conceitos básicos relacionados à arquitetura do MPLS, no capítulo 3 serão apresentados os conceitos de engenharia de tráfego de redes e sua aplicação com o MPLS. Algoritmos de roteamento com interferência mínima serão apresentados e discutidos no capítulo 4. Resultados obtidos através de simulação comparando os algoritmos de interferência mínima apresentados serão discutidos no capítulo 5. O capítulo 6 apresenta um mecanismo de dimensionamento dinâmico de LSPs e avaliação de seu desempenho. As conclusões e indicações de trabalhos futuros serão apresentadas no capítulo 7.

Capítulo 2

Arquitetura do MPLS

Neste capítulo a arquitetura do MPLS é apresentada. Será realizada uma breve discussão sobre o histórico da tecnologia e em seguida serão apresentados os conceitos básicos, principais características e modo de funcionamento do MPLS.

2.1 Histórico

Tradicionalmente, sistemas de comunicação foram desenvolvidos para o transporte de tipos específicos de informação. Por exemplo, o sistema telefônico foi desenvolvido para o tráfego de voz ao passo que as redes de comutação de pacotes foram desenvolvidas para dados textuais. Essas redes adaptavam-se mal a outros tipos de dados diferentes daqueles a que foram inicialmente projetadas. A idéia de uma única rede capaz de oferecer todos esses serviços motivou o conceito das Redes Digitais de Serviços Integrados (RDSI).

O ATM (*Asynchronous Transfer Mode*) foi a tecnologia escolhida pelo *Broad Band Task Group* do ITU-T para a implementação das RDSI-FL (Redes Digitais de Serviços Integrados de Faixa Larga) [SLC95]. As RDSI-FL foram projetadas para oferecer serviços como vídeo sob-demanda, música com qualidade de CD, aplicações de uso científico com requisitos rígidos de QoS dentre outros.

O ATM foi projetado, especificamente, para oferecer os elementos necessários à inte-

gração dos serviços em uma única estrutura física. Altas taxas de transmissão, garantias de QoS e facilidades de gerenciamento são algumas de suas características que permitem a integração dos vários serviços em uma única rede. Para oferecer os vários serviços com tráfegos de características distintas, o ATM implementa esquemas de diferenciação de fluxos, propostos pelo ITU-T e ATM Fórum, chamados de classes de serviço. Por exemplo, o tráfego de vídeo é muito sensível a atrasos, então pode-se utilizar a classe de serviço CBR (Constant Bit Rate) de forma a minimizar o atraso entre as chegadas de células, que são as unidades de transmissão das redes ATM.

Apesar de toda flexibilidade e robustez do ATM e da possibilidade de um completo gerenciamento da rede, ele é uma tecnologia que não tem se disseminado como esperado. O aspecto que contribui de forma decisiva para esta pouca disseminação é a grande popularização da Internet, e conseqüentemente, do protocolo IP.

A maioria das aplicações utilizadas nas redes de computadores utiliza o protocolo IP. Portanto, seria inviável adaptar todas as aplicações já em funcionamento, em função do ATM. Embora existam maneiras já padronizadas que propiciam a interoperabilidade do IP com o ATM: o CLIP - *Classical IP* e LANE (*LAN Emulation*), estas soluções fazem com que muitos atributos da rede ATM não sejam utilizados, além de tornar a estrutura resultante demasiado complexa.

Apesar da sua grande popularização, o protocolo IP não oferece a estrutura necessária ao transporte do tráfego das aplicações emergentes. O serviço oferecido é do tipo melhor esforço, onde todas as aplicações têm a mesma prioridade. Além disso, as redes IP utilizam o encaminhamento por saltos (*hop-by-hop*) baseado no endereço de destino. Ou seja, cada pacote tem o endereço de destino contido no seu cabeçalho IP examinado em cada roteador visitado da rede.

Este processo é muito custoso em roteadores do núcleo da rede dado o tamanho das tabelas de roteamento destes equipamentos. Além disso, a agregação de endereços causada pelo CIDR (*Classless Inter-Domain Routing*) onde o prefixo de rede tem tamanho variável aumenta ainda mais o tempo de busca nas tabelas de roteamento já que a busca deve ser

pelo prefixo de rede mais longo que coincida com o endereço de destino, o que aumenta consideravelmente a complexidade da mesma.

A complexidade do processo de encaminhamento do protocolo IP aumenta consideravelmente o tempo de encaminhamento e conseqüentemente o atraso fim-a-fim. Além disso, o *jitter* também aumenta já que os endereços têm tamanho variável permitindo uma variação do tempo de encaminhamento em cada salto.

Assim, a idéia de comutação IP surgiu com a necessidade de aumentar a eficácia das redes IP, comportar o crescimento contínuo da Internet e suportar o surgimento de novas aplicações [MC01]. Pensando nisso, algumas iniciativas que visavam a introdução da comutação por rótulos em redes IP começaram a surgir a partir da década de 90. Estas tecnologias visavam integrar as vantagens da comutação por rótulos à família de protocolos IP, tornando-a mais adequada ao suporte das aplicações emergentes.

As principais tecnologias de comutação de rótulos eram o CSR *Cell Switched Router* da Toshiba, *Tag Switching* da CISCO, ARIS (*Aggregate Route-Based IP Switching*) da IBM, *IP Switching* da *Ipsilon Networks*. Estas tecnologias utilizavam o *hardware* dos comutadores ATM devido à sua velocidade de comutação, mas não os seus protocolos de sinalização. Os protocolos de sinalização utilizados eram proprietários. Por exemplo, o *IP Switching* utilizava o IFMP (*Ipsilon Flow Management Protocol*), a *Tag Switching* utilizava o TDP (*Tag Distribution Protocol*), etc. [DR00].

Por serem tecnologias proprietárias, não havia interoperabilidade entre as tecnologias de comutação e algumas empresas como a CISCO e a IBM como pretendiam fazer de seus produtos o padrão a ser seguido. Entretanto, em 1996 foi formado um grupo de trabalho com o objetivo de integrar o roteamento da camada 3 e a comutação de rótulos da camada 2 em uma tecnologia que englobasse os principais aspectos das tecnologias de comutação de rótulos. Foi criado assim o grupo de trabalho MPLS (*MultiProtocol Label Switching*).

Nas próximas seções, serão discutidos os principais aspectos da tecnologia MPLS. Na seção 2.2 serão apresentadas algumas definições e os conceitos básicos do MPLS. Na seção 2.3 será apresentado o conceito de rótulo e o modo de encaminhamento de pacotes do

MPLS. Na seção 2.4 será definido o conceito de LSP e na seção 2.5 serão discutidos os protocolos de distribuição de rótulos usados no MPLS.

2.2 Conceitos Básicos

Em redes IP convencionais, os pacotes são encaminhados baseados na análise do cabeçalho de rede dos pacotes. A análise dos cabeçalhos é realizada a cada roteador o que, como já mencionado, aumenta consideravelmente o tempo de encaminhamento dos datagramas.

No MPLS, os pacotes são agrupados em Classes de Equivalência de Encaminhamento ou FEC's (*Forwarding Equivalence Classes*). Pacotes pertencentes à mesma FEC seguirão o mesmo caminho e terão o mesmo tratamento através da rede. A utilização de FEC's permite que os pacotes sejam encaminhados baseados não somente no endereço de destino dos pacotes como é tradicionalmente feito em redes IP, como também baseados em outras políticas, já que uma FEC pode representar uma política administrativa, fluxos agregados com os mesmos requisitos de QoS, precedência dos pacotes ou classe de serviço.

Os pacotes são associados às FEC's somente uma vez, quando entram na rede. A FEC é então representada por um identificador numérico, de tamanho fixo, denominado rótulo (*label*). Nos demais roteadores, não existe análise do cabeçalho de rede, só do rótulo o que acelera o processo de encaminhamento dos pacotes.

Uma vez associados a uma FEC, os pacotes seguirão pela mesma rota na rede. A esta rota é dado o nome de Caminho Comutado por Rótulos (*Label Switched Paths*). Os LSP's são unidirecionais, ou seja, suportam encaminhamento dos datagramas em um único sentido e são criados por protocolos do plano de controle ou por ação da gerência da rede. A rota a ser estabelecida pode ser escolhida por protocolos de roteamento tradicionais como o *OSPF* ou utilizar algoritmos de Roteamento Baseado em Restrições (*Constraint-based Routing*) [RVC01][MC01].

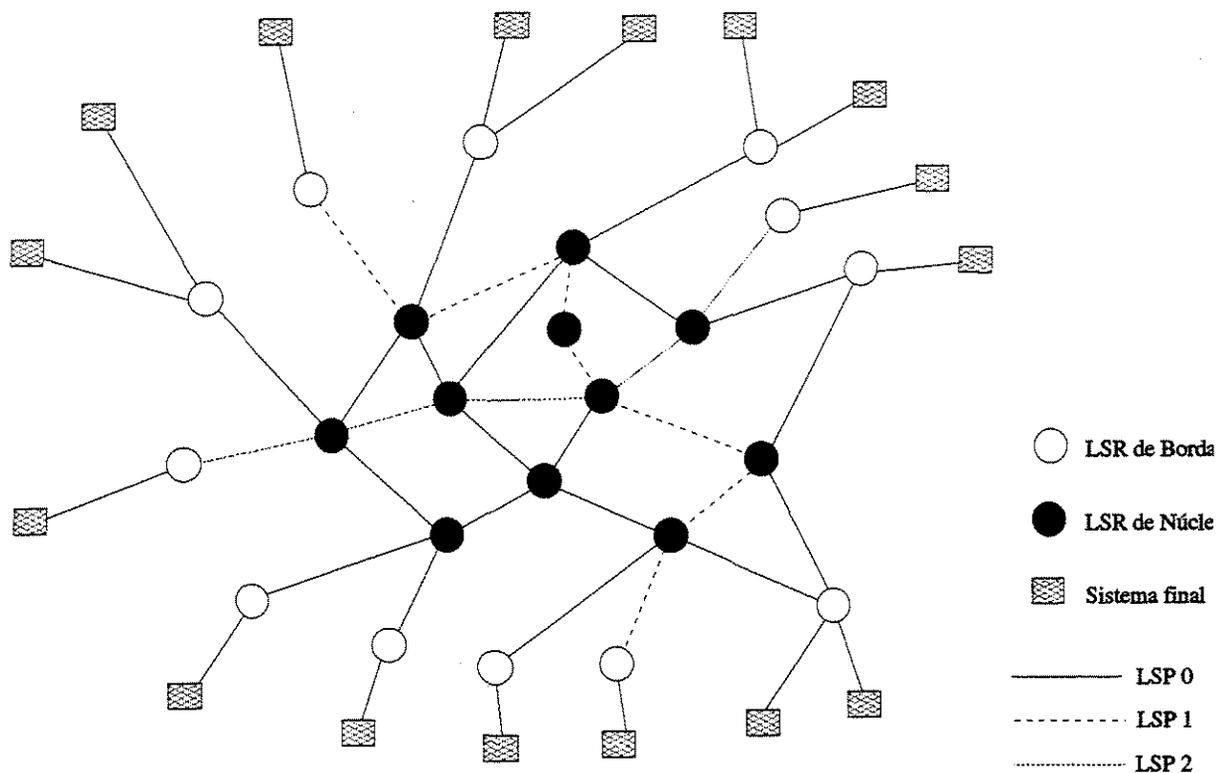


Figura 2.1: Componentes de uma rede MPLS

2.2.1 Componentes das redes MPLS

A Figura 2.1 mostra os principais componentes de uma rede MPLS. Os círculos representam os Roteadores de Comutação por Rótulos ou LSR's (*Label Switch Router*). Estes são subdivididos em duas classes: LSR de Núcleo (*Core LSR*) e são representados pelos círculos pretos. Os LSR's representados pelos círculos brancos são chamados LSR's de Borda (*Edge LSR*) ou LER (*Label Edge Router*). Os LSR de núcleo fazem o encaminhamento dos pacotes no menor tempo que lhes é possível. Os LSR's de Borda são responsáveis pela classificação dos pacotes em FEC's, entre outras coisas. O conjunto de todos os LSR's é chamado de nuvem MPLS.

Cada FEC está associada a um ou mais LSP's. Os LSPs também são ilustrados na figura e podem ser caracterizados por uma sequência de LSR's. O primeiro LSR de um LSP é chamado LSR de ingresso (*Ingress LSR*) e o último é chamado de LSR de Egresso (*Egress LSR*).

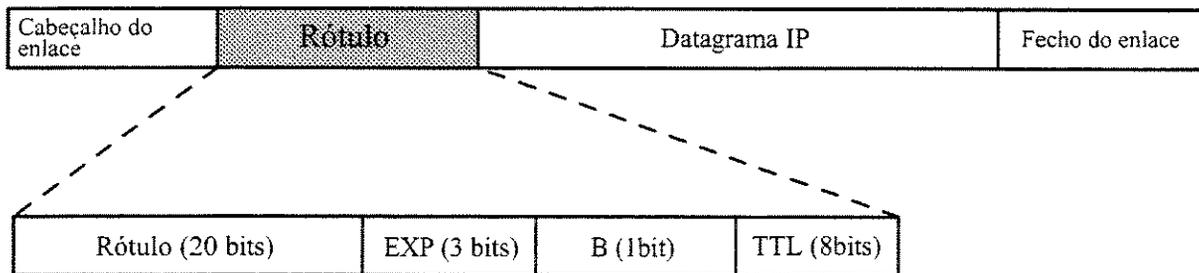


Figura 2.2: Rótulos

O LSR de ingresso de um LSP ao receber um pacote, verifica se o mesmo pertence a uma FEC. Caso afirmativo, o datagrama recebe um rótulo e é encaminhado pelo LSP associado à FEC. Durante o tráfego pelo LSP, os rótulos são trocados a cada LSR. Ao final do LSP, o rótulo é retirado e o datagrama original, sem o rótulo, é encaminhado.

2.3 Rótulos

Um rótulo é um pequeno identificador, de tamanho fixo e com significado local, associado aos circuitos das redes orientadas a conexão. Especificamente nas redes MPLS, os rótulos presentes nos pacotes estão associados a FEC's e esta associação permite que os LSR's decidam qual tipo de encaminhamento o pacote receberá.

O MPLS suporta a superposição de rótulos, ou seja, a adição de rótulos em pacotes já rotulados permitindo a criação de níveis hierárquicos de roteamento. O conjunto de rótulos associado a um pacote é chamado de pilha de rótulos (*Label Stack*). Esta pilha é organizada na forma *last-in, first-out*, isto é, as operações de adição e remoção de rótulos só devem ser realizadas nos rótulos do topo da pilha.

2.3.1 Codificação dos Rótulos

Dado que a arquitetura MPLS não prevê a utilização de *hardware* ou tecnologia de rede/enlace específicas, é necessário que haja uma definição quanto ao significado dos rótulos. A arquitetura MPLS suporta várias técnicas de codificação de rótulos que de-

pendem do tipo de dispositivo utilizado para encaminhar os pacotes.

Se os quadros do protocolo de enlace utilizado sob o MPLS suportarem a utilização de rótulos como identificadores, os mesmos devem ser usados para encaminhar os pacotes de modo que o processamento no nível da camada de redes seja evitado. Porém, alguns protocolos de enlace não utilizam identificadores de conexão, como por exemplo, PPP (*Point-to-Point Protocol*) e Ethernet (IEEE 802.3).

A solução neste caso é a utilização de um Encapsulamento Genérico ilustrado na Figura 2.2. Este encapsulamento genérico, conhecido como *Shim Header*, é posicionado entre os cabeçalhos dos protocolos de rede e enlace, podendo ser usado com qualquer protocolo de rede ou enlace. O *Shim Header* consiste dos seguintes campos:

- um rótulo de 20 bits,
- um campo experimental EXP de 3 bits,
- um campo TTL *Time To Live* de 8 bits e
- um campo B de 1 bit para indicar se o rótulo é o último de uma pilha de rótulos.

Nas redes que utilizam identificadores de conexão como ATM e Frame Relay, os campos de VPI/VCI e DLCI (*Data Link Connection Identifier*) das células e quadros dessas redes, respectivamente, podem ser utilizados para a codificação dos rótulos.

Os procedimentos de encaminhamento de pacotes no MPLS, por razões históricas inclusive, são muitos parecidos com os do ATM. O ATM usa os valores contidos nos campos de VPI/VCI das células como índice para consulta nas tabelas dos comutadores, de onde obtém os valores da porta e dos campos VPI/VCI de saída. Dessa forma, os rótulos MPLS podem ser codificados diretamente nos campos das células ATM e, com algumas atualizações de software, os equipamentos ATM podem ser usados como LSR, sendo assim chamados “ATM-LSR”.

Assumindo o uso da AAL5, a arquitetura MPLS prevê três formas de codificação de rótulos no cabeçalho das células ATM, descritas a seguir:

1. Codificação SVC

Esta técnica usa os campos de VPI/VCI para codificar o rótulo do topo da pilha de rótulos. Cada LSP é representado como Circuito Virtual Comutado ou SVC (*Switched Virtual Circuit*) ATM. Os ATM-LSR's não podem realizar operações na pilha de rótulos já que todos os bits dos campos de VPI/VCI são utilizados para representar um único rótulo.

2. Codificação SVP

Esta técnica usa o campo VPI para codificar o rótulo que está no topo da pilha de rótulos e o campo VCI para codificar um segundo rótulo, se presente. Esta técnica tem a vantagem de permitir comutação por caminho virtual (*VP-switching*).

3. Codificação SVP Multiponto

Esta técnica usa o campo VPI para codificar o rótulo que está no topo da pilha de rótulos. Usa parte do campo VCI para codificar um segundo rótulo, se presente. E o restante do campo VCI para identificar o LSR de ingresso do LSP. Isto permite o uso de conexões multiponto-ponto.

Caso existam mais rótulos na pilha do que os ATM-LSR são capazes de suportar, os mesmos devem ser codificados utilizando o encapsulamento genérico.

2.3.2 Rótulos inválidos

Em algumas casos, um pacote pode chegar a um LSR com um rótulo inválido, ou seja, um rótulo que não corresponde a nenhum rótulo pertencente à sua Base de Informações de Rótulos, ou LIB (*Label Information Base*). A remoção do rótulo e encaminhamento do pacote IP sem rótulo pode provocar a formação de *loops* no caso de LSP's criados explicitamente. A Figura 2.3 ilustra a criação de tais *loops*. Por inconsistências nas LIBs dos LSR "A" e "B", o LSR "A" adiciona o rótulo Z1 no pacote e encaminha o pacote ao LSR "B". Entretanto, na LIB do LSR "B" não consta nenhuma informação a respeito do

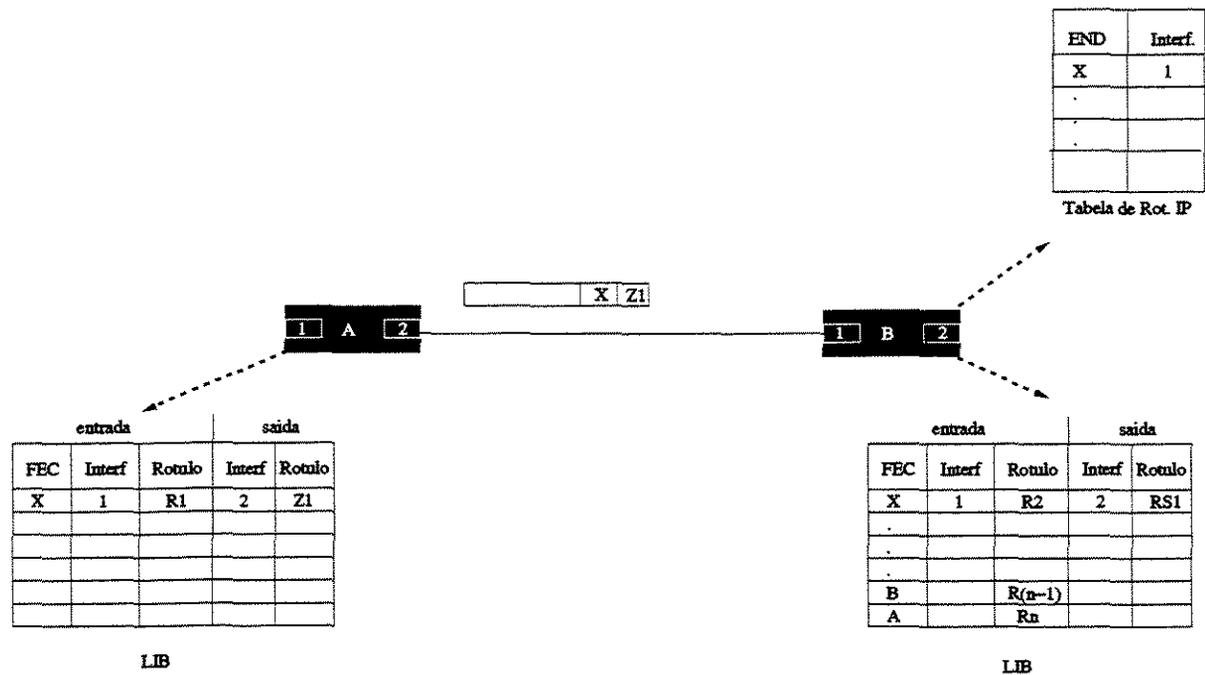


Figura 2.3: Exemplo de problema causado por rótulo inválido

rótulo Z1. Se “B” encaminhar o pacote sem o rótulo, o mesmo retornará para o LSR “A” formando assim um *loop*.

Para evitar este tipo de situação, a arquitetura MPLS [RVC01] determina que os pacotes com rótulos inválidos devem ser descartados a menos que haja outros meios, não especificados pela arquitetura, de evitar *loops* ou outros problemas.

2.4 LSP's

Essencialmente, um LSP é uma rota formada por uma sequência de LSR's. Como já mencionado, os LSP's são unidirecionais e estão associados a uma FEC. A granularidade dos fluxos de pacotes e a quantidade de LSP's são flexíveis.

O controle dos LSP's pode se dar de duas formas: controle independente e ordenado de LSP's. No controle independente, cada LSR ao reconhecer uma FEC em particular, decide se deve ou não associar um rótulo à FEC em questão e distribuí-lo aos LSR's adjacentes.

No controle ordenado, um LSR só associa um rótulo a uma FEC se ele é o LSR de egresso do LSP ou se já recebeu uma notificação de algum LSR adjacente. A arquitetura MPLS [RVC01] indica o uso de controle ordenado de LSP's se o tráfego pertencente à FEC em questão deve seguir por um caminho com propriedades especiais. Pois no controle independente, alguns LSR's podem realizar o encaminhamento do tráfego da FEC antes que o LSP esteja completamente criado fazendo com que parte do tráfego da FEC siga um caminho que não possui o conjunto de características desejadas.

2.4.1 Túneis LSP

Como já mencionado, o MPLS permite a superposição de rótulos formando uma pilha de rótulos. O uso deste tipo de mecanismo possibilita a definição de níveis de hierarquia de rótulos que traz consequências importantes na arquitetura da rede e uma dessas consequências é o tunelamento de informações.

O tunelamento de informações permite, por exemplo, que um provedor possa enviar informações através de domínios administrados por outros provedores sem se preocupar com políticas administrativas, questões de segurança e *hardware* utilizados nos outros domínios.

Um LSP cujo rótulo é o “ m -ésimo” na pilha de rótulos é chamado de “LSP de nível m ”. Todo LSR pertencente ao LSP, exceto o último, recebe pacotes com m rótulos na pilha e toma suas decisões de encaminhamento baseado no “ m -ésimo” rótulo. Além disso, exceto no último e no penúltimo LSRs, as operações de comutação de rótulos em todos os LSRs devem resultar em uma pilha com m rótulos.

O penúltimo LSR de um LSP é responsável pela remoção do rótulo do topo da pilha. Isto acontece porque se o LSR de egresso recebe um pacote com m rótulos, ele analisa o rótulo do topo e como resultado disto constata que é o LSR de egresso do LSP. O próximo passo então é remover o rótulo do topo da pilha e encaminhar o pacote baseado na análise do pacote sem o “ m -ésimo” rótulo. Se existir outro rótulo, o pacote será encaminhado baseado numa pilha de profundidade $m - 1$. Caso não exista mais nenhum rótulo o pacote

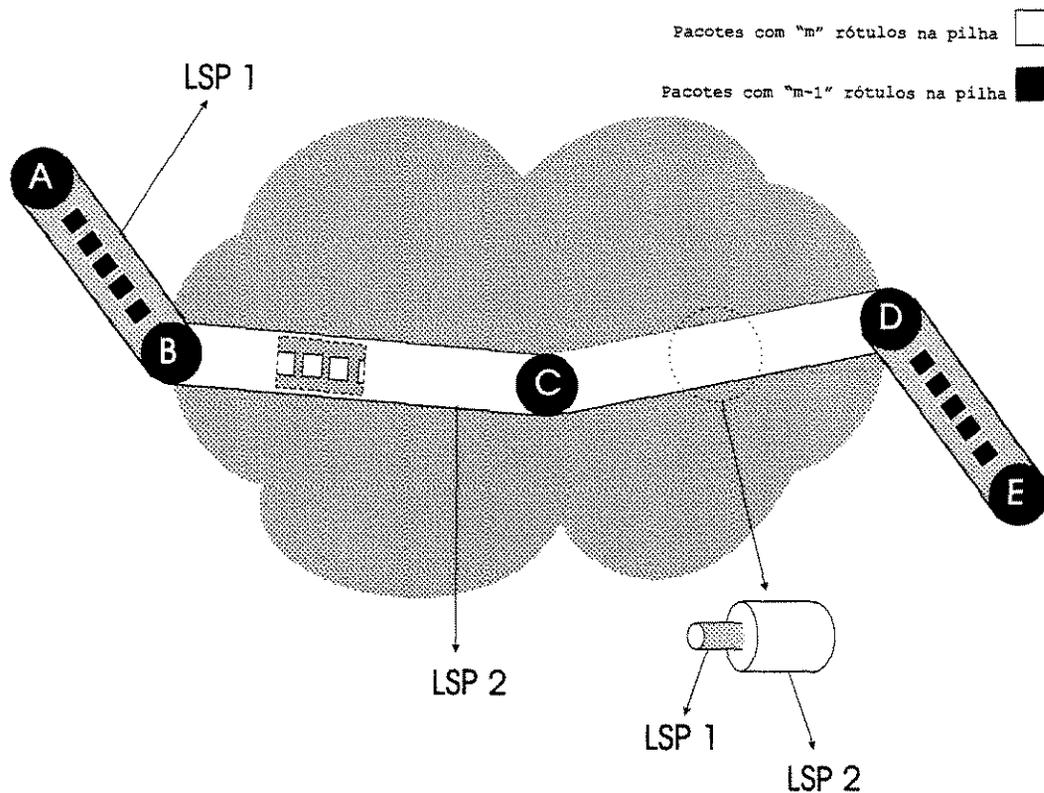


Figura 2.4: Túneis LSP

é encaminhado baseado no endereço de destino do cabeçalho de rede. Pode-se notar que a retirada do rótulo do topo da pilha pelo penúltimo LSR faz com que o LSR de egresso analise o pacote apenas uma vez e não duas como no cenário anterior.

A Figura 2.4 ilustra o tunelamento de LSPs. O LSR "A" é o LSR de ingresso do LSP 1 e o LSR "E" é o LSR de egresso. Os pacotes que trafegam pelo LSP possuem uma pilha de rótulos de profundidade "m-1" (se $m - 1 > 1$ significa que o LSP é um túnel usado por algum outro LSP). Quando os pacotes provenientes do LSP 1 chegam ao LSR "B", que é o LSR de ingresso do LSP 2, um novo rótulo é adicionado à pilha de rótulos. Durante o trajeto pelo LSP 2, os pacotes são comutados usando o rótulo de profundidade "m" até que cheguem ao LSR "C", onde o "m-ésimo" rótulo é retirado.

2.4.2 Roteamento de LSPs

O roteamento de LSPs corresponde ao método usado na seleção das rotas durante o estabelecimento de um LSP. Duas opções são previstas pela arquitetura MPLS:

- roteamento por saltos (*hop-by-hop*) e
- roteamento explícito.

No roteamento por saltos (*hop-by-hop*), é permitido que cada nó escolha de forma independente o próximo nó para cada FEC. Este é o modo convencional do roteamento IP. Um LSP roteado por saltos é um LSP cuja rota é selecionada pelo roteamento por saltos. [RVC01][MC01].

No roteamento explícito, um único LSR, geralmente o LSR de ingresso ou LSR de egresso, escolhe todos os LSR's pelos quais o tráfego deve passar. Quando todo o caminho é indicado, diz-se que se trata de um LSP roteado explicitamente de forma estrita. Por outro lado, se apenas alguns LSR's do caminho forem indicados, diz-se que se trata de um LSP roteado explicitamente de forma fraca.

O roteamento explícito pode ser realizado por configuração ou dinamicamente e a sua maior aplicabilidade é na Engenharia de Tráfego. Apesar de ser possível o roteamento IP na origem (*source routing*), ele nunca foi empregado pois cada datagrama necessita transportar sua rota no campo de opções do cabeçalho IP, aumentando o tamanho e conseqüentemente o tempo de processamento do datagrama.

No MPLS, o roteamento explícito estabelece a rota quando o LSP é criado. Assim, datagramas trafegando pelo LSP não precisam transportar sua rota já que a mesma é definida pelo rótulo.

2.5 Protocolos para Distribuição de Rótulos

Depois que um LSR decide qual rótulo irá utilizar para representar uma FEC, ele deve divulgar esta informação para os LSRs adjacentes. Para tal, precisa utilizar um protocolo

de distribuição de rótulos.

A arquitetura MPLS não define uma solução para a distribuição de rótulos na rede. Entretanto, basicamente duas propostas têm sido empregadas. A primeira é a extensão dos protocolos de roteamento e sinalização já existentes. Extensões para o protocolo BGP (*Border Gateway Protocol*) têm sido feitas de modo que as notificações sobre associações de rótulos a FECs sejam enviadas nas mensagens BGP. Outra extensão vem sendo feita para o protocolo RSVP (*Resource ReserVation Protocol*). Esta extensão consiste em dotar o protocolo RSVP de capacidade de distribuição de rótulos.

A segunda proposta é a criação de um protocolo específico para a distribuição de rótulos em redes MPLS. Este protocolo é chamado LDP (*Label Distribution Protocol*) e sua principal diferença em relação ao RSVP é que no LDP, as associações entre rótulo-FEC são *hard-state*, ou seja, uma vez feitas, elas são mantidas até que sejam explicitamente desfeitas. Já no RSVP, as associações são do tipo *soft-state*, ou seja, a associação deve ser periodicamente confirmada pois caso contrário, será desfeita por estouro do temporizador.

Dois LSR's que utilizam um protocolo de distribuição de rótulos para trocar informações de associações rótulo-FEC são chamados com relação à informação que trocam de Parceiros de Distribuição de Rótulos (*Label Distribution Peers*), ou simplesmente LSR parceiros. A distribuição de rótulos do MPLS denomina-se Distribuição a Jusante (*Downstream*) ou seja, dado o sentido de um fluxo, as associações FEC-rótulo devem ser criadas a jusante (*downstream*) e distribuídas a montante (*upstream*) [RVC01][MC01].

Duas estratégias são utilizadas para a distribuição de associações rótulo-FEC:

- **Sob Demanda** (*Downstream On-Demand*): onde um LSR só realiza a associação de um rótulo a uma FEC em particular quando solicitado por um LSR par a montante;
- **Sem Solicitação** (*Unsolicited Downstream*): cada LSR associa um rótulo a uma FEC e distribui a seus LSR pares a montante mesmo sem ser solicitado a fazê-lo.

2.6 Considerações Finais

O MPLS desde sua concepção tem se tornado uma tecnologia promissora para a provisão de QoS na Internet. Por ser facilmente interoperável com o protocolo IP, e com a pilha de protocolos usada na Internet, a sua adoção por parte dos ISPs (*Internet Service Providers*) é mais atraente. Além disso, o MPLS permite aos provedores de serviço o oferecimento de novos serviços. O uso de encaminhamento de pacotes por caminhos comutados por rótulos permite aos provedores, por exemplo, a possibilidade de esquemas de tarifação diferenciados pelo tipo de serviço utilizado pelos usuários. Além disso, a cada LSP podem ser atribuídas características distintas e preços diferenciados de acordo com tais características.

Também a sua capacidade de “criação” de uma rede baseada em circuitos em uma rede sem conexão como a Internet com um baixo *overhead* é um aspecto fundamental que torna o MPLS atrativo. Além disso, o MPLS permite que técnicas de Engenharia de Tráfego (apresentadas no capítulo 3) sejam utilizadas na Internet, propiciando uma utilização balanceada dos recursos da rede e eliminação de pontos de congestionamentos causados pela má distribuição do tráfego. Tal eliminação de pontos de congestionamentos causados pela má distribuição do tráfego possibilita a transmissão de dados sensíveis a perdas ou a atrasos com a QoS necessária.

O MPLS pode ser usado também juntamente com esquemas de provisionamento de QoS na Internet, como por exemplo, os serviços diferenciados. A cada classe de tráfego pode ser associado um LSP, ou seja, a cada DSCP é associado um rótulo correspondente.

O uso conjunto de técnicas de Engenharia de Tráfego com o MPLS, bem como de esquemas de provisionamento de QoS são problemas ainda em aberto e diversas pesquisas neste sentido têm sido feitas.

Capítulo 3

Engenharia de Tráfego

Como já dito, a simples utilização de mecanismos de provisionamento de Qualidade de Serviço não é garantia de que os requisitos de tráfego das aplicações sejam atendidos. Em redes como a Internet onde existe frequente mudança de topologia e demanda de tráfego, um enlace pode ser saturado, causando perdas, pela distribuição não balanceada do tráfego pela rede.

Assim, a adoção de técnicas de Engenharia de Tráfego é altamente recomendada pois permite aos operadores de rede o oferecimento de serviços em uma rede sem pontos críticos de congestionamentos.

Este capítulo apresenta conceitos, taxonomia e um modelo para a Engenharia de Tráfego [Awd99].

3.1 Introdução

O Congestionamento impacta negativamente na capacidade de oferecimento de QoS das redes. Duas causas principais propiciam o seu acontecimento: os recursos da rede são insuficientes ou inadequados para acomodar a carga oferecida; os fluxos de pacotes são mapeados insatisfatoriamente, fazendo com que determinadas partes da rede sejam sub-utilizadas, enquanto outras encontram-se sobrecarregadas.

Quando a rede começa a entrar em congestionamento com maior frequência, a solução típica é aumentar a capacidade dos seus enlaces. Obviamente, em redes de grande porte essa nem sempre é uma saída viável devido ao alto custo. Tornou-se perceptível a necessidade de realizar o mapeamento dos fluxos na estrutura física da rede através da aplicação sistemática de metodologias bem definidas.

A Engenharia de Tráfego (*Traffic Engineering* ou TE) é o aspecto das redes que está ligado à otimização de desempenho. Segundo Awduche et al. [AJ02], a Engenharia de Tráfego na Internet engloba tecnologia e aplicação de princípios científicos para medir, modelar, caracterizar e controlar o tráfego Internet, de forma a facilitar a operacionalidade e resultar em uma utilização balanceada dos recursos da rede e minimização da ocorrência de congestionamentos.

Através da Engenharia de Tráfego pode-se alcançar benefícios como a capacidade de evitar pontos de congestionamento, roteamento rápido dos fluxos em caso de falhas, uso mais eficiente da banda disponível e possibilidade de oferecimento de melhores garantias de QoS [MC01].

3.2 Modelo do processo de Engenharia de Tráfego

O processo de Engenharia de Tráfego é uma sequência de passos que se deve realizar a fim de otimizar o desempenho da rede. Este processo é ilustrado na Figura 3.1.

A primeira fase do processo de Engenharia de Tráfego é a definição da(s) política(s) de controle que é(são) mais relevante(s) para a operação da rede. Esta política de controle, por sua vez, depende de fatores pertinentes ao contexto da rede como: restrições de operação, custos e critérios de sucesso.

A segunda fase do processo é a observação do estado da rede através de funções de monitoramento. Esta fase é o componente de informações do processo. É nesta fase que as informações sobre o comportamento da rede são obtidas, através da aquisição de dados através de medições. Modelos matemáticos que representem o comportamento do

tráfego ou estimativas através de dados sintéticos também podem ser usados caso não seja possível a aquisição de dados reais.

A terceira fase é a caracterização do tráfego e análise do estado da rede. Várias técnicas, tanto quantitativas quanto qualitativas, podem ser aplicadas nesta fase, assim, fatores que podem estar diminuindo o desempenho da rede são identificados. Técnicas como modelagem analítica ou simulação podem ser usadas nesta fase. Os resultados desta fase são utilizados para otimização do desempenho, dimensionamento dos recursos e re-projeto da rede.

A análise do desempenho da rede pode ser proativa e/ou reativa. Na análise proativa, ocorre a identificação de problemas potenciais que poderão vir a existir no futuro. Por outro lado, na análise reativa, há a identificação de problemas existentes, identificação de suas possíveis causas e indicação de possíveis soluções para o problema.

A quarta fase é a otimização do desempenho da rede. Ela é acompanhada da aplicação de medidas de controle de forma a levar a rede ao estado desejado de acordo com a política de controle estabelecida na primeira fase. Estas medidas de controle podem ser o uso de técnicas apropriadas que controlam o tráfego da rede ou a distribuição do mesmo na rede, uso de enlaces adicionais ou aumento das capacidades dos enlaces, aquisição de novo hardware como roteadores ou *switches* ou ainda:

- modificação das restrições da rede,
- manipulação dos parâmetros de tráfego,
- modificação dos parâmetros associados ao roteamento,
- manipulação dos parâmetros de gerenciamento de tráfego, etc.

Otimização do desempenho da rede pode também envolver mudanças na arquitetura, capacidade, tecnologias empregadas, topologia da rede e a configuração dos elementos de rede de forma que a escalabilidade da mesma seja garantida.

Cabe ressaltar que a Engenharia de Tráfego é um processo adaptativo, o que implica dizer que as fases descritas anteriormente são continuamente repetidas.

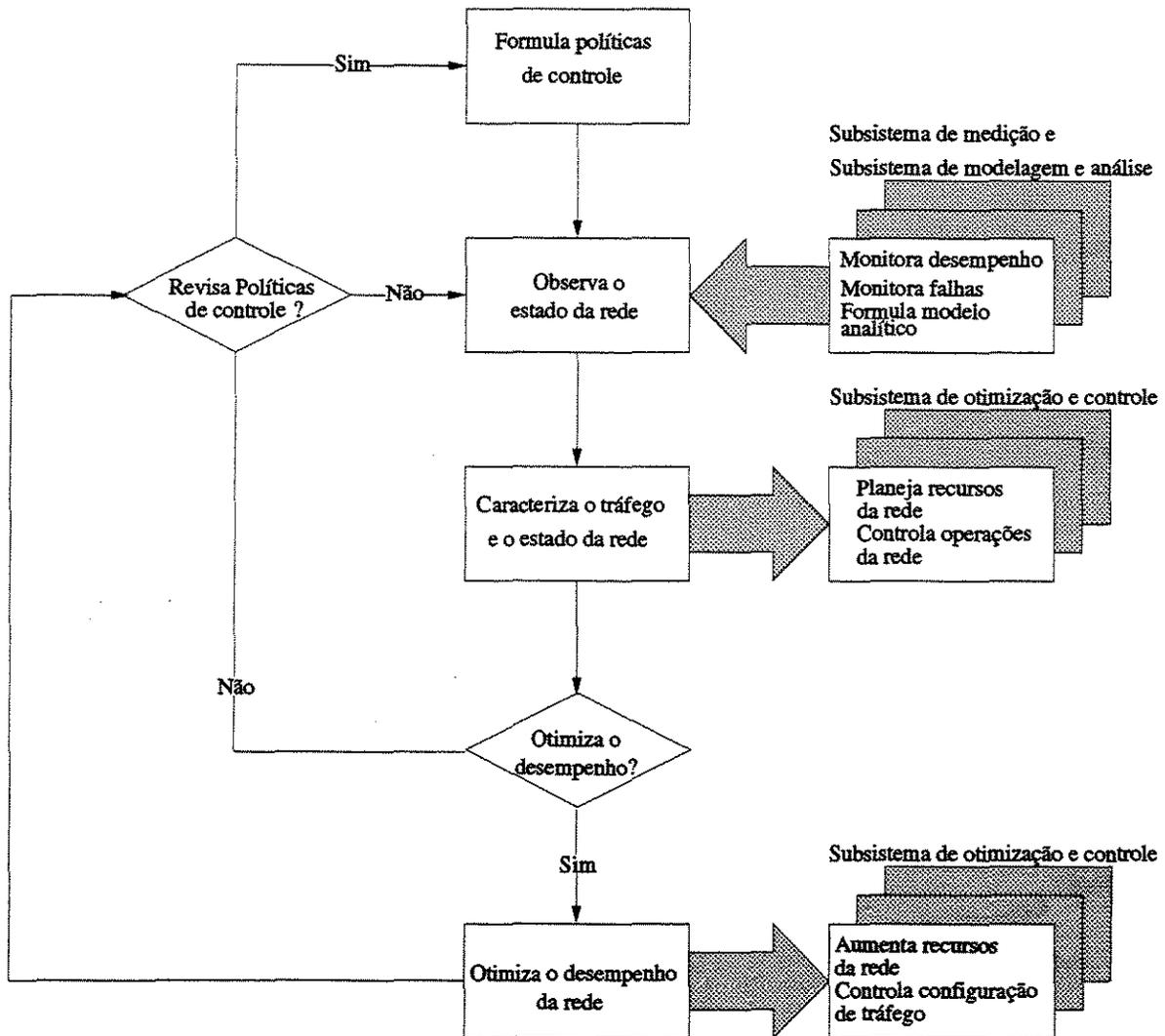


Figura 3.1: Modelo do processo de Engenharia de Tráfego, adaptado de [Awd99]

O processo de Engenharia de Tráfego utiliza alguns componentes básicos que implementam funções que são essenciais. São eles: subsistema de medição, subsistema de modelagem e análise e subsistema de otimização da rede. O subsistema de medição é um componente de fundamental importância pois é o responsável por todas as informações concernentes ao estado da rede. O subsistema de medição pode ser utilizado para determinar a qualidade dos serviços da rede e avaliar a eficácia das políticas de Engenharia de Tráfego [Cav01]. Além disso, os dados gerados por esse subsistema podem ser utilizados pelo subsistema de otimização e modelagem para adaptativamente otimizar o desempenho em resposta a estímulos gerados dentro e fora da rede. A medição do tráfego pode ocorrer em diferentes níveis, como por exemplo, medições podem ser usadas para a verificação de características no nível de pacotes, fluxos individuais ou agregados ou a nível de usuário [Cav01].

O subsistema de modelagem e análise inclui a representação de modelos físicos ou abstratos que descrevem características relevantes do tráfego e atributos da rede. Simulação é uma ferramenta relevante no contexto de Engenharia de Tráfego dados a complexidade e os custos envolvidos na utilização de modelos reais para a análise da rede. O uso de simuladores de qualidade pode, por exemplo, ajudar a visualizar características da rede em vários contextos e condições de forma segura e não-destrutiva. Entretanto, para que o uso de ferramentas de simulação seja eficaz, é necessário um modelo fiel da rede, que capture suas características e atributos. Tais modelos podem facilitar a análise da rede e podem indicar o desempenho da mesma sob diversas circunstâncias, bem como guiar planos de expansão.

O subsistema de otimização visa a otimização dos recursos da rede, o que deve ser realizado de forma contínua, por meio de gerenciamento e de medições de tráfego. A otimização do desempenho da rede pode ser corretiva ou preventiva. Na otimização corretiva, o objetivo é remediar algum problema de desempenho encontrado na rede. Por outro lado, na otimização preventiva, o objetivo é levar a rede a um estado onde o desempenho seja maximizado mesmo sem a existência de problemas. Dentre os recursos

gerenciados pelo subsistema de otimização estão a banda passante dos enlaces, espaço de *buffer* e recursos computacionais.

3.3 Taxonomia

Segundo [Awd99][A1.02a] os sistemas de Engenharia de Tráfego podem ser classificados em:

- **Estrutural ou comportamental**

Nos sistemas de Engenharia de Tráfego estruturais, o foco está voltado aos aspectos relativos à organização da rede e de seus componentes. No modelo comportamental, as atenções estão voltadas para a dinâmica da rede e do tráfego a ela submetido.

- **Dependente do estado da rede ou dependente do instante de tempo:**

No controle dependente do tempo, as informações históricas baseadas em variações periódicas do tráfego juntamente com projeções do tráfego são usadas para pré-programar decisões de roteamento.

No controle dependente do estado da rede, as decisões de roteamento são tomadas de acordo com o estado corrente da rede. A idéia é que baseado no estado corrente da rede é possível prover informações adicionais sobre o tráfego real que não poderiam ser preditas usando informações históricas. Algumas métricas podem ser usadas como parâmetro do estado da rede como utilização, atraso dos pacotes e taxa de perdas.

- ***Offline X Online***

Os sistemas *offline* são aqueles que não precisam ser executados em tempo real. Nestes sistemas, as demandas são conhecidas, ou estimadas, *a priori* e são levadas em consideração no momento da realização dos cálculos de roteamento e estabelecimento de conexões. Uma observação importante é que, pelo fato dos cálculos

não precisarem ser realizados em tempo real, técnicas mais sofisticadas podem ser usadas para a computação das rotas fazendo com que o sistema se aproxime dos valores ótimos para a utilização dos recursos da rede, desde que as demandas sejam estimadas corretamente.

Entretanto, em redes como a Internet onde estimar precisamente as demandas não é uma tarefa fácil, o uso de políticas *offline* pode levar a situações desfavoráveis. Assim, o uso de sistemas *online* é mais recomendado. Sistemas *online* são aqueles que são executados em tempo real. O uso desses sistemas é recomendado quando as decisões de roteamento devem adaptar-se às mudanças de condições da rede. Os cálculos envolvidos devem ser simples e o resultado pode estar deslocado de um ponto ótimo.

- **Centralizado ou distribuído**

No controle centralizado, existe uma unidade central responsável pela determinação das rotas e planos de roteamento. É esta unidade central que é responsável pela coleta periódica de informações do estado da rede, cálculo de rotas e determinação a todos os roteadores sobre os planos de roteamento.

No controle distribuído, cada roteador autonomamente seleciona as rotas dos fluxos que passam por ele baseado em sua visão do estado da rede, obtido por alguma medição que pode ser feita, por exemplo, a partir das informações sobre os enlaces da rede propagadas pelos protocolos de roteamento.

- **Global ou local**

Algoritmos podem requerer informações globais ou locais. As informações locais pertencem ao estado de uma porção do domínio. Exemplos incluem a largura de banda de um caminho e a taxa de perda de pacotes de um caminho em particular. Informações de estado locais podem ser suficientes para certos tipos de sistemas controlados distribuídamente.

Informações globais dizem respeito ao estado do domínio inteiro. Exemplos podem incluir a matriz de tráfego e informações sobre banda disponível em cada enlace. Usualmente os sistemas de Engenharia de Tráfego com controle centralizado são os que utilizam informações de estado globais.

- **Descritivo ou prescritivo**

Um sistema descritivo é aquele que descreve o estado de uma rede sem contudo sugerir ações a serem tomadas. Por outro lado, um sistema prescritivo é aquele em que são prescritas atitudes a serem tomadas.

- **De malha aberta ou fechada**

Engenharia de Tráfego com controle de malha aberta não usa informações do estado atual da rede, ao passo que o controle com malha fechada utiliza tanto informações do estado da rede, baseadas em informações do histórico da rede ou de medições atualizadas.

3.4 Políticas de controle de congestionamento

Minimização de congestionamentos é um aspecto de grande importância para a Engenharia de Tráfego na Internet. Esta seção faz uma breve explanação sobre as políticas de controle de congestionamento propostas em [Awd99][Al.02a].

As políticas para controle de congestionamentos são classificadas por [Awd99] em relação aos critérios de (1) escala de tempo (longa, média e curta duração), (2) comportamento em relação a possíveis situações de congestionamento (reativo ou preventivo) e (3) esquema de gerência de congestionamento (suprimento ou demanda).

- **De longa, média ou curta duração**

1. **Longa duração (semanas a meses)**

Políticas de controle de congestionamento de longa duração relacionam-se ao planejamento de capacidade e baseiam-se em informações históricas para fazer previsões sobre as futuras demandas de tráfego na rede. Em redes operacionais, a atualização de roteadores e enlaces demandam tempo além de serem custosas. Dessa forma, tais atualizações demoram dias e podem levar até mesmo meses de duração.

2. Média duração (minutos a dias)

As políticas de controle de congestionamento de média duração incluem: (1) o ajuste de parâmetros IGP (*Interior Gateway Protocol*) e/ou BGP (*Border Gateway Protocol*) para desviar o tráfego de certos segmentos da rede, (2) o estabelecimento e ajuste de alguns LSPs de rotas explícitas (*Explicated routed Label Switched Paths*) em redes MPLS de forma a desviar o tráfego de alguma região congestionada ou para alguma região com baixa utilização de recursos e (3) reconfiguração da topologia lógica da rede para se aproximar mais da distribuição espacial do tráfego, usando, por exemplo, caminhos de tecnologias orientadas a conexão como LSPs em redes MPLS ou PVCs em redes ATM.

Muitos dos esquemas de resposta em escala de tempo de média duração estão embutidos em um sistema de medição que monitora mudanças na distribuição de tráfego, desvio de tráfego e utilização de recursos da rede e, consequentemente, constituem ferramentas para realimentação de informações para o mecanismo *online* e/ou *offline* de Engenharia de Tráfego que está sendo utilizado.

3. Curta duração (picosegundos a minutos)

Esta categoria inclui processamento a nível de pacotes e eventos na ordem de alguns RTT (*Round Trip Time*). Ela inclui mecanismos de roteamento e gerenciamento de *buffer* tanto passivo quanto ativo, que podem ser usados para controlar o congestionamento ou sinalizar a ocorrência de congestionamento para o sistema final de forma que ele possa adaptativamente regular a taxa

com a qual o tráfego é injetado na rede.

- **Reativas ou preventivas**

1. **Reativas**

Políticas de controle de congestionamento reativas são aquelas que reagem a congestionamentos já existentes de forma a diminuí-los. As políticas de escala de tempo de curta duração, por exemplo, caem nesta classe, especialmente aquelas baseadas no monitoramento e identificação de problemas de congestionamentos já existentes.

2. **Preventivas**

Políticas preventivas são aquelas em que ações são tomadas para prevenir o congestionamento, baseadas em estimativas ou previsões de potenciais congestionamentos futuros. Algumas políticas de escala de tempo de longa e média duração pertencem a esta categoria. Elas não respondem imediatamente a problemas de congestionamentos já existentes. Ao contrário, previsões de tráfego e distribuição da carga são consideradas e as ações são tomadas para evitar a ocorrência de congestionamentos no futuro.

- **De suprimento ou demanda**

Políticas de suprimento são as que procuram aumentar a capacidade efetiva disponível dos enlaces para controlar o congestionamento. Outra forma é proporcionar uma distribuição relativamente balanceada do tráfego pela rede. As políticas de demanda visam controlar o tráfego oferecido para amenizar os problemas de congestionamento.

3.5 MPLS e Engenharia de Tráfego em redes IP

Por ser uma convergência do paradigma de encaminhamento orientado a conexão e do modelo sem conexão utilizado nas redes IP, a tecnologia *MultiProtocol Label Switching*

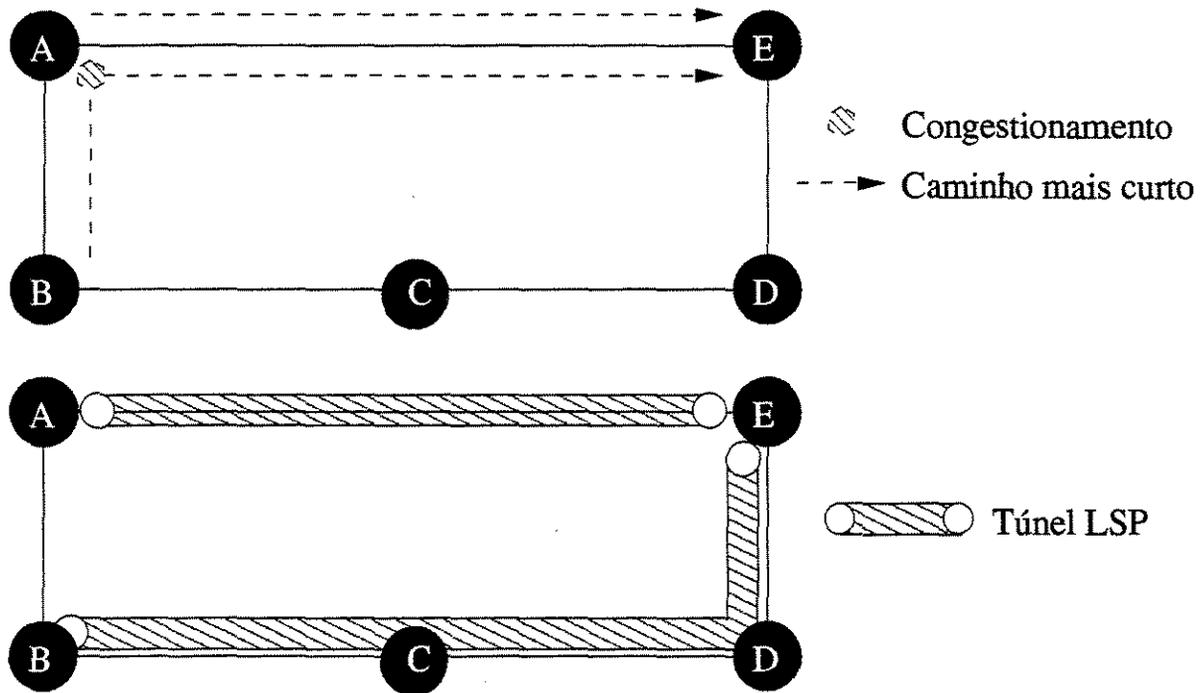


Figura 3.2: Congestionamento causado pela concentração de tráfego no caminho mais curto.

(MPLS) tem sido apontada como fundamental no processo de Engenharia de Tráfego na Internet. Dentre as vantagens do MPLS, menciona-se a capacidade de criar dinamicamente LSP's explícitos a baixo custo operacional.

Esta capacidade do MPLS permite que esquemas sofisticados de roteamento baseado em restrições sejam usados para estabelecimento de LSPs. Assim, rotas menos custosas e que satisfaçam vários requisitos de QoS podem ser utilizadas, fazendo com que o encaminhamento dos pacotes através de tais LSPs seja baseado nas características das FECs a que eles pertencem e não mais no endereço de destino como é feito em redes IP tradicionais.

No contexto de Engenharia de Tráfego em redes IP, os pacotes que pertencem à mesma FEC, ou seja, pacotes que receberão o mesmo encaminhamento através da rede são agregados em troncos de tráfego. Quando o MPLS é utilizado na Engenharia de Tráfego em redes IP, os troncos de tráfego são mapeados na topologia física da rede através de LSPs

criados explicitamente. Os termos túneis LSP (*LSP tunnel* e *traffic engineering tunnel*) são frequentemente usados para se referir à combinação entre MPLS e troncos de tráfego.

A Figura 3.2 ilustra a utilização de túneis LSP na redistribuição de tráfego de modo a evitar congestionamentos causados por protocolos IGP. Como já mencionado, os protocolos IGP utilizam algoritmos SPF que retornam sempre o menor caminho (no caso da figura, menor caminho em número de nós). No caso de redes com alta demanda, os enlaces pertencentes a esse caminho tornam-se congestionados causando diminuição da QoS. LSP's podem ser criados por caminhos distintos, usando algoritmos de roteamento baseados em restrições, para distribuir o tráfego e diminuir a ocorrência de congestionamentos.

Alguns problemas são inerentes ao contexto de Engenharia de Tráfego com MPLS. São eles: como mapear pacotes em FECs, como mapear as FECs em troncos de tráfego e como mapear tronco de tráfego em LSPs [Cav01]. Além desses problemas, outros inerentes à arquitetura do MPLS também afetam o desempenho de sistemas de engenharia de tráfego que são: roteamento de LSPs, particionamento de tráfego entre múltiplos LSPs e o dimensionamento adequado dos LSPs para atender a demanda efetiva de tráfego [Al.02b].

Um modelo de Engenharia de Tráfego usando MPLS possui os seguintes componentes funcionais [Awd99]: componente de gerenciamento de caminhos (*Path Management*), atribuição de tráfego (*Traffic assignment*), disseminação das informações de estado da rede (*Network state information dissemination*) e gerenciamento de rede (*Network Management*).

Gerenciamento de caminhos diz respeito à seleção de rotas explícitas, instanciação e manutenção dos caminhos. O componente de gerenciamento de caminhos usa três funções básicas que são:

- **Seleção de rotas:** Esta função seleciona uma rota para o túnel LSP do nó de origem até o destino, que pode ser definida administrativamente ou computada por algoritmos de roteamento baseados em restrições.

- **Criação de caminhos:** Esta função é usada para instanciar os túneis LSP usando um protocolo de sinalização como RSVP ou CR-LDP.
- **Manutenção de caminhos:** Esta função é responsável, como o nome diz, pela manutenção dos caminhos e pela sua terminação.

O componente de atribuição de tráfego é responsável por todos os aspectos relativos à associação do tráfego a um LSP já estabelecido. A aplicação de filtros que examinam características dos pacotes é um dos aspectos concernentes ao mapeamento de tráfego em um LSP. Por exemplo, quando pacotes sem rótulos chegam em um roteador de borda, a classificação é feita através de filtros que examinam os campos do cabeçalho IP como endereço de destino, endereço de origem e portas de origem e destino, para que haja a posterior atribuição dos pacotes a um LSP.

Outro aspecto importante relacionado ao componente de atribuição de tráfego é o balanceamento de carga entre múltiplos LSPs estabelecidos entre nós de origem e destino em comum. Políticas de particionamento e balanceamento de carga quando bem implementadas, podem diminuir a carga de um LSP, evitando que o mesmo se torne congestionado.

Disseminação de informações do estado da rede diz respeito à disseminação de informações relevantes em relação à topologia da rede MPLS. Estas informações sobre topologia podem posteriormente ser usadas por algoritmos de roteamento baseado em restrições para calcular rotas explícitas já que o conhecimento da topologia é necessário para este tipo de algoritmo.

O componente de gerenciamento de rede inclui um conjunto de funções de desempenho, medições, gerenciamento de falhas, contabilização e tarifação que juntas permitem que uma rede MPLS possa ser administrada com eficácia.

3.6 Considerações Finais

Como mencionado, a adoção de técnicas de oferecimento de garantias de QoS, como por exemplo diferenciação de serviços e/ou integração de serviços, não é garantia estrita da ausência de congestionamentos na rede. Isto é verdade pois nem sempre o tráfego é bem distribuído pelos caminhos da rede, o que pode, num cenário não muito pessimista, ocasionar a concentração de tráfego em um único caminho, podendo acarretar a saturação do mesmo. Dessa forma, a implantação de serviços que exijam da rede o rígido cumprimento de requisitos de QoS requer a adoção de políticas de Engenharia de Tráfego que permitam esse cumprimento.

Em redes IP, como é o caso da Internet, o MPLS pode ser de fundamental importância para que o tráfego seja mapeado na rede de forma satisfatória. Isto porque o MPLS permite, como já mencionado, a associação de circuitos às redes sem conexão. Assim, tais circuitos podem ser usados para conduzir o tráfego por caminhos designados por esquemas sofisticados de roteamento, utilizados de acordo com a política de Engenharia de Tráfego adotada na rede.

Capítulo 4

Roteamento com interferência mínima

A escolha de caminhos que satisfaçam múltiplos requisitos independentes de QoS é um problema NP-completo, ou seja, não se conhece algoritmo polinomial para resolvê-lo [CN98, GJ79]. A solução para problemas de roteamento baseado em QoS na Internet passa pela adoção de heurísticas que encontrem caminhos factíveis em tempo real.

A escolha de um caminho para o tráfego entre um par origem-destino pode seguir critérios diversos. Os algoritmos de interferência mínima o fazem de forma a minimizar a probabilidade de bloqueio das (futuras) requisições para estabelecimento de caminhos. A minimização da probabilidade de bloqueio é realizada através da minimização da redução do fluxo máximo entre outros pares origem-destino. A redução no fluxo máximo entre pares origem-destino devido ao roteamento de uma conexão específica entre um outro par origem-destino é chamada interferência.

Diversos trabalhos vêm sendo propostos para resolver o problema de roteamento com interferência mínima. Neste capítulo, os algoritmos de roteamento com interferência mínima existentes na literatura, bem como uma nova heurística para o problema denominada *Light Minimum Interference Routing* (LMIR) [FdFM03] são apresentados.

Este capítulo apresenta os conceitos de fluxos em redes utilizados na construção dos

algoritmos de roteamento com interferência mínima, bem como os algoritmos de roteamento. A seção 4.1 apresenta os principais conceitos de fluxos em redes relacionados a roteamento com interferência mínima. A seção 4.2 apresenta o problema de roteamento com interferência mínima, a seção 4.4 apresenta o algoritmo *Minimum Interference Routing Algorithm - MIRA*, a seção 4.5 apresenta o algoritmo de Wang, Su e Cheng [WSC02] e por fim, a seção 4.6 apresenta o algoritmo LMIR.

4.1 Fluxos em Redes

Esta seção apresenta conceitos de fluxos em redes que serão utilizados na construção dos algoritmos de roteamento com interferência mínima que serão posteriormente apresentados na seção 4.2.

Formalmente fluxos em redes podem ser definidos da seguinte maneira: seja $G = (V, E)$ um grafo direcionado, onde $V(G)$ é o conjunto dos vértices e $E(G)$ é o conjunto de arestas. Seja c uma função capacidade tal que que cada aresta $(u, v) \in E(G)$ tem uma capacidade não negativa $c(u, v) \geq 0$ associada. Se $(u, v) \notin E$ supõe-se que $c(u, v) = 0$. Assume-se, por conveniência, que para todo vértice $v \in V$, existe um caminho $s \rightsquigarrow v \rightsquigarrow d$, que o grafo não orientado subjacente a G é conexo e que $|E| \geq |V| - 1$. Seja s o vértice de origem, e d o vértice destino. Um fluxo em G é uma função real $f : V \times V \rightarrow R$ que satisfaz às seguintes propriedades [CLR90]:

- **restrição de capacidade:** Para todo $u, v \in V$, $f(u, v) \leq c(u, v)$
- **simetria oblíqua:** Para todo $u, v \in V$, $f(u, v) = -f(v, u)$
- **conservação de fluxo:** Para todo $u \in V - \{s, d\}$, $\sum_{v \in V} f(u, v) = 0$

A restrição da capacidade afirma que o fluxo de um vértice até outro não deve exceder a capacidade dada. A simetria oblíqua é uma conveniência de notação que afirma que o fluxo de um vértice u até um vértice v é o valor negativo do fluxo no inverso. A

propriedade de conservação de fluxo afirma que o fluxo total para fora de um vértice que não seja a origem ou o destino é 0.

A quantidade $f(u, v)$ é chamada fluxo do vértice u ao vértice v . O valor de um fluxo f é definido como

$$|f| = \sum_{v \in V} f(s, v) \quad (4.1)$$

O fluxo total positivo que entra num vértice v é definido por:

$$\sum_{u \in V | f(u, v) > 0} f(u, v). \quad (4.2)$$

O fluxo total positivo que sai de um vértice é definido de forma simétrica. Define-se o **fluxo total líquido** em um vértice como o fluxo total positivo que sai de um vértice menos o fluxo total positivo que entra num vértice.

No problema de fluxo máximo, deseja-se encontrar o fluxo de valor máximo do vértice de origem ao vértice de destino.

4.1.1 O método de Ford-Fulkerson

O método¹ de Ford-Fulkerson depende de três idéias importantes que são apresentadas nas próximas seções: redes residuais, caminhos aumentantes (ou caminhos em ampliação) e cortes.

Método 1 Ford-Fulkerson

Ford-Fulkerson

- 1: Inicializar fluxo f como 0
 - 2: enquanto existir caminho aumentante p faça
 - 3: ampliar fluxo f ao longo de p
-

O método de Ford-Fulkerson é iterativo. Ele começa fazendo $f(u, v) = 0$ para todo $u, v \in V$. A cada iteração o valor do fluxo é aumentado ao encontrar um caminho

¹Assim chamado, em vez de algoritmo pois engloba diversas implementações com diferentes tempos de execução [CLR90]

aumentante (que a princípio pode ser imaginado como um caminho desde a origem até o destino ao longo do qual pode-se empurrar mais fluxo). Este processo é repetido até que não haja mais nenhum caminho aumentante.

4.1.2 Redes residuais

Define-se capacidade residual de uma aresta como sendo

$$c_f(u, v) = c(u, v) - f(u, v), \quad (4.3)$$

onde $c_f(u, v)$ é a capacidade residual da aresta (u, v) .

Assim, uma rede residual (ou grafo residual) é a rede $G' = (V, E')$ induzida por f onde

$$E' = \{(u, v) \in V \times V : c_f(u, v) > 0\}.$$

4.1.3 Caminhos Aumentantes

Dado um grafo $G = (V, E)$ e um fluxo f , um caminho aumentante (ou caminho em ampliação) p é um caminho desde o vértice de origem até o vértice de destino no grafo residual G' .

A quantidade máxima pela qual se pode aumentar o fluxo em cada aresta do caminho sem que a restrição de capacidade seja violada é chamada de **capacidade residual de p** e é dada por:

$$c_f(p) = \min c_f(u, v) : (u, v) \text{ está em } p.$$

4.1.4 Cortes de fluxo em redes

Um corte (S, T) de um fluxo em rede é uma partição do conjunto de vértices V em um conjunto S e em um conjunto $T = V - S$ tal que $s \in S$ e $d \in T$. Se f é um fluxo, então o fluxo líquido pelo corte (S, T) é definido por $f(S, T)$ e a capacidade do corte (S, T) é

$c(S, T)$. Um corte mínimo de um grafo é um corte cuja capacidade é mínima dentre todos os cortes da rede.

Como visto, o método de Ford-Fulkerson aumenta repetidamente o fluxo ao longo de caminhos aumentantes até ser encontrado um fluxo máximo. O teorema do fluxo máximo e corte mínimo informa que um fluxo é máximo se o grafo não contém nenhum caminho aumentante.

Teorema de fluxo máximo e corte mínimo [CLR90] 1 Se f é um fluxo em $G = (V, E)$ com origem s e destino t , então as condições a seguir são equivalentes:

1. f é um fluxo máximo em G
2. O grafo residual G' não contém nenhum caminho aumentante
3. $|f| = c(S, T)$ para algum corte (S, T) de G .

Prova A prova do teorema pode ser encontrada em [CLR90], página 520.

4.2 Algoritmos de roteamento com interferência mínima

Os algoritmos de roteamento com interferência mínima objetivam reduzir a probabilidade de bloqueio da rede, encontrando caminhos que minimizem a redução do fluxo máximo entre outros pares origem-destino.

A idéia central dos algoritmos de interferência mínima é que quanto maior for o fluxo máximo disponível entre um par origem-destino menor será a probabilidade de bloqueio das requisições desse par. Assim, os algoritmos de interferência mínima tentam rotear uma requisição de uso de banda passante por caminhos que maximizem o fluxo máximo disponível entre outros pares origem-destino, de forma a poder acomodar futuras requisições. Este é um problema NP-Difícil [KL00]. Existem, porém, heurísticas precisas para este problema. A Figura 4.1 ilustra o conceito de interferência mínima. Após uma requisição de um caminho entre $S3-D3$, com “ n ” unidades de banda, o algoritmo tenta escolher o

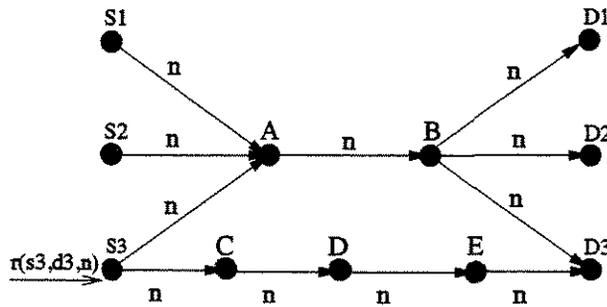


Figura 4.1: Interferência: exemplo

melhor caminho. Se o caminho escolhido for o caminho mais curto $S3 - A - B - D3$, então todas as futuras requisições entre os pares $S1-D1$ e $S2-D2$ são perdidas. Assim, o caminho com interferência mínima seria o $S3 - C - D - E - D3$, mesmo sendo mais longo que $S3 - A - B - D3$.

4.3 Notação e definição do problema

Nesta seção é apresentada a notação utilizada pelos algoritmos de roteamento com interferência mínima, bem como as premissas básicas assumidas pelos mesmos.

A rede é modelada como um grafo $G = (V, E)$ onde $V(G)$ é o conjunto dos vértices e $E(G)$ é o conjunto das arestas. Os dados transmitidos entre dois vértices de G são modelados como fluxos que podem ser enviados entre estes vértices.

Uma capacidade positiva, representada por $c(u, v)$, é associada a cada aresta (u, v) . Ela representa a quantidade máxima de fluxo que pode ser enviada pela aresta (u, v) . Como já definida na seção 4.1.2, a capacidade residual de cada aresta (u, v) é representada por $c_f(u, v)$ e modela a banda disponível em cada enlace.

Seja E' um conjunto de arestas com capacidade residual maior do que zero. O grafo residual $G' = (V, E')$ é definido como o subgrafo induzido por E' .

Um subconjunto P formado por pares ordenados do conjunto vértices é conhecido como conjunto dos pares origem-destino. Os elementos de P representam os únicos pontos de entrada/saída de tráfego da rede e somente entre cada par origem-destino os LSPs serão

criados. Entretanto, não é necessário que haja um LSP entre todo par origem-destino.

Os LSPs são criados através de roteamento explícito, o que significa que para determinar a rota entre par origem-destino, o vértice de origem necessita conhecer toda a topologia da rede. Assim, assume-se que a topologia é conhecida administrativamente ou que um protocolo de roteamento baseado no estado do enlace está operacional e que sua base de dados é acessível.

Os protocolos de roteamento mantêm registro de todas as capacidades residuais e assume-se que todas as capacidades iniciais são conhecidas e assim, a capacidade inicial da rede também. Tais protocolos são também responsáveis pela atualização das informações de banda residual dos enlaces e mudanças de topologia. Falhas dos LSPs são detectadas via protocolos de sinalização como (ex. RSVP-TE ou CR-LDP).

O tráfego é modelado em termos de requisições para estabelecimento de LSPs com requisitos específicos de largura de banda. Assim, uma requisição i é caracterizada pela tripla (S_i, D_i, bw_i) , onde S_i denota o vértice de origem, D_i denota o vértice de destino e bw_i denota a largura de banda requerida pela requisição i .

Além disso, as requisições para estabelecimento de LSPs chegam *on-line*, uma por vez, e não existe conhecimento das características das futuras demandas nem de perfis estatísticos do tráfego. O objetivo dos algoritmos é encontrar no grafo residual G' um caminho de S_i para D_i pelo qual cada aresta tenha pelo menos bw unidades de banda. Caso não haja tal caminho a requisição é rejeitada. Além disso, o caminho escolhido deve minimizar a redução do fluxo máximo entre os outros pares origem-destino e o particionamento do fluxo requerido entre múltiplos caminhos não é permitido.

4.4 O algoritmo MIRA

O primeiro algoritmo publicado que introduziu a terminologia “interferência mínima”, foi o *Minimum Interference Routing Algorithm - MIRA* [KL00]. Este algoritmo assume que os pares origem-destino são conhecidos, bem como a topologia. Além disso, ele assume



a existência de protocolos de sinalização responsáveis pela difusão das informações sobre topologia, banda residual e rotas explícitas.

O fluxo máximo disponível entre um par origem-destino é conhecido como *maxflow* e associado a ele está um corte mínimo [CLR90]. O *maxflow* de um par origem-destino diminui quando a capacidade de uma aresta que compõe um conjunto de corte mínimo diminui. Sempre que uma nova requisição é aceita entre um par origem-destino, o fluxo máximo residual desse par diminui já que o fluxo obrigatoriamente passa por uma aresta de corte. A idéia do algoritmo *MIRA* é evitar rotar as requisições por arestas que pertencem ao conjunto de corte mínimo de outros pares *origem-destino*.

O algoritmo *MIRA* é apresentado a seguir e pode ser descrito como segue.

Algoritmo 2 MIRA

ENTRADA

Um grafo residual $G' = (V, E')$ e $ri(a, b, bw)$, uma requisição de bw unidades de banda entre o par a, b .

SAIDA

Rota entre a e b com bw unidades de banda

MIRA

- 1: Calcule o fluxo máximo para todo $(s, d) \in P \setminus (a, b)$,
 - 2: Determine o conjunto de enlaces críticos C_{sd}
 - 3: Calcule os pesos $w(u, v) = \sum_{(s,d):(u,v) \in C_{sd}} \alpha_{sd} \forall (u, v) \in E$
 - 4: Elimine as arestas com capacidade residual menor do que bw
 - 5: Execute o algoritmo de *Dijkstra* usando $w(u, v)$ como peso.
 - 6: Estabeleça o LSP entre a e b e atualize a capacidade das arestas.
-

O algoritmo possui três etapas. A primeira compreende a computação do *maxflow* entre todos os pares $(s, d) \in P \setminus (a, b)$, onde P é o conjunto de todos . A segunda etapa inclui o cálculo das arestas críticas e a atribuição dos pesos às mesmas. Uma aresta (u, v) é considerada crítica se as três condições são satisfeitas:

- $c_f(u, v) = 0$
- $v \notin S$ e $u \notin T$
- Não existe caminho entre u e v no grafo residual G' ,

onde S o conjunto de vértices alcançáveis a partir de a no grafo residual e seja T o conjunto de vértices alcançáveis a partir de b no grafo residual.

A primeira condição, $c_f(u, v) = 0$, diz que a aresta (u, v) deve ter capacidade igual a zero no grafo residual. A segunda condição, $v \notin S$ e $u \notin T$, diz que u não pode ser alcançável a partir de a e v não pode ser alcançável a partir de b . A terceira condição diz que não deve existir nenhum caminho entre u e v no grafo residual. Após uma aresta ser identificada como crítica, ela é incluída no conjunto C_{sd} que é o conjunto de todas as arestas críticas para os par (s, d)

Os pesos são calculados segundo a expressão:

$$w(u, v) = \sum_{(s,d):(u,v) \in C_{sd}} \alpha_{sd} \quad \forall (u, v) \in E \quad (4.4)$$

onde α_{sd} é o peso do par (s, d) e C_{sd} é o conjunto das arestas críticas para o par (s, d) . O uso de peso para pares origem-destino pode ser especialmente útil quando se quer atribuir prioridades entre os mesmos. A terceira etapa do algoritmo é a execução do algoritmo de *Dijkstra*, usando $w(u, v)$ como peso das arestas.

4.5 O algoritmo de Wang, Su, e Chen (WSC)

Um outro algoritmo de interferência mínima foi proposto por Wang, Su e Chen [WSC02]. Ele é baseado no algoritmo *MIRA* e tem as mesmas premissas básicas, ou seja, a existência de protocolos especializados na difusão das informações de estado e conhecimento da topologia da rede. Suas principais diferenças estão na detecção da criticalidade das arestas e na função de custo.

Segundo Wang et. al [WSC02], o algoritmo *MIRA* pode falhar pois está centrado na detecção de enlaces críticos somente para um único par origem-destino. Assim, ele não pode detectar a criticalidade de um enlace para um grupo de pares, rejeitando muitas requisições em topologias específicas como em grafos com vértices concentradores ou

vértices distribuidores.

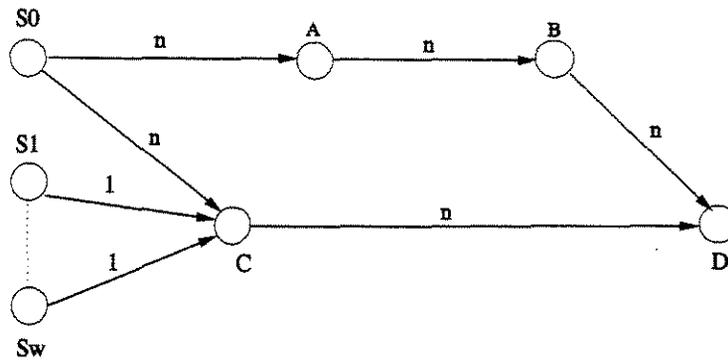


Figura 4.2: Rede com vértice concentrador

A Figura 4.2, extraída de [WSC02], ilustra o problema em redes com vértices concentradores. Suponha que chegue na rede uma demanda de $w + 1$ requisições, em ordem, para (S_0, D) , (S_1, D) , ..., (S_w, D) . A primeira requisição requer um LSP com n unidades de banda, enquanto que todas as demais requerem um LSPs com 1 unidade de banda.

Quando a primeira requisição chega, o primeiro passo do algoritmo *MIRA*, é o cálculo do fluxo máximo e identificação das arestas críticas entre todos os outros pares origem-destino, a saber, (S_1, D) , ..., (S_w, D) . É fácil notar que as arestas críticas, e consequentemente as únicas com $w(u, v) \neq 0$, serão as arestas (S_1, C) , ..., (S_w, C) já que a aresta (C, D) não está no corte mínimo de nenhum dos pares origem-destino em questão. Dessa forma, os pesos de todas as arestas dos dois possíveis caminhos entre (S_0, D) seriam iguais e devido ao menor número de vértices, o caminho selecionado seria o caminho $\{S_0, C, D\}$.

Após o LSP ser estabelecido entre (S_0, D) a capacidade residual da aresta (C, D) é atualizada para 0 e nenhuma das requisições subsequentes poderão ser atendidas.

Por essas razões, em [WSC02] os autores propuseram uma maneira de identificar o impacto relativo à utilização dos enlaces nas futuras requisições. A idéia é calcular a contribuição do enlace em relação ao *maxflow* entre um par origem-destino, evitando os

enlaces de menor contribuição. Os pesos utilizados nas arestas são calculados por

$$w(u, v) = \sum_{(s', d') \in P} \frac{f_{(u,v)}^{s'd'}}{\theta^{s'd'} \cdot c_f(u, v)}, \quad (u, v) \in E, \quad (4.5)$$

onde $c_f(u, v)$ é a capacidade residual do enlace (u, v) , $\theta^{s'd'}$ representa o fluxo máximo entre o par (s', d') e $f_{(u,v)}^{s'd'}$ é a contribuição da aresta (u, v) no fluxo máximo entre o par (s', d') .

Uma vez computados os pesos, as arestas que têm capacidade residual inferior à banda requisitada são eliminadas e o algoritmo de *Dijkstra* é executado usando $w(l)$ como peso [WSC02].

A detecção das arestas críticas é, sem dúvida, a tarefa crucial dos dois algoritmos apresentados nesta seção pois é o que determina a sua eficiência. Os dois algoritmos apresentados utilizam o *maxflow* para a detecção das arestas críticas. O algoritmo *MIRA* [KL00] utiliza o conjunto de corte mínimo associado ao *maxflow* para realizar esta detecção, ao passo que o algoritmo de Wang, Su e Chen apresentado em [WSC02] baseia-se na contribuição de cada enlace para realizar a determinação de criticalidade dos mesmos.

Algoritmo 3 WSC

ENTRADA

Um grafo residual $G' = (V, E')$ e $ri(a, b, bw)$, uma requisição de bw unidades de banda entre o par a, b .

SAIDA

Rota entre a e b com bw unidades de banda

WSC

- 1: Calcule o fluxo máximo para todo $(s, d) \in P$,
 - 2: Calcule os pesos $w(u, v)$ de acordo com a equação 4.5,
 - 3: Elimine as arestas com capacidade residual menor do que bw ,
 - 4: Execute o algoritmo de *Dijkstra* usando $w(u, v)$ como peso.,
 - 5: Estabeleça o LSP entre a e b e atualize a capacidade das arestas.
-

A necessidade de execução do *maxflow* majora a complexidade de ambos os algoritmos mencionados, sendo uma tarefa custosa e proibitiva em redes de médio/grande porte. O método mais comum para cálculo do *maxflow* é o método de *Ford-Fulkerson* [CLR90]. O

algoritmo de *Edmonds-Karp* é a sua implementação mais conhecida e usa uma busca em largura no grafo para descobrir os caminhos aumentantes. Sua complexidade é $O(V \cdot E^2)$.

Em redes como a Internet, o grau médio (*average degree*) chega a ser igual a 3.5 [ZCB96]. Em outras palavras, o número de arestas é significativamente maior que o de vértices, o que leva o algoritmo de *Edmonds-Karp* a trabalhar próximo do pior caso (grafos densos). O algoritmo de *maxflow* mais rápido é o de Goldberg et. al [GR98]. Ele tem complexidade igual a $O(\min(V^{2/3}, E^{1/2}) \cdot E \cdot \log(V^2/E) \cdot \log(U))$ em uma rede com $|V|$ vértices, $|E|$ arestas e enlaces com capacidades no intervalo $[1, U]$ [GR98]. Apesar do algoritmo de Goldberg apresentar uma redução significativa na complexidade do cálculo do *maxflow*, outras abordagens que evitem o uso de algoritmos de *maxflow* podem ser utilizadas para reduzir ainda mais a complexidade dos algoritmos de interferência mínima, tal como é feito neste trabalho.

Além do cálculo do *maxflow*, os algoritmos de interferência mínima demandam também o tempo para determinação da criticalidade das arestas, alocação dos pesos e mais $O(V^2 + E)$ para execução do algoritmo de *Dijkstra*. Contudo, a complexidade computacional dessas etapas ainda é menor que a do algoritmo do *maxflow*.

4.6 O algoritmo *Light Minimum Interference Routing*

Como mencionado anteriormente, os algoritmos de *maxflow* podem causar um impacto significativo na complexidade dos algoritmos de roteamento com interferência mínima. Introduz-se neste capítulo, um novo algoritmo denominado *Light Minimum Interference Routing (LMIR)* para roteamento com interferência mínima que não utiliza algoritmos *maxflow*.

O LMIR tenta solucionar o problema através de uma abordagem diferente, buscando encontrar os K caminhos com menor capacidade entre todos os pares origem-destino para então determinar as arestas críticas.

A idéia é que o caminho entre um par origem-destino com menor capacidade, contém

a aresta de menor capacidade, ou seja, a aresta crítica. Dessa forma, encontrar os K piores caminhos significa encontrar, pelo menos, as K arestas críticas entre o par origem-destino em questão. Assim, o algoritmo LMIR busca as K arestas críticas entre cada par origem-destino e procura evitá-las.

O algoritmo LMIR tenta encontrar rotas que minimizem a interferência entre os pares origem-destino da rede, gerando uma utilização balanceada dos recursos da mesma. Ao chegar uma requisição de bw unidades de banda passante para um par origem-destino (s, d) , $ri(s, d, bw)$, o algoritmo LMIR procura os K caminhos críticos contendo, pelo menos, K arestas críticas entre os outros pares origem-destino. Note que um caminho de capacidade mínima pode conter mais do que uma aresta crítica. Esta busca por arestas críticas é feita, encontrando-se os K caminhos com menor capacidade. Desde que o fluxo de um caminho é limitado pela aresta de menor capacidade, os K caminhos em questão contém as K arestas críticas.

Note que para se alcançar uma utilização balanceada dos caminhos entre um par origem-destino e maximizar o fluxo, as arestas críticas entre esse par devem ser evitadas se houver mais de um caminho entre a origem e o destino. Dessa forma, se a utilização da aresta crítica (u, v) não for evitada, o caminho P ao qual (u, v) pertence pode ser saturado e provocar uma utilização desbalanceada mesmo que existam outros caminhos disponíveis na rede. Além disso, se (u, v) pertencer ao conjunto de corte mínimo, ela deverá ser evitada pois dentre as arestas que reduzem o fluxo máximo, ela tende a ser saturada mais rapidamente.

Depois de encontrados os K caminhos de menor capacidade, todas as arestas que os compõem recebem pesos calculados de acordo com a equação (4.6). A utilização destes pesos permite que as arestas dos caminhos críticos recebam pesos inversamente proporcionais à sua capacidade residual.

Os pesos das arestas são computados como:

$$w(u, v) = \sum_{(s', d') \in P} \frac{f_G^{(s', d')}}{c_f(u, v)}, \quad \forall (u, v) \in E \quad (4.6)$$

onde $c_f(u, v)$ é a capacidade residual do enlace (u, v) e $f_G^{(s', d')}$ é o fluxo do caminho de menor capacidade entre s' e d' .

A última etapa do algoritmo é a execução do algoritmo de *Dijkstra* usando $w(u, v)$ como peso. Esta etapa permite a seleção de arestas não críticas ou arestas com baixa criticalidade. A seguir o algoritmo LMIR é apresentado.

Algoritmo 4 LMIR

ENTRADA

Um grafo residual $G' = (V, E')$ e $ri(a, b, bw)$, uma requisição de bw unidades de banda entre o par a, b .

SAIDA

Rota entre a e b com bw unidades de banda

LMIR

- 1: Encontre os K caminhos com menor capacidade $\forall (s', d') \in P$
 - 2: Calcule os pesos das arestas pertencentes aos caminhos encontrados, de acordo com a equação (4.6)
 - 3: Elimine as arestas com capacidade residual menor do que bw
 - 4: Execute o algoritmo de *Dijkstra* usando $w(u, v)$ como peso.
 - 5: Estabeleça o LSP entre a e b e atualize a capacidade das arestas.
-

O passo 1 do algoritmo LMIR é uma busca por caminhos de menor capacidade. O algoritmo usado para fazer isso é uma variação do algoritmo de *Dijkstra*, denominada "MenorCapacidade". Este algoritmo é mostrado a seguir:

No algoritmo MenorCapacidade v é um vértice qualquer da rede, D é um vetor que contém a capacidade mínima encontrada no caminho entre s e v . O vetor π contém o vértice predecessor de v e o vetor di indica a distância em número de arestas entre s e v . Q representa a lista de vértices cuja adjacência ainda não foi visitada, a função $extract_min(Q)$ retorna o elemento $u \in Q$ cujo valor $D[u]$ é o menor e ADJ representa a lista de adjacência do vértice v .

Assim como nos algoritmos de interferência apresentados nas seções 4.4 e 4.5, a identificação das arestas críticas é a tarefa que mais impacta na complexidade computacional do algoritmo LMIR. A atribuição dos pesos e o algoritmo de *Dijkstra* impactam da mesma forma a complexidade de todos os algoritmos de roteamento com interferência mínima em

Algoritmo 5 MenorCapacidade

```

1: for all (  $v \in |V|$ ) do
2:    $D[v] = \infty$ 
3:    $di[v] = \infty$ 
4:    $\pi[v] = NIL$ 
5:  $D[s] = 0.0$ 
6:  $di[s] = 0.0$ 
7:  $Q \leftarrow s$ 
8: while ( $Q$ ) do
9:    $u \leftarrow extract\_min(Q)$ 
10:  for all  $v \in ADJ[u]$  do
11:     $\gamma = min[c(u, v), D[u]]$ 
12:    if  $[(\gamma < D[v]) \vee ((\gamma == D[v]) \wedge (di[u] < di[v]))]$  then
13:       $D[v] = \gamma$ 
14:       $di[v] = di[u] + 1$ 
15:       $\pi[v] = u$ 
16:       $Q \leftarrow Q \cup v$ 

```

questão. Portanto, a diferença entre as complexidades dos três algoritmos de roteamento com interferência mínima apresentados nesta dissertação é devido à forma como as arestas críticas são identificadas. Como já mencionado, o algoritmo LMIR usa o algoritmo MenorCapacidade, para encontrar as arestas críticas. Uma análise da complexidade deste algoritmo é apresentada a seguir.

O passo 8 do algoritmo MenorCapacidade é executado $|V|$ vezes, já que todos os vértices são visitados. O passo 10 é executado $|V|$ vezes resultando em uma complexidade de $O(V^2)$. Além disso, o passo *extract_min(Q)* em MenorCapacidade tem complexidade igual a $O(E)$ no pior caso, se a fila Q de vértices cuja adjacência ainda não foi visitada for implementada como uma lista encadeada. Assim, a complexidade total do algoritmo é de $O(V^2 + E) = O(V^2)$ [CLR90]. Como cada par executa este algoritmo K vezes, a complexidade para identificar arestas críticas seria $O(K \cdot V^2) = O(V^2)$, enquanto que os outros algoritmos de interferência mínima apresentam, na etapa de identificação de arestas críticas, complexidade igual a $O(\min(V^{2/3}, E^{1/2}) \cdot E \cdot \log(V^2/E) \cdot \log(U))$.

Para mostrar que a complexidade do MenorCapacidade é inferior à do *maxflow* de

Goldberg et al, que é o algoritmo de *maxflow* mais rápido, a análise da complexidade de ambos será feita em separado para grafos densos e esparsos.

Em grafos densos, pode-se assumir $|E| = |V|^2$ assim, o algoritmo de Goldberg et al. ficaria igual a

$$O((V^{2/3}) \cdot V^2 \cdot \log(V^2/V^2) \cdot \log(U))$$

enquanto que a do MenorCapacidade tem complexidade igual a $O(V^2)$. Assumindo que o termo $\log(V^2/V^2)$ possui termos omitidos, já que a complexidade não pode ser zero, e que $\log(V^2/V^2) \cdot \log(U) > 1$, tem-se que

$$\text{maxflow de Goldberg et al.} = O(V^{2/3} \cdot V^2 = V^{8/3}) \text{ e } V^{8/3} > V^2 = \text{MenorCapacidade}$$

Assim, $V^2 = O(V^{8/3}) \Rightarrow \text{MenorCapacidade} = O(\text{maxflow de Goldberg et al.}) \square$

Em grafos esparsos pode-se assumir que $|E| = |V|$. Neste tipo de grafos, Q pode ser implementada como um *heap*[CLR90]. Assim, tem-se

$$\text{MenorCapacidade} = O(V \cdot \log V).$$

já que o passo *extract.min(Q)* tem complexidade igual a $O(\log V)$. Além disso, assumindo que $\log(V^2/E) \cdot \log(U) > 1$, tomando os dois primeiros fatores de $O(\min(V^{2/3}, E^{1/2}) \cdot E \cdot \log(V^2/E) \cdot \log(U))$, tem-se

$$\text{Goldberg et al.} = V^{1/2} \cdot V \text{ e } \text{MenorCapacidade} = O(V \cdot \log V)$$

Assim, descartando o fator V , presente nos dois algoritmos e aplicando \log tem-se

$$\text{MenorCapacidade} = \log(\log V) \text{ e } \text{Goldberg et al.} = \log V^{1/2} = 1/2 \cdot \log V$$

Então temos,

$$\log(\log(V)) = O(1/2 \cdot \log V) \Rightarrow \text{MenorCapacidade} = O(\text{Goldberg et al.})$$

□

O que mostra que tanto em grafos densos quanto em grafos esparsos, a detecção de arestas críticas no LMIR é menos custosa que nos algoritmos que usam *maxflow*.

4.7 Limite superior para K

Um aspecto importante para o bom desempenho do algoritmo LMIR é a escolha adequada do valor de K . Como ver-se-á no capítulo 6 a escolha do parâmetro interfere no número final de requisições rejeitadas. Isto acontece porque ao procurar K caminhos de capacidade mínima entre um par origem-destino, o algoritmo está tentando encontrar as arestas críticas, que no melhor caso, correspondem às arestas do corte mínimo para aquele par origem-destino.

Dessa forma, ao escolher um valor de K que esteja longe de corresponder ao número de arestas pertencentes ao corte mínimo, o algoritmo poderá identificar como críticas arestas que não deveriam ser classificadas como tal, atribuindo-lhes pesos. Supondo que uma aresta não crítica e seja classificada como crítica. Se e for erroneamente identificada como sendo crítica para todos os pares origem-destino poderá ser-lhe atribuído um peso maior do que o peso atribuído a uma aresta verdadeiramente crítica. Dessa forma, no passo 4 do algoritmo LMIR (algoritmo 4) a aresta e poderá ser evitada ao invés de uma aresta verdadeiramente crítica.

Nesta seção, pretende-se fornecer um limitante superior para o valor de K .

Definição [Wes96] 1 *Seja $G = (V, E)$ um grafo com conjunto de vértices $V(G)$ e conjunto de arestas $E(G)$. Um conjunto separador (ou desconectante) é definido como um conjunto $F \subseteq E(G)$ tal que $G - F$ tem mais do que um componente. A conectivi-*

dade nas arestas do grafo G , escrita como $\kappa'(G)$ é a cardinalidade do conjunto separador mínimo

Definição [Wes96] 2 *Dois caminhos entre u e v são internamente disjuntos se eles não possuem arestas internas (ou vértices) em comum.*

Teorema 1 *Seja $\lambda(u, v)$ o número de caminhos internamente disjuntos (ou simplesmente disjuntos) entre u e v , então*

$$\lambda(u, v) \leq \min\{d(u), d(v)\},$$

onde $d(u)$ e $d(v)$ representam os graus de u e v , respectivamente.

Prova Sejam P e P' dois caminhos internamente disjuntos entre u e v . Suponha, por contradição que $\lambda(u, v) > \min\{d(u), d(v)\}$. É fácil notar que se $\lambda(u, v) > \min\{d(u), d(v)\}$, significa que pelo menos alguma aresta incidente a u ou a v foi contabilizada em dois caminhos internamente disjuntos P e P' . Logo, se existe aresta comum a P e a P' eles não são internamente disjuntos, o que é uma contradição. ■

Depois de definidos os conceitos de conjunto separador e caminhos disjuntos, o teorema de Menger [Wes96] pode ser apresentado:

Teorema (Menger-1927) 2 *Sejam u e v dois vértices de um grafo G e $(u, v) \notin E(G)$ então $\max(\kappa'(G)) = \min(\lambda(u, v))$.*

A prova do teorema de Menger pode ser encontrada em [Wes96]. A relação do Lema 1 com o teorema de Menger, pode sugerir um limitante superior para K . Entretanto, como se pode ver no grafo G da Figura 4.3 tal idéia é inapropriada. No grafo G , existem apenas dois caminhos disjuntos nas arestas, porém, o conjunto de corte mínimo possui quatro arestas. Dessa forma, o algoritmo se o valor escolhido para K fosse igual a dois o algoritmo deixaria de identificar duas aresta críticas.

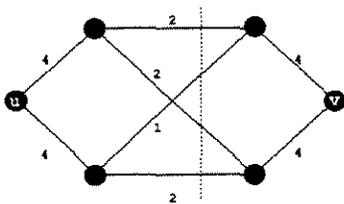


Figura 4.3: Quantidade de caminhos disjuntos menor que a quantidade de arestas no corte mínimo

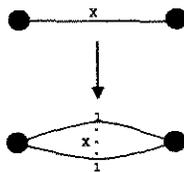


Figura 4.4: Transformação na rede original

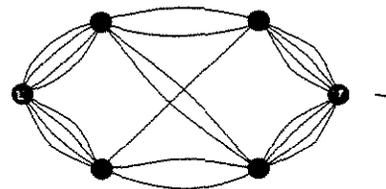


Figura 4.5: Rede da Figura 4.3 transformada

Entretanto, algumas mudanças podem ser feitas em G de forma que o teorema possa ser aplicado. Dado um grafo G com capacidades inteiras, pode-se criar um grafo G' criando, como ilustra a Figura 4.4, para cada aresta de capacidade x , x arestas de capacidade 1, como ilustrado na Figura 4.5

Assim, tem-se um segundo teorema relacionado fluxo máximo e o teorema de Menger (Teorema 2:

Teorema 3

$$\lambda(u, v) \geq \theta^{uv} = \min c(S, T) \geq \kappa'(G').$$

Desde que sempre $\kappa'(G') \geq \lambda(u, v)$, a igualdade é válida. Onde $\lambda(u, v)$ é o número de caminhos internamente disjuntos entre u e v , θ^{uv} é o valor do fluxo máximo entre u e v , $c(S, T)$ é a capacidade do corte mínimo entre S e T tal que $u \in S$ e $v \in T$ e $\kappa'(G)$ é o tamanho do conjunto separador mínimo de G

A prova do Teorema 3 pode ser encontrada em [Wes96]

Teorema 4 Seja $f_G^{(u,v)}$ o fluxo do caminho de menor capacidade entre u e v no grafo G , tem-se $n = f_G^{(u,v)} = \sum_{i=1}^n f_{G'}^{(u,v)}$

Prova Desde que em G' cada aresta tem capacidade 1, produzir um fluxo de n unidades de u a v equivale a mandar fluxo por n caminhos. Além disso, sabe-se que a capacidade de um caminho de u a v em G é limitada pela aresta de menor capacidade neste caminho.

Desde que o grafo G' é construído a partir do grafo G , usar uma aresta de capacidade r em G equivale a usar r arestas que ligam dois pontos em G' . Assim, a igualdade vale.

■

Pode-se então definir um limitante superior para K da seguinte forma:

Teorema 5 *Seja $K(\theta)$ um limitante superior para K , $\omega^{(u,v)}$ a capacidade do menor caminho entre u e v , $\theta^{(u,v)}$ o fluxo máximo entre u e v tem-se:*

$$K(\theta) = d(u), \text{ se } \theta^{(u,v)} = \sum_{\forall i:(u,i) \in E(G)} c(u, i) \quad (4.7a)$$

ou

$$K(\theta) = d(v), \text{ se } \theta^{(u,v)} = \sum_{\forall i:(i,v) \in E(G)} c(i, v) \quad (4.7b)$$

ou

$$K(\theta) = \left\lceil \frac{\theta^{(u,v)}}{\omega^{(u,v)}} \right\rceil \quad (4.7c)$$

Prova CASO 1:

$$\theta^{(u,v)} = \sum_{\forall i:(u,i) \in E(G)} c(u, i)$$

Sabe-se, pelo Teorema do Max-flow Min-cut [Wes96], que o fluxo máximo da rede é limitado por um corte de capacidade mínima. Dessa forma, se

$$\theta^{(u,v)} = \sum_{\forall i:(u,i) \in E(G)} c(u, i)$$

significa que as capacidades das arestas de u é que limitam o fluxo máximo.

CASO 2:

$$\theta^{(u,v)} = \sum_{\forall i:(i,v) \in E(G)} c(i, v)$$

Pelo raciocínio análogo ao do caso 1.

CASO 3:

Seja

$$n = \theta_{(G')}^{(u,v)} = \theta_{(G)}^{(u,v)},$$

sabe-se que em G' todos os caminhos têm capacidade 1, assim,

$$n = \theta_{(G)}^{(u,v)} = n \cdot f_{G'}^{(u,v)} = \sum_{i=1}^n f_{G'}^{(u,v)}$$

tomando-se ω e $K(\theta)$ tais que $\omega \cdot K(\theta) \leq n$ temos

$$n = \theta_{(G)}^{(u,v)} \geq K(\theta) \cdot \sum_{i=1}^{\omega} f_{G'}^{(u,v)}. \quad (4.8)$$

Porém, pelo teorema 4 e pela equação 4.8 o seguinte vale

$$K(\theta) \cdot \omega_{(G)}^{(u,v)} \leq \theta_{(G)}^{(u,v)} \quad (4.9)$$

. Como em G tanto $\theta_{(G)}^{(u,v)}$ quanto $\omega_{(G)}^{(u,v)}$ são conhecidos, podemos definir $K(\theta)$ como

$$K(\theta) = \left\lfloor \frac{\theta_{(G)}^{(u,v)}}{\omega_{(G)}^{(u,v)}} \right\rfloor$$

■

A equação 4.7a pode ser usada quando o fluxo máximo é limitado pelas arestas incidentes ao vértice de origem u ou v o que significa que tais arestas é que pertencem ao corte mínimo entre u e v . Por outro lado, a equação 4.7b pode ser usada se o fluxo máximo é limitado pelas arestas incidentes a v , neste caso estas são as arestas do corte mínimo. Se existe um corte mínimo diferente da vizinhança de u e v , a equação 4.7 pode ser usada.

Esta é uma premissa razoável e realista já que se pode assumir que os enlaces da rede têm capacidades próximas. Por outro lado, se a rede possuir enlaces cujas capacidades são muito inferiores aos demais, tais enlaces podem ser desconsiderados, assume-se que

sua capacidade é nula, até que os demais enlaces da rede alcancem capacidades próximas das suas.

Pode-se também perceber que neste último caso, a determinação do parâmetro K depende do conhecimento do fluxo máximo. Contudo, o algoritmo não precisa executar o cálculo do fluxo máximo a cada requisição. O cálculo do fluxo máximo entre u e v pode ser feito no vértice u de cada par origem-destino apenas uma única vez. Cada par origem-destino mantém uma variável que informa o valor do fluxo máximo entre todos os pares origem-destino da rede.

A partir daí, pode-se usar um raciocínio simples para acompanhar as mudanças no valor do fluxo máximo. Se uma requisição por um LSP com n unidades de banda for aceita entre u e v , o fluxo máximo entre este par será reduzido de n unidades, já que obrigatoriamente uma aresta pertencente ao conjunto de corte será utilizada. Daí, pode-se usar o protocolo de sinalização responsável pela atualização das capacidades dos enlaces para informar a todos os outros pares origem-destino o novo valor do fluxo máximo entre u e v . Caso a requisição seja rejeitada, nenhuma informação adicional precisa ser enviada.

Capítulo 5

Comparação Entre os Algoritmos de Roteamento com Interferência Mínima

No capítulo 4 foi apresentado o algoritmo LMIR. De forma a avaliar o seu desempenho em relação aos algoritmos de roteamento com interferência mínima também apresentados no capítulo 4, experimentos exaustivos de simulações foram realizados . Este capítulo apresenta os resultados obtidos através destas simulações. A seção 5.1 apresenta os resultados obtidos com carga de trabalho e topologia introduzidas na literatura ([KL00, WSC02]). A seção 5.2 apresenta os resultados obtidos de simulações feitas em redes de grande porte densas e esparsas. Por fim, a seção 5.4 apresenta as considerações finais a respeito dos resultados obtidos.

5.1 Validação através da comparação com os Resultados Encontrados na Literatura

Experimentos de simulação foram realizados a fim de se verificar a precisão e o desempenho do algoritmo LMIR. A topologia utilizada pode ser vista na Figura 5.1. Esta topologia foi utilizada em [WSC02] e em [KL00]. Os pares origem-destino $(S1, D1)$, $(S2, D2)$, $(S3, D3)$, $(S4, D4)$, $(S5, D5)$ são mostrados na figura e são os únicos pontos de entrada e saída de tráfego da rede. Os enlaces mais claros têm capacidade de 1200 unidades de banda e os enlaces mais escuros têm 4800 unidades. Estes valores simbolizam a taxa do STM-4 e do STM-16, respectivamente.

Cada enlace na figura é bidirecional, ou seja, representa dois enlaces de igual capacidade e sentidos opostos. As requisições entre os pares foram geradas aleatoriamente com distribuição uniforme e a quantidade de banda requisitada é uniformemente distribuída no intervalo $[1, 4]$.

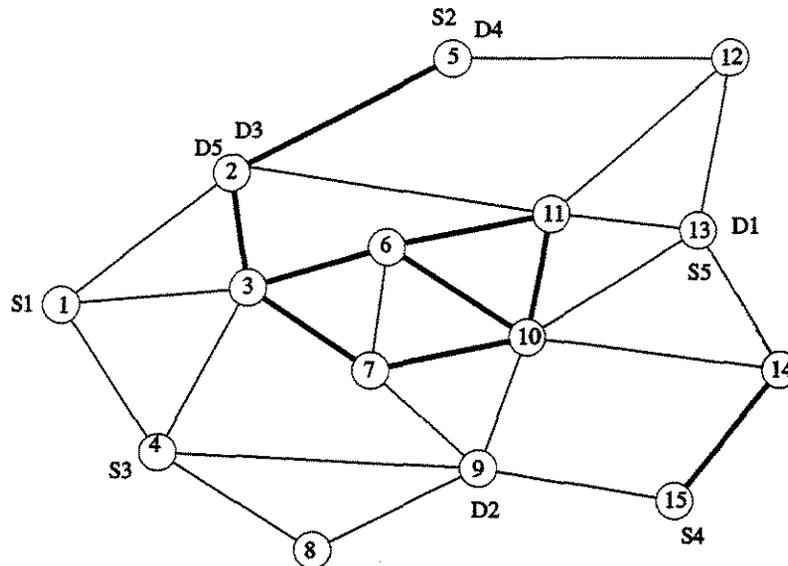


Figura 5.1: Topologia usada nas simulações

Nas legendas das figuras apresentadas a seguir, os algoritmos de interferência mínima serão representados por WSC [WSC02] MIRA [KL00] e LMIR. No algoritmo LMIR, o

valor de K utilizado nas simulações foi igual a 5. Assim como em [KL00] e em [WSC02], além dos algoritmos de interferência mínima apresentados, mais dois algoritmos foram utilizados nas simulações para que comparações fossem feitas também com algoritmos que não levam em consideração o fenômeno da interferência. O *Minimum Hop Algorithm (MHA)* é uma implementação do algoritmo de *Dijkstra* e o *Widest Shortest Path Algorithm (WSP)* encontra o caminho com maior capacidade da rede. Se mais de um caminho com mesma capacidade existir, o algoritmo WSP devolve o caminho com o menor número de arestas.

Nestes experimentos, assume-se que os LSPs têm vida longa, ou seja, uma vez aceito, ele ocupa os recursos até o final dos experimentos. Foram geradas aleatoriamente 8.000 requisições entre os cinco pares já mencionados.

Sabe-se que quanto menor for a taxa de diminuição do fluxo máximo menor será a probabilidade de bloqueio. A Figura 5.2 mostra a diminuição do fluxo máximo da rede em função do número de chegadas de requisições. Pode-se ver que a taxa de diminuição do fluxo total da rede é menor para os algoritmos de interferência mínima. Os algoritmos WSP e MHA têm uma redução mais acentuada pois utilizam arestas críticas indiscriminadamente, causando a redução do *maxflow* entre diversos pares origem-destino o que demonstra a importância dos algoritmos de interferência mínima. O LMIR tem uma taxa de diminuição do fluxo máximo inferior à do WSC e igual à do MIRA até as 3.000 requisições. A partir deste ponto, tem um decaimento levemente mais acentuado que os dois até as 5.000 requisições. A partir das 5.000 requisições, o comportamento de todos os algoritmos torna-se bem parecido, quase linear.

A Figura 5.3 mostra o número de requisições rejeitadas como função do número de requisições que chegam no sistema. O MHA é o algoritmo que rejeita o maior número de conexões para um menor número de requisições demandadas, o que se verifica logo após 3.000 requisições. Isso acontece porque ele sempre escolhe o caminho mais curto, saturando rapidamente as arestas que fazem parte deste caminho. O WSP começa a rejeitar conexões a partir 5.000 conexões. Entretanto, por não levar em consideração a

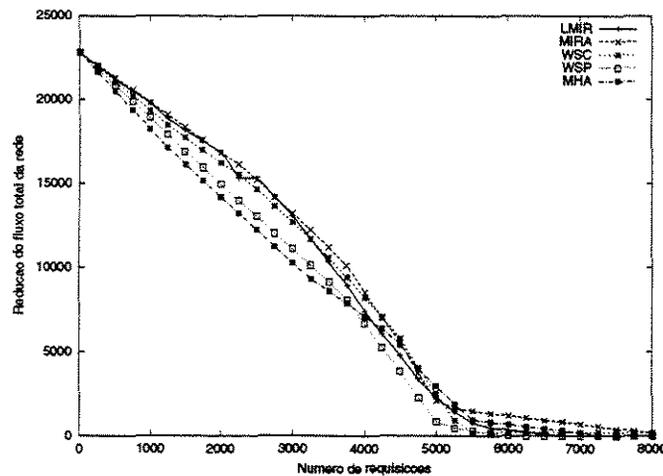


Figura 5.2: Largura de banda total entre todos os pares origem-destino da rede.

criticalidade das arestas, ele pode escolher caminhos que possuem arestas críticas de vários pares, provocando a saturação das mesmas. Entre os algoritmos de interferência mínima, o LMIR e o MIRA são os algoritmos que apresentam um menor número de conexões rejeitadas.

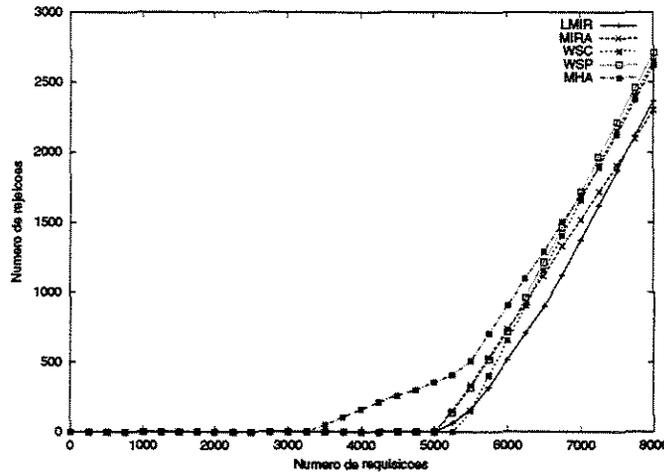


Figura 5.3: Número de rejeições.

Conforme mencionado, interferência é a diminuição do fluxo de um par origem-destino devido ao roteamento de requisições entre outros pares. A Figura 5.4 mostra um experimento realizado para avaliar a interferência sofrida pelo par $(S1, D1)$ quando requisições entre outros pares são roteadas na rede. As requisições de conexão chegam aleatoriamente

para os pares $(S2, D2)$, $(S3, D3)$, $(S4, D4)$ e $(S5, D5)$. Pretende-se verificar a influência destas requisições no fluxo de $(S1, D1)$.

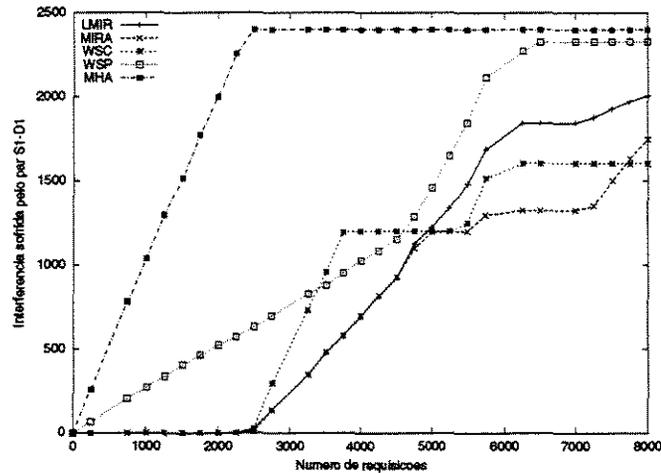


Figura 5.4: Interferência sofrida pelo par S1-D1.

A Figura 5.4 mostra a interferência no par $(S1, D1)$ quando conexões são roteadas entre os outros pares. É fácil ver que os algoritmos MHA e WSP, que não levam em consideração o fenômeno de interferência, provocam uma diminuição do *maxflow* entre $(S1, D1)$ antes mesmo de 1.000 conexões. Por usar sempre o menor caminho, o algoritmo MHA provoca a saturação logo após 2.000 conexões. A partir daí o caminho torna-se saturado e não existe mais diminuição. Muito embora provoque interferência desde as primeiras conexões, o algoritmo WSP demora mais para saturar do que o MHA. Isto acontece porque ele escolhe outros caminhos com maior capacidade à medida que vai saturando. A maior acentuação da interferência quando se usa o algoritmo WSP é percebida entre as 4.500 e 6.500, onde existe uma estagnação da curva até 8.000 requisições.

Entre os algoritmos de interferência mínima, a interferência só é percebida após a chegada de 2.500 requisições, ou seja, não existe redução do *maxflow* entre $(S1, D1)$ até esta chegada. A partir desta chegada, o algoritmo LMIR consegue um desempenho melhor, juntamente com o algoritmo MIRA, até bem próximo da chegada de 5.000 requisições. Mesmo causando uma diminuição acentuada no *maxflow* de $(S1, D1)$ logo após as 2.500 requisições, o algoritmo WSC consegue o melhor desempenho a longo prazo pois conse-

	WSC	MIRA	LMIR $K=1$	LMIR $K=2$	LMIR $K=3$
Time	7.23s	7.24s	5.017s	5.67s	6.54s
		LMIR $K=4$	LMIR $K=5$	LMIR $K=6$	
Time		6.94s	7.01s	7.14s	

Tabela 5.1: LMIR: tempo de execução para 8,000 requisições.

que uma maior estabilização do que o algoritmo LMIR e o algoritmo MIRA. O LMIR e o MIRA, têm um desempenho muito parecido até próximo da chegada das 5.000 requisições, o que talvez possa ser explicado pelo fato de que até 5.000 requisições, as K arestas críticas vistas pelo algoritmo LMIR formem conjuntos de corte mínimo. Assim sendo, as arestas críticas seriam as mesmas para os dois algoritmos. Depois de 5.000 requisições o algoritmo MIRA começa a estabilizar encontrando caminhos que não afetem $(S1, D1)$ e o algoritmo LMIR permanece causando interferência, mesmo que muito branda até as 8.000 requisições. Após 8.000 requisições, ainda existe uma significativa distância entre o LMIR e o WSP que não leva em consideração interferência que possui o melhor desempenho. Note que neste experimento, só é levada em consideração a interferência no par $(S1, D1)$, o que pode não refletir o desempenho global do algoritmo. É importante ressaltar que o LMIR consegue melhor desempenho que o WSC quando a interferência entre todos os pares é levada em consideração, o que pode ser visto na Figura 5.2.

Como visto no capítulo 4, o algoritmo LMIR encontra os K caminhos de menor capacidade para cada par. A escolha do valor deste parâmetro pode alterar o desempenho do algoritmo. A Figura 5.5 mostra o número de conexões rejeitadas em função de K . Pode-se ver que o número mínimo de conexões rejeitadas, nesta topologia, foi alcançado com $K = 5$. Depois disso, os caminhos começam a se sobrepor e a identificação das arestas críticas passa a ser uma tarefa difícil.

A Tabela 5.1 mostra o tempo de execução dos algoritmos com 8.000 requisições entre todos os pares. Os valores foram obtidos utilizando o comando *time* do Linux, usando máquinas Intel Celeron 1.2 GHz, 256 MB de RAM.

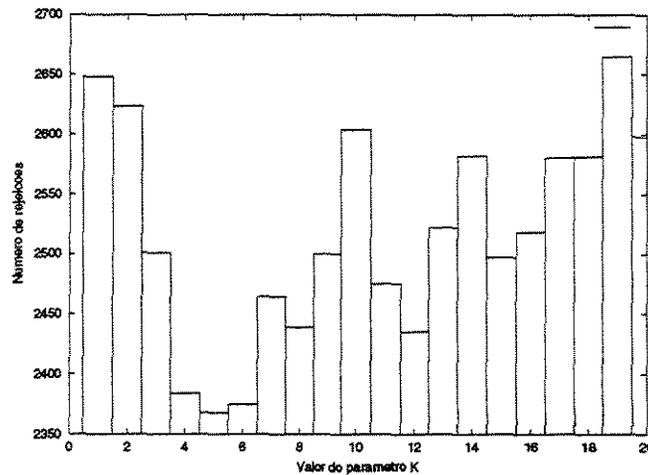


Figura 5.5: Influência do parâmetro K no número de conexões rejeitadas.

Como era de se esperar, pode-se perceber que o tempo gasto na execução do algoritmo LMIR é menor que o tempo de execução do algoritmo WSC e do algoritmo MIRA. Além disso, o algoritmo LMIR tem a vantagem de conseguir oferecer um compromisso velocidade \times precisão dependente da escolha do parâmetro K . A partir de $K = 7$, na topologia apresentada, os caminhos começam a se sobrepôr e a detecção das arestas críticas torna-se mais difícil, deteriorando o desempenho do LMIR. Os algoritmos WSP e MHA têm seu tempo de execução em torno de 4 segundos. Muito embora mais rápidos, estes algoritmos por não levarem em consideração o fenômeno da interferência, provocam um número excessivo de rejeições e por essas razões foram omitidos.

5.2 Resultados para redes de grande porte

Nesta seção, serão apresentados resultados obtidos em redes densas e esparsas com 30, 40 e 50 vértices. As redes são modeladas como grafos conexos com capacidade nas arestas. Os grafos foram gerados aleatoriamente através do método de Waxman [ZCB96], [Wax88] e os parâmetros são sumarizados na Tabela 5.2. No método de Waxman, a probabilidade

	$ V $	Tipo	α	β	L	$ E $
1	30	Denso	0.4	0.4	141	222
2	30	Esparso	0.2	0.2	141	118
3	40	Denso	0.4	0.4	141	354
4	40	Esparso	0.2	0.2	141	170
5	50	Denso	0.4	0.4	141	544
6	50	Esparso	0.2	0.2	141	302

Tabela 5.2: Parâmetros usados na geração aleatória dos grafos.

de existência de uma aresta entre u e v é dada por

$$P(u, v) = \alpha e^{-d/(\beta L)},$$

onde $0 < \alpha, \beta \leq 1$ são parâmetros do modelo d é a distância euclidiana entre u e v e L é a distância euclidiana máxima entre quaisquer dois vértices do grafo. O grafo é não direcionado, o que representa enlaces bidirecionais e as arestas têm capacidade de 1200 ou 4800 unidades de banda, atribuídas aleatoriamente, com a mesma probabilidade.

Nas legendas das figuras apresentadas a seguir, os algoritmos de interferência mínima serão representados por WSC [WSC02], MIRA [KL00] e LMIR. As comparações em redes de grande porte é restrita aos algoritmos de interferência mínima por apresentarem na seção 5.1 resultados muito superiores aos demais. Mais uma vez, no algoritmo LMIR, o valor de K utilizado nas simulações foi igual a 5.

Mais uma vez, nos experimentos, assume-se que cada LSP tem vida longa, ou seja, uma vez aceito, ele ocupa os recursos até o final dos experimentos. Foram geradas aleatoriamente, com distribuição uniforme, 30.000 requisições entre os cinco pares origem-destino $(S1, D1)$, $(S2, D2)$, $(S3, D3)$, $(S4, D4)$, $(S5, D5)$. Novamente, as requisições entre os pares origem-destino foram geradas aleatoriamente e a quantidade de banda requisitada em cada requisição é uniformemente distribuída no intervalo $[1, 4]$.

Cada experimento foi realizado com 10 sementes de números aleatórios diferentes e

suas médias foram computadas e apresentadas nas figuras. O intervalo de confiança de tais médias foi obtido usando-se um nível de confiança de 95%. Entretanto, a largura de tais intervalos é muito pequena e por isso não são visualizados.

5.2.1 Redes densas de 30 vértices

Esta seção apresenta os resultados numéricos dos experimentos realizados em redes densas (Tabela 5.2) com 30 vértices. Os resultados serão discutidos a seguir:

Na Figura 5.6, observa-se a redução do fluxo máximo entre os pares origem-destino mencionados. Pode-se constatar que o desempenho de todos os algoritmos avaliados é muito semelhante

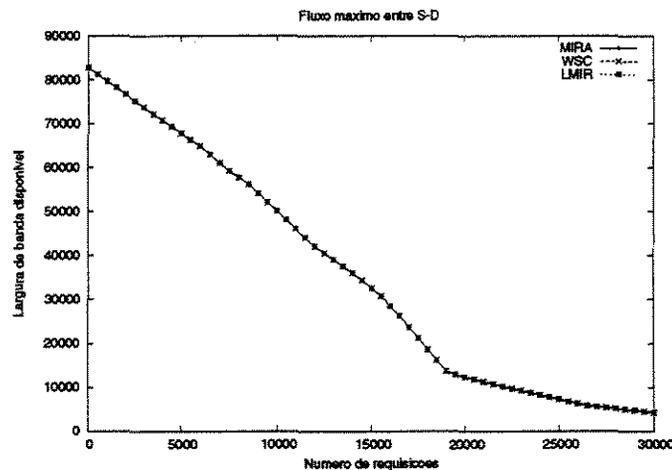


Figura 5.6: Largura de banda total (redes densas de 30 vértices).

A Figura 5.7 mostra em escala ampliada as últimas 1.000 requisições da Figura 5.6. Pode-se constatar a semelhança dos algoritmos pela sobreposição dos intervalos de confiança.

Como já mencionado, conjectura-se que a minimização da redução do fluxo máximo entre pares origem-destino faz com que a probabilidade de bloqueio do algoritmo seja menor. É isso que se pode observar na Figura 5.8. Pode-se perceber que a primeira rejeição de estabelecimento de LSP se dá após 15.000 requisições e considerou-se os três algoritmos. Observa-se também que o algoritmo WSC a partir de 15.000 requisições

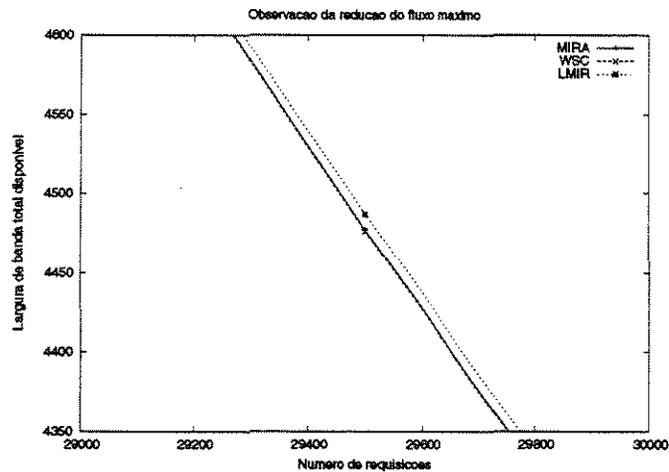


Figura 5.7: Largura de banda total (Vista ampliada).

rejeita sempre um número maior de conexões dos que os algoritmos MIRA e LMIR. Estes últimos tem um comportamento muito semelhante, sendo que a taxa de rejeições do LMIR é levemente maior do que a do algoritmo MIRA. Entretanto, vê-se que por volta das últimas 100 requisições, o algoritmo LMIR apresenta um número menor de rejeições.

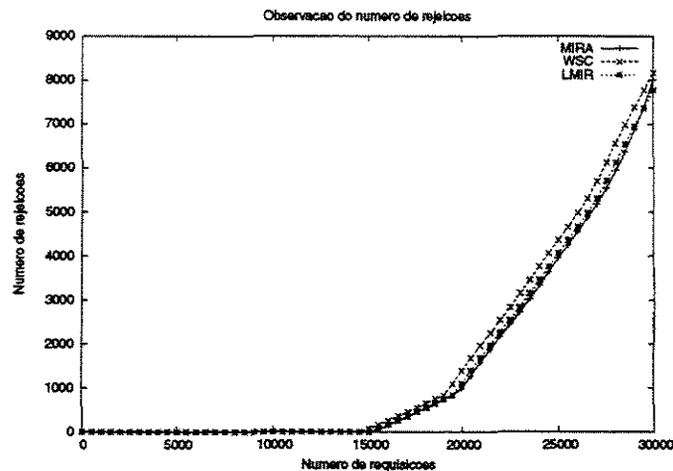


Figura 5.8: Número de requisições rejeitadas (redes densas de 30 vértices).

A Figura 5.9 ilustra a interferência sofrida por um par origem-destino. O experimento foi realizado isolando-se um par, ou seja, não há requisições de estabelecimento de LSPs para esse par enquanto requisições chegam para os outros quatro pares. Os resultados mostram que não há interferência.

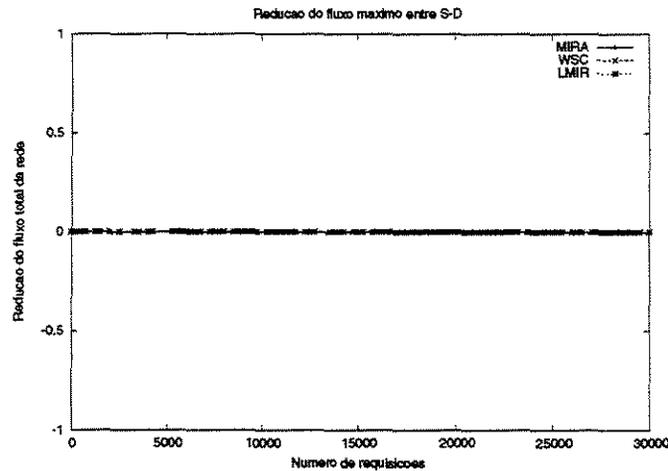


Figura 5.9: Interferência sofrida pelo par S1-D1 (redes densas de 30 vértices).

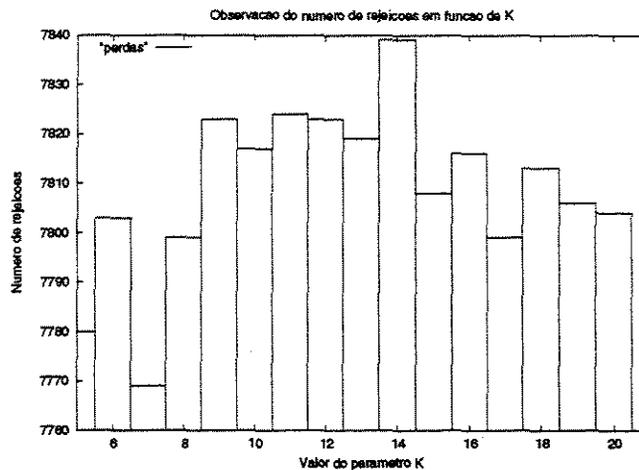


Figura 5.10: Número de perdas em função de K (redes densas de 30 vértices).

5.2.2 Redes esparsas de 30 vértices

Esta seção apresenta os resultados numéricos dos experimentos realizados em redes esparsas (Tabela 5.2) com 30 vértices.

A Figura 5.11 ilustra a redução do fluxo máximo entre os pares origem-destino. Percebe-se que o comportamento dos três algoritmos, WSC, MIRA e LMIR é muito semelhante. Pode-se ver na escala ampliada da Figura 5.12 que a diferença entre os algoritmos é quase nula, podendo ser considerada desprezível.

A Figura 5.13 apresenta o número de rejeições experimentadas pelos algoritmos no

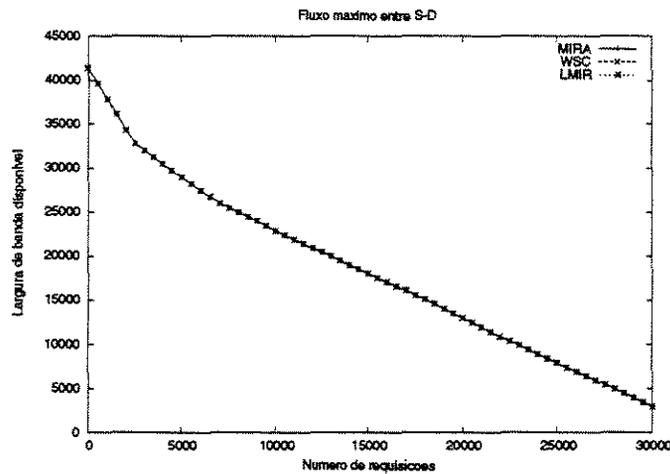


Figura 5.11: Largura de banda total (redes esparsas de 30 vértices).

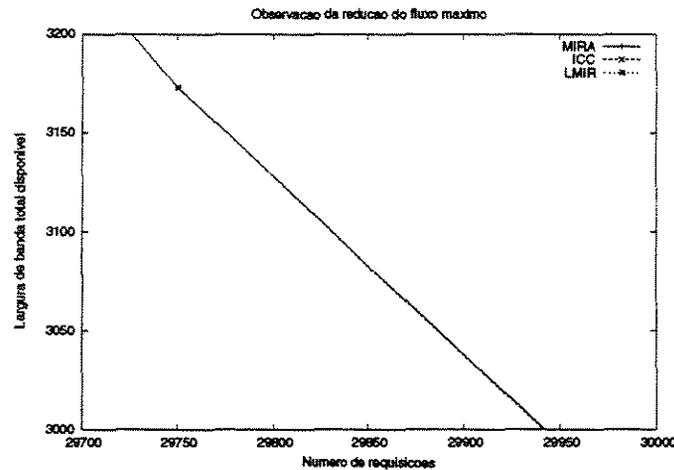


Figura 5.12: Largura de banda total (redes esparsas de 30 vértices).

grafo esparso de 30 vértices. Pode-se perceber que o número de rejeições é bem maior que nas redes densas de 30 vértices. Isto acontece devido ao menor número de caminhos existentes entre cada par origem-destino, o que obviamente diminui o número de alternativas que cada algoritmo possui. As primeiras rejeições acontecem antes da requisição 5.000. Dentre os três algoritmos sob avaliação, o último a rejeitar conexão em redes esparsas de 30 vértices foi o LMIR, o penúltimo o MIRA. Observa-se também que o WSC apresenta um número maior de rejeições, seguido do MIRA até próximo de 17.000 requisições quando o LMIR começa a rejeitar mais conexões. Entretanto, nas últimas 5.000

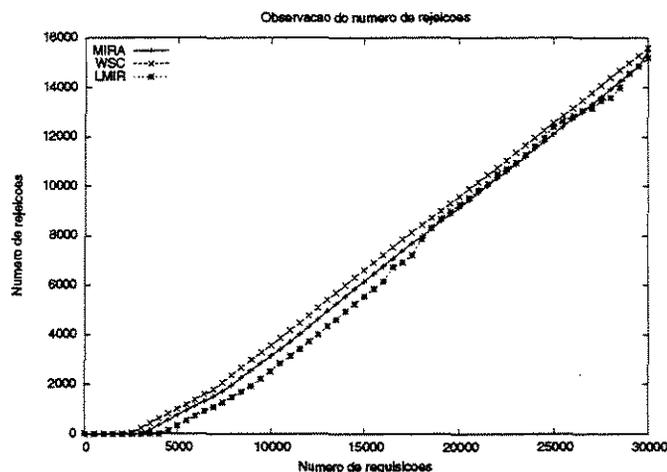


Figura 5.13: Número de requisições rejeitadas (redes esparsas de 30 vértices).

requisições, o LMIR consegue aceitar um número maior de conexões do que o MIRA.

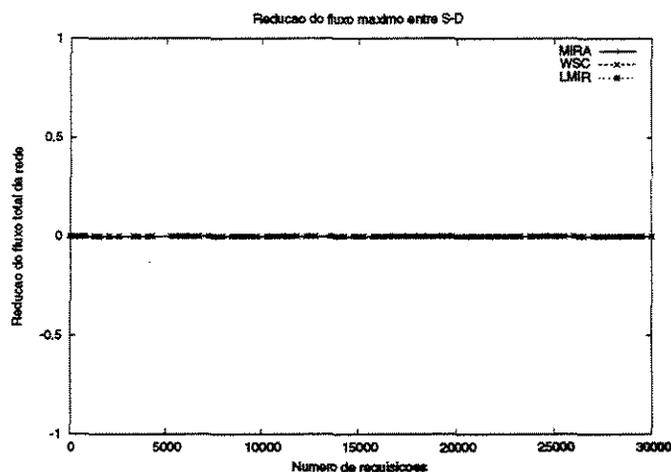


Figura 5.14: Interferência sofrida pelo par S1-D1 (redes esparsas de 30 vértices).

A Figura 5.14 ilustra a interferência sofrida pelo par origem-destino ($S1 - D1$). De novo, não há interferência.

5.2.3 Redes densas com 40 vértices

Esta seção apresenta os resultados numéricos dos experimentos realizados em redes densas (Tabela 5.2) com 40 vértices.

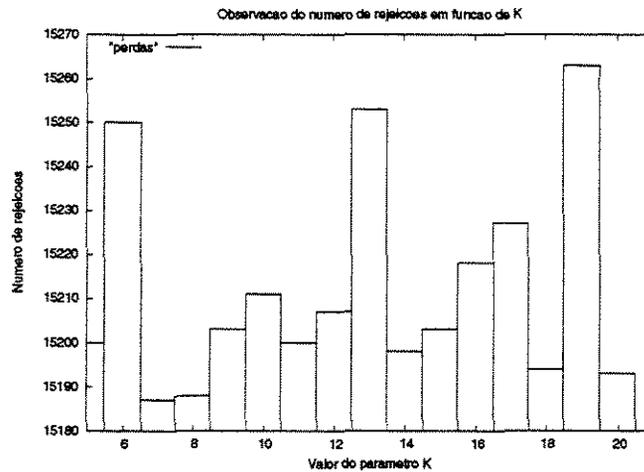


Figura 5.15: Número de perdas em função de K (redes esparsas de 30 vértices).

A Figura 5.16 apresenta a redução do fluxo máximo entre os pares origem-destino. Pode-se observar que o comportamento dos algoritmos LMIR e WSC é muito parecido. Com uma clara degradação do desempenho em relação ao MIRA. A partir de aproximadamente 12.500 requisições, a redução do fluxo máximo entre os pares origem-destino no algoritmo LMIR e WSC é muito mais acentuada do que no algoritmo MIRA. Isto acontece possivelmente por falha na detecção das arestas que pertencem ao corte mínimo dos outros pares origem-destino. Como já mencionado, o uso de tais arestas provoca a diminuição do fluxo máximo entre aqueles pares.

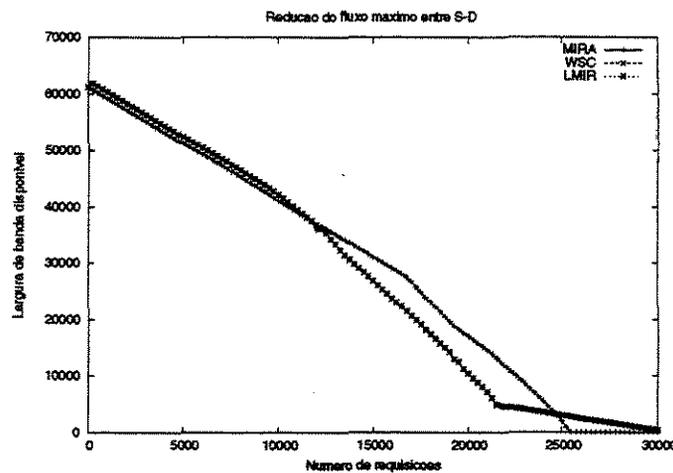


Figura 5.16: Largura de banda total (redes densas de 40 vértices).

O efeito da rápida diminuição do fluxo máximo pode ser percebido no número de conexões rejeitadas pelos algoritmos LMIR e WSC. Na Figura 5.17, vê-se uma clara desvantagem destes em relação ao algoritmo MIRA. Até próximo de 20.000 requisições, nota-se que o número de conexões rejeitadas nos três algoritmos é muito parecido. A partir daí, vê-se que o algoritmo MIRA começa a rejeitar um número menor de conexões do que o LMIR e o WSC.

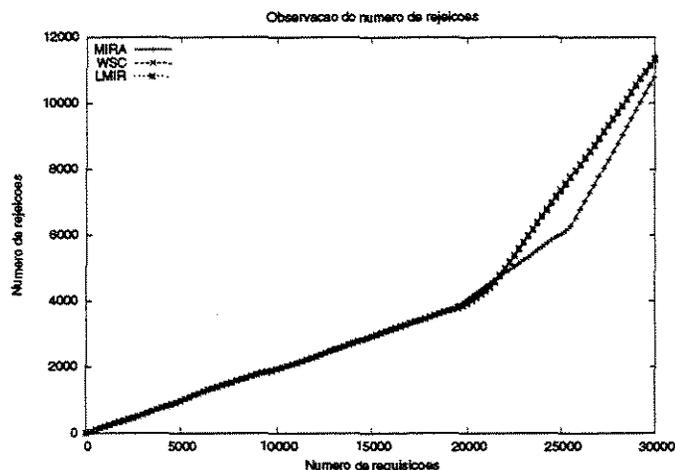


Figura 5.17: Número de requisições rejeitadas (redes densas de 40 vértices).

Em relação ao fenômeno da interferência, novamente verificou-se a inexistência do mesmo como pode ser observado na Figura 5.18.

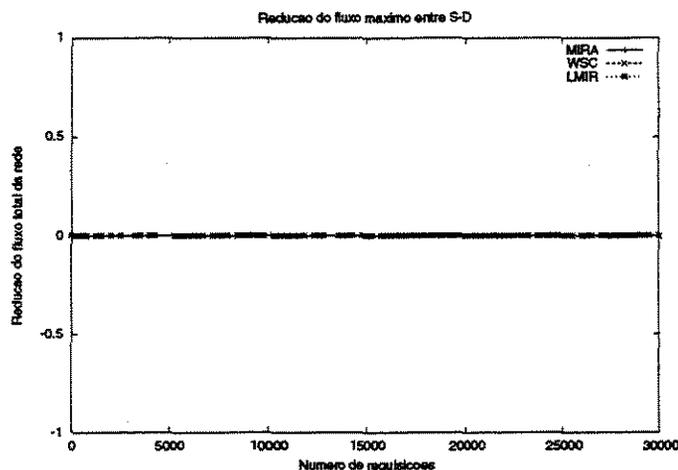


Figura 5.18: Interferência sofrida pelo par S1-D1 (redes densas de 40 vértices).

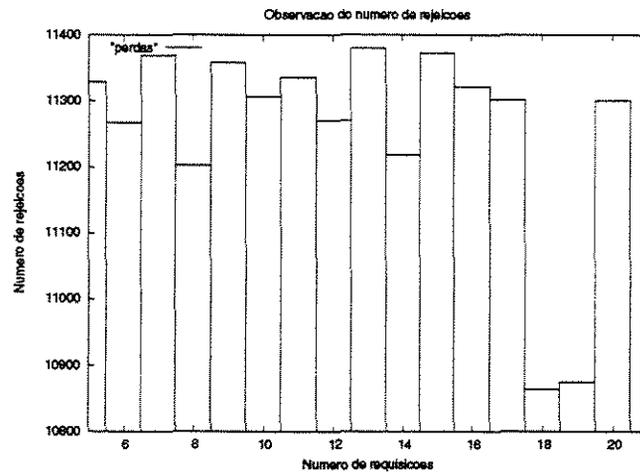


Figura 5.19: Número de perdas em função de K (redes densas de 40 vértices).

5.2.4 Redes esparsas de 40 vértices

Esta seção apresenta os resultados numéricos dos experimentos realizados em redes esparsas (Tabela 5.2) com 40 vértices.

Na Figura 5.20, observa-se um decréscimo acentuado do fluxo máximo. Esse decréscimo pode ser explicado pelo fato de que o grau dos vértices é pequeno, o que implica em uma pequena quantidade de caminhos alternativos entre os pares origem-destino. Após 6.000 requisições o fluxo máximo total é zero em todos os algoritmos, o que, como pode-se ver na Figura 5.21, tem implicações no desempenho dos algoritmos

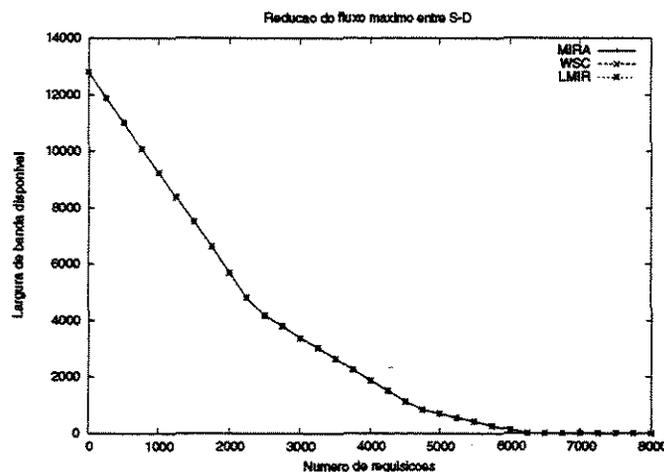


Figura 5.20: Largura de banda total (redes esparsas de 40 vértices).

Pode-se ver claramente na Figura 5.21, o alto número de requisições rejeitadas. Outro fato notório é o comportamento similar de todos os algoritmos. Por volta de 2.500 requisições acontecem as primeiras rejeições e logo após 5.000 requisições o comportamento do gráfico é quase linear denotando que quase todas as requisições são rejeitadas. Fazendo um paralelo com a Figura 5.20 nota-se que, aproximadamente, neste mesmo período, o fluxo máximo entre todos os pares é quase nulo, ou seja, existe um número muito pequeno de caminhos com a capacidade diferente de 0. No entanto, estes caminhos podem ter a capacidade inferior à requerida, resultando em rejeição da requisição.

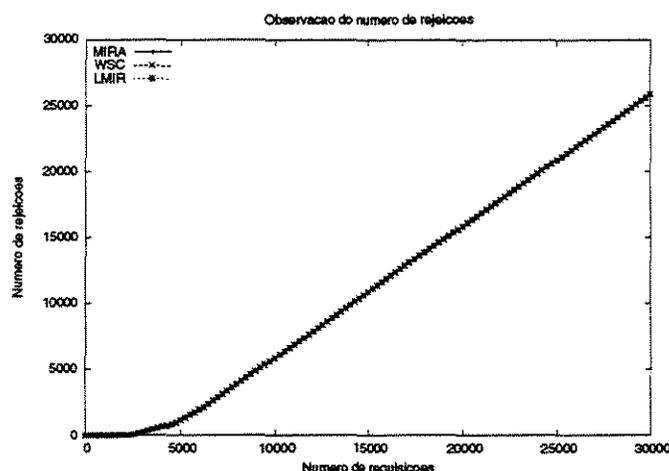


Figura 5.21: Número de requisições rejeitadas (redes esparsas de 40 vértices).

Não foi percebida nenhuma interferência, como se pode verificar na Figura 5.22.

5.2.5 Redes densas com 50 vértices

Esta seção apresenta os resultados numéricos dos experimentos realizados com redes densas (Tabela 5.2) com 50 vértices.

Como mostrado na Tabela 5.2 as redes densas com 50 vértices possuem 544 arestas, o que implica em um alto número de caminhos. Com experimentos realizados usando 30.000 requisições, não foi observada nenhuma rejeição de conexão. Assim, de forma que a rede fosse submetida a uma situação de sobrecarga foram atribuídas a todas as arestas desta rede capacidades iguais a 1.200 unidades de banda.

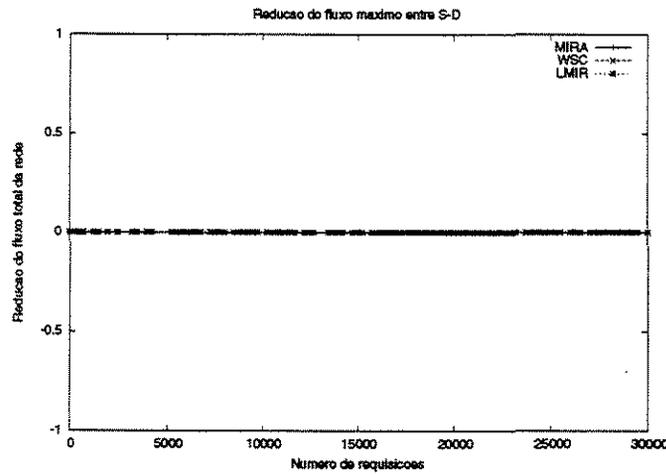


Figura 5.22: Interferência sofrida pelo par S1-D1 (redes esparsas de 40 vértices).

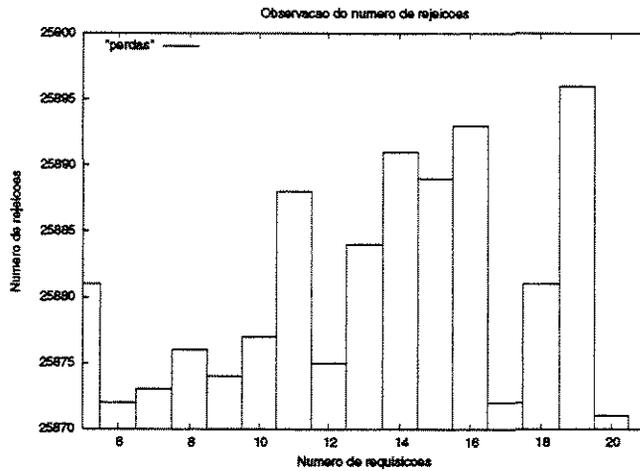


Figura 5.23: Número de perdas em função de K (redes esparsas de 40 vértices).

Pode-se perceber na Figura 5.24 que o comportamento dos algoritmos WSC, MIRA e LMIR é muito parecido até 20.000 requisições. Somente a partir deste ponto, é que o algoritmo MIRA consegue uma redução do fluxo máximo um pouco mais amena que os algoritmos WSC e LMIR.

Na Figura 5.25, é apresentado o número de rejeições em função do número de requisições em redes densas com 50 vértices. Observa-se que o algoritmo LMIR rejeita a primeira requisição aproximadamente depois de 17.500 requisições, ao passo que os algoritmos WSC e MIRA rejeitam a sua primeira requisição próximo às 16.000 requisições. No

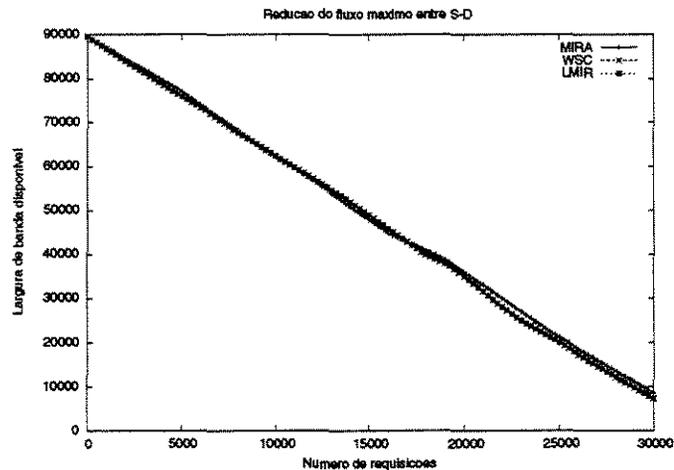


Figura 5.24: Largura de banda total (redes densas de 50 vértices).

intervalo de [16.000, 25.000], os algoritmos MIRA e WSC rejeitam um número maior de requisições do que o LMIR. Entretanto, a partir deste ponto, os três algoritmos produzem um número muito próximo de conexões rejeitadas.

Pode-se ainda notar que a quantidade de rejeições neste tipo de redes é substancialmente inferior às outras redes utilizadas nas simulações. Isto pode ser explicado pela grande quantidade de caminhos alternativos pelos quais os algoritmos podem estabelecer os LSPs.

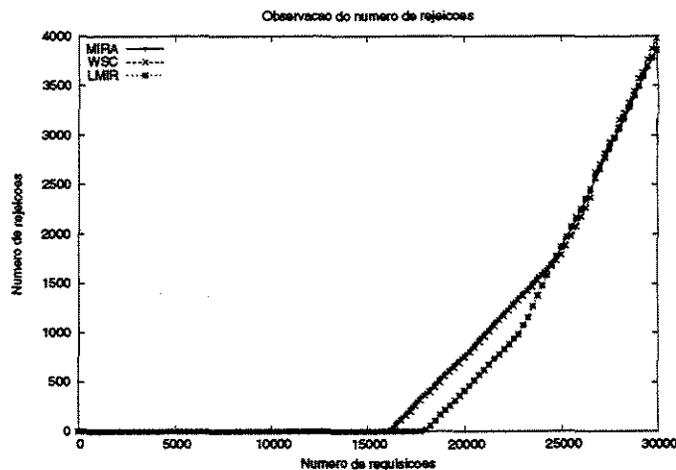


Figura 5.25: Número de requisições rejeitadas (redes densas de 50 vértices).

Não foi percebida nenhuma interferência, como se pode verificar na Figura 5.26.

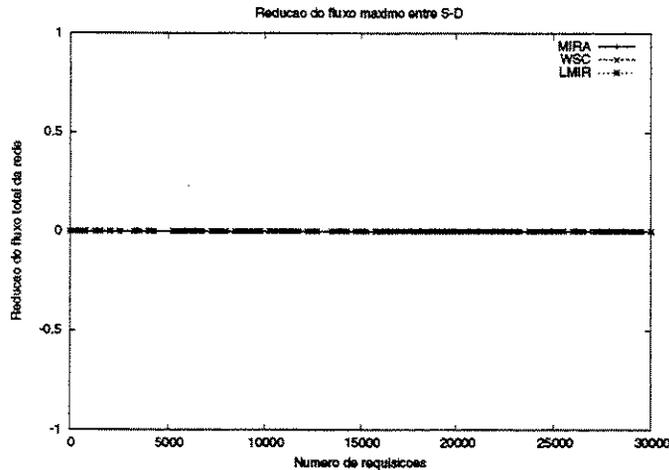


Figura 5.26: Interferência sofrida pelo par S1-D1 (redes densas de 50 vértices).

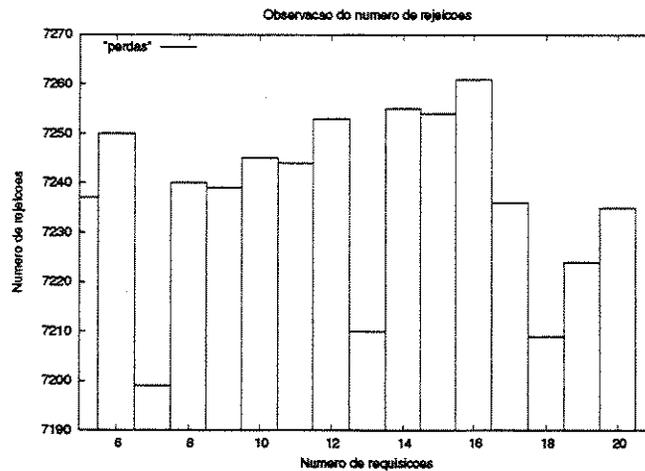


Figura 5.27: Número de perdas em função de K (redes densas de 50 vértices).

5.2.6 Redes esparsas com 50 vértices

Esta seção apresenta os resultados numéricos dos experimentos realizados em redes esparsas (Tabela 5.2) com 50 vértices.

A Figura 5.28 mostra a redução do fluxo máximo em redes esparsas com 50 vértices. Nota-se, claramente, uma menor redução do fluxo máximo provocada pelo algoritmo LMIR em relação aos algoritmos WSC e MIRA. Já a partir de aproximadamente 4.000 requisições, o algoritmo WSC apresenta uma redução mais acentuada que os algoritmos MIRA e LMIR, que se comportam de maneira semelhante até próximo às 6.000 requisições,

quando pode-se verificar uma diminuição mais lenta do fluxo máximo do algoritmo LMIR até 14.000 requisições quando o fluxo máximo torna-se nulo.

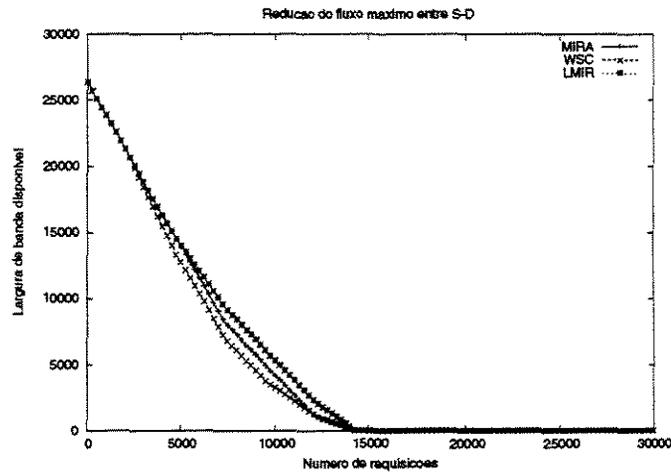


Figura 5.28: Largura de banda total (redes esparsas de 50 vértices).

Na Figura 5.29 é possível observar o número de requisições rejeitadas pelos algoritmos WSC, MIRA e LMIR em redes esparsas com 50 vértices. Pode-se observar que as primeiras requisições são rejeitadas após 5.000 requisições. No intervalo [5.000, 10.000] os algoritmos têm um comportamento muito parecido, o que começa a mudar a partir de 10.000 requisições. Vê-se que o algoritmo LMIR rejeita durante todo o experimento um número menor de requisições chegando a aproximadamente 19.500 requisições rejeitadas. O segundo melhor algoritmo neste experimento foi o algoritmo MIRA com 20.000 requisições rejeitadas ao final do experimento, seguido pelo WSC com pouco menos de 20.500 rejeições.

Não foi percebida nenhuma interferência, como se pode verificar na Figura 5.30.

5.3 Limitante superior de K nos exemplos numéricos

Um aspecto importante para o bom desempenho do algoritmo LMIR é a escolha adequada do valor de K como pode ser visto nas Figuras 5.10, 5.15, 5.19, 5.23, 5.27, 5.31. Esta escolha interfere no número final de requisições rejeitadas. Isto acontece porque ao

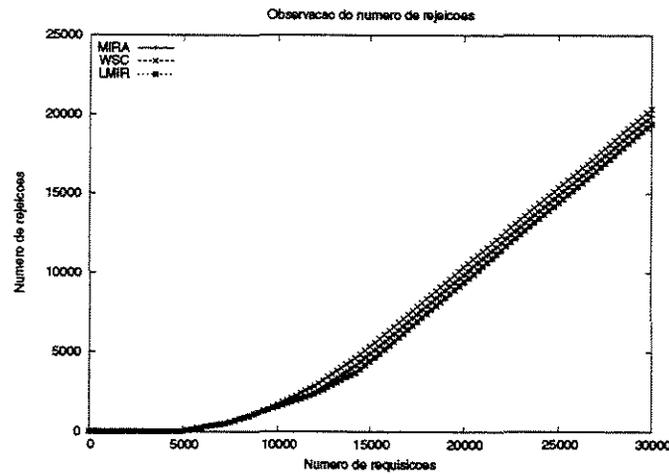


Figura 5.29: Número de requisições rejeitadas (redes esparsas de 50 vértices).

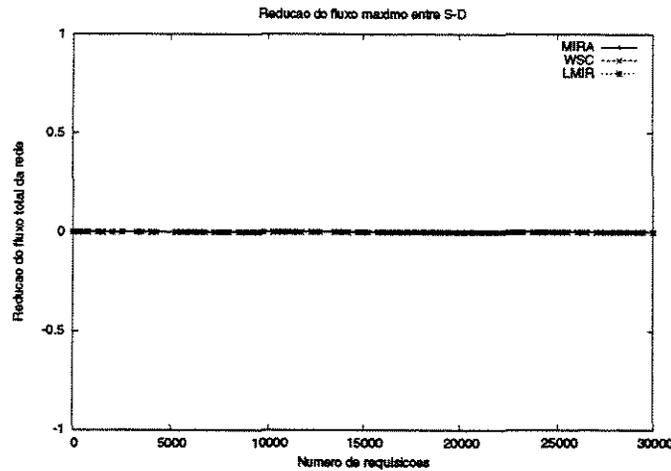


Figura 5.30: Interferência sofrida pelo par S1-D1 (redes esparsas de 50 vértices).

procurar K caminhos de capacidade mínima entre um par origem-destino, o algoritmo está tentando encontrar as arestas críticas, que no melhor caso, correspondem às arestas do corte mínimo para aquele par origem-destino.

Na Figura 5.6 pode-se ver que a soma da largura de banda entre todos cinco pares origem-destino indica, quando o número de requisições é zero, 80.000 unidades de banda disponíveis. Este valor, em média, representa um fluxo máximo de 16.000 unidades de banda entre cada par. Como os enlaces da rede têm capacidade mínima de 1.200 unidades de banda e a capacidade do menor caminho é limitada pela sua aresta de menor capacidade

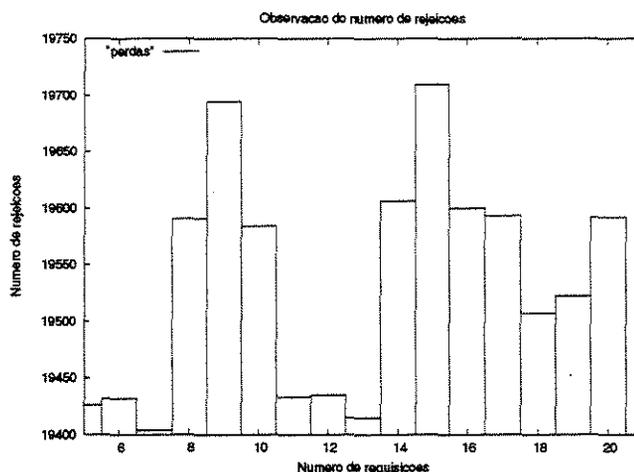


Figura 5.31: Número de perdas em função de K (redes esparsas de 50 vértices).

	50 (D)	50 (E)	40 (D)	40 (E)	30 (D)	30 (E)
$K(\theta)$	15	5	10	3	13	7

Tabela 5.3: Limitante superior $K(\theta)$

é razoável assumir que $\omega^{(u,v)} = 1.200$. Dessa forma, tem-se que o limitante superior $K(\theta) = 14$. Pode-se ver pela Figura 5.10 que a menor quantidade de perdas ocorre quando $K \leq 13$, mais precisamente em $K = 7$.

A Tabela 5.3 foi construída utilizando raciocínio análogo ao apresentado no parágrafo anterior. A análise das Figuras 5.10, 5.15, 5.19, 5.23, 5.27 e 5.31 indica que, à exceção da figura 5.23 e 5.19 a menor quantidade de rejeições se dá quando o valor de K é escolhido dentro do intervalo $[5, K(\theta)]$.

Contudo, é válido salientar que o limitante superior não é justo, dessa forma pode, como foi visto, haver uma diferença entre o limitante superior e o valor de K que oferece o menor número de rejeições.

A Tabela 5.4 apresenta os tempos de execução dos algoritmos quando executados nas redes acima mencionadas. Na tabela 5.4, são apresentados resultados para o algoritmo LMIR para valores de K entre 5 e 10. A partir desse valor, a diferença nos tempos

	50 (D)	50 (E)	40 (D)	40 (E)	30 (D)	30 (E)
MIRA	3m24.973s	2m50.480s	2m25.537s	2m52.195s	2m2.371s	2m54.309s
WSC	3m20.414s	2m52.490s	2m27.357s	2m52.872s	1m59.846s	2m34.611s
LMIR K =5	2m5.703s	2m23.506s	2m8.195s	1m49.877s	2m0.363s	2m19.619s
LMIR K =6	2m23.486s	2m31.346s	2m19.670s	2m13.734s	2m3.732s	2m31.441s
LMIR K =7	2m27.641s	2m43.831s	2m36.943s	2m21.713s	2m9.452s	2m38.125s
LMIR K =8	2m30.270s	2m47.927s	2m39.949s	2m54.207s	2m16.373s	2m52.719s
LMIR K =9	3m4.961s	2m53.270s	3m0.137s	3m3.336s	2m26.437s	3m6.293s
LMIR K =10	3m28.766s	3m2.535s	3m3.193s	3m8.703s	2m37.876s	3m18.678s

Tabela 5.4: Tempo de execução para 30,000 requisições

	50 (D)	50 (E)	40 (D)	40 (E)	30 (D)	30 (E)
MIRA	0.00	1%	1%	0.00	0.00	0.00
WSC	2%	0.0	0.00	0%	2%	20%
LMIR K =5	39%	17%	13%	36%	2%	28%
LMIR K =6	30%	12%	5%	23%	-1%	22%
LMIR K =7	28%	5%	-7%	18%	-6%	19%
LMIR K =8	27%	3%	-9%	-1%	-11%	11%
LMIR K =9	10%	0%	-22%	-6%	-20%	4%
LMIR K =10	2%	-6%	-24%	-6%	-29%	-2%

Tabela 5.5: Ganho relativo no tempo de execução para 30,000 requisições

de execução do algoritmo LMIR em relação ao MIRA e ao WSC torna-se superior a 40 segundos. entretanto, como se pode constatar através das Figuras 5.31, 5.27, 5.23, 5.19, 5.15 e 5.10, com exceção às redes densas com 40 vértices, o valor de K escolhido dentro do intervalo $[5, K(\theta)]$ resulta, no pior caso, na segunda melhor escolha que pode ser feita.

A Tabela 5.5 mostra o ganho relativo de cada algoritmo em relação ao tempo de execução para 30.000 requisições. Em cada rede, foi tomado como base o algoritmo que apresentou o maior tempo de execução. A partir daí, foi calculado o ganho de cada algoritmo.

Pode-se notar que o algoritmo LMIR usando $K = 5$, que foi o valor de K utilizado nos experimentos anteriores, chega a apresentar um ganho de 39% em relação ao algoritmo

	50 (D)	50 (E)	40 (D)	40 (E)	30 (D)	30 (E)
MIRA	6.836×10^{-3}	5.683×10^{-3}	4.851×10^{-3}	5.740×10^{-3}	4.079×10^{-3}	6.477×10^{-3}
WSC	6.680×10^{-3}	5.750×10^{-3}	4.912×10^{-3}	5.762×10^{-3}	3.995×10^{-3}	5.154×10^{-3}
LMIR K =5	4.190×10^{-3}	4.784×10^{-3}	4.273×10^{-3}	3.663×10^{-3}	4.012×10^{-3}	4.654×10^{-3}
LMIR K =6	4.783×10^{-3}	5.045×10^{-3}	4.656×10^{-3}	4.458×10^{-3}	4.124×10^{-3}	5.048×10^{-3}
LMIR K =7	4.921×10^{-3}	5.461×10^{-3}	5.231×10^{-3}	4.724×10^{-3}	4.315×10^{-3}	5.271×10^{-3}
LMIR K =8	5.009×10^{-3}	5.598×10^{-3}	5.332×10^{-3}	5.807×10^{-3}	4.546×10^{-3}	5.757×10^{-3}
LMIR K =9	6.165×10^{-3}	5.776×10^{-3}	6.005×10^{-3}	6.111×10^{-3}	4.881×10^{-3}	6.076×10^{-3}
LMIR K =10	6.959×10^{-3}	6.085×10^{-3}	6.106×10^{-3}	6.290×10^{-3}	5.263×10^{-3}	6.623×10^{-3}

Tabela 5.6: Tempo médio de execução para conexões individuais

	50 (D)	50 (E)	40 (D)	40 (E)	30 (D)	30 (E)
MIRA	0.00	1%	1%	0.00	0.00	0.00
WSC	2%	0.0	0.00	0%	2%	20%
LMIR K =5	39%	17%	13%	36%	2%	28%
LMIR K =6	30%	12%	5%	23%	-1%	22%
LMIR K =7	28%	5%	-7%	18%	-6%	19%
LMIR K =8	27%	3%	-9%	-1%	-11%	11%
LMIR K =9	10%	0%	-22%	-6%	-20%	4%
LMIR K =10	2%	-6%	-24%	-6%	-29%	-2%

Tabela 5.7: Ganho relativo no tempo de execução para conexões individuais

com maior tempo de execução (redes densas de 50 vértices) e 37% em relação ao algoritmo com segundo maior tempo de execução. No pior caso (redes densas de 30 vértices), o LMIR apresenta um tempo de execução 2% inferior ao algoritmo com maior tempo e igual ao algoritmo com segundo maior tempo de execução.

A Tabela 5.6 mostra o tempo médio de execução para cada conexão. Os tempos foram calculados dividindo-se o tempo total de execução para todas as requisições, apresentado na Tabela 5.4 pelo número de requisições.

A Tabela 5.7 mostra o ganho relativo de cada algoritmo em relação ao tempo de execução de cada conexão. Novamente, foi tomado como base o algoritmo que apresentou o maior tempo de execução. A partir daí, foi calculado o ganho de cada algoritmo.

5.4 Considerações finais

Entre os principais desafios da Engenharia de Tráfego na Internet, usando MPLS, está o roteamento adequado dos LSPs de forma a maximizar o número de conexões atendidas. Pela falta de informações sobre demandas futuras, o roteamento deve ser realizado de forma que as conexões sejam atendidas sem detrimento do atendimento das futuras requisições. Assim, o roteamento com interferência mínima, assume papel primordial no processo de Engenharia de Tráfego.

No capítulo 4, um novo algoritmo de roteamento com interferência mínima foi introduzido. Este algoritmo, diferentemente dos algoritmos de roteamento com interferência mínima existentes, não executa o algoritmo de *maxflow* para detecção das arestas críticas. Em substituição ao algoritmo de *maxflow*, modificações foram feitas no algoritmo de *Dijkstra* para que caminhos com menor capacidade fossem encontrados. Após encontradas K arestas críticas (arestas que limitam o fluxo dos menores caminhos), pesos são atribuídos às arestas e um algoritmo para encontrar o menor caminho é executado.

Resultados obtidos através de simulações mostraram que o algoritmo LMIR tem precisão semelhante a dos algoritmos de roteamento com interferência mínima. No entanto, o algoritmo LMIR apresenta menor complexidade computacional do que tais algoritmos.

Capítulo 6

Um mecanismo para dimensionamento dinâmico de LSPs

As redes MPLS têm a vantagem de incorporar circuitos às redes IP. Assim sendo, podem oferecer serviços diferenciados às aplicações, que tipicamente possuem requisitos distintos. Faz-se necessário, portanto, que os LSPs estejam com recursos suficientes alocados de forma que o tráfego gerado pelas aplicações seja satisfatoriamente transmitido.

A alocação adequada dos recursos não é uma tarefa trivial pois dependendo de como os recursos são alocados pode ocorrer uma degradação da utilização da rede ou desperdício de recursos, ocasionando rejeições desnecessárias de estabelecimento de LSPs e prejuízos de ordem financeira aos provedores de serviço de rede.

Neste capítulo, questões relativas à alocação adequada dos recursos serão discutidas bem como a apresentação de um mecanismo que permite a identificação de quando os recursos alocados já não correspondem à demanda de tráfego [FMdFR02].

6.1 Banda Passante Efetiva

Uma das características das redes de comunicação modernas é a multiplexação estatística, ou seja, o compartilhamento de um enlace por várias fontes, aproveitando-se das carac-

terísticas estatísticas do fluxos de tráfego gerados pelas fontes. A questão é determinar qual a quantidade de banda passante que cada fluxo deve receber para que se obtenha o máximo de aproveitamento em relação ao número de fluxos acomodados.

Se a métrica utilizada é a taxa de pico, obviamente haverá uma sub-utilização dado que nem sempre as fontes transmitirão à taxa máxima. Por outro lado, se a métrica utilizada é a taxa média, um número excessivo de fontes pode ser aceito o que ocasionará degradação do serviço. A idéia, então, é utilizar uma medida que represente um compromisso entre os diferentes tipos de fontes, levando em consideração suas propriedades estatísticas e seus requerimentos de QoS. Essa métrica é a Banda Passante Efetiva (*Effective Bandwidth - EBW*).

A banda passante efetiva corresponde, tipicamente, a um valor intermediário entre a taxa média e a taxa de pico e representa a quantidade de banda que um fluxo precisa de forma que seus requisitos de QoS sejam satisfeitos. Para tal, a banda passante efetiva é estimada em função da probabilidade de perda requerida P_l (qualidade de serviço) e as propriedades estocásticas próprias de um fluxo (ou seja, função de autocorrelação ou taxa de pico e taxa média junto com duração média de rajadas). Tendo sido a EBW do fluxo j determinado, $E_j(P_l)$ (ou, $E_j(P(Q > B))$), é necessário garantir que

$$\sum_{j=1}^N E_j(P_l) < C$$

onde N é o número de fluxos multiplexados.

Várias técnicas, incluindo decomposição espectral de fluxo Markoviano [EM93], teoria de grandes desvios [CW95],[KWC93] e a teoria de processos envelope [Cha94] têm sido usadas para derivar a EBW que descreve um fluxo.

Em [CW95], a EBW é definida por

$$E_j(P_l) = \tau_{av,j} + \frac{\delta\gamma_i}{2B}$$

onde

$$\gamma_i = \lim_{t \rightarrow \infty} \frac{1}{t} \text{var}(A_j[0, t])$$

é o índice de dispersão da sequência de chegada e

$$\delta = -\frac{\log(P_l)}{B}.$$

A EBW de um fluxo é independente das propriedades de todos os outros fluxos bem como do número de fontes N e da capacidade da linha C . Ele é determinado a partir das propriedades estocásticas do fluxo, probabilidade de perda requerida P_l e o tamanho do *buffer* B .

Uma definição alternativa para a EBW é dada em [Kel96]

$$E_j(s, t) = \frac{1}{st} \log E[e^{sA_j[0, t]}]$$

de forma que a probabilidade de cauda da distribuição do tamanho da fila satisfaça

$$\lim_{N \rightarrow \infty} \frac{1}{N} \log(\text{Pr}(Q > B)) = \sup_t \inf_s \left[st \sum_j \rho_j E_j(s, t) - s(b + ct) \right]$$

com $C = Nc$, $B = Nb$ e $N\rho_j$ fontes do tipo j .

Essa definição baseia-se na teoria dos grandes desvios e tem sido usada para fornecer resultados gerais do comportamento assintótico do $\log \text{Pr}(Q > B)$ [DO95].

6.2 Redimensionamento dinâmico de LSPs

Como visto na seção anterior, o uso da Banda Passante Efetiva (EBW) fornece ao operador da rede um método eficiente de estimar a capacidade que os circuitos devem possuir. Entretanto, em redes como a Internet, onde a demanda de tráfego é de difícil predição ou mesmo desconhecida, torna-se difícil avaliar a escala de tempo útil de tais circuitos, em outras palavras, qual o intervalo de tempo em que as capacidades dos circuitos devem ser

avaliadas de forma a estarem em conformidade com as demandas de tráfego a que estão sendo submetidos.

Dado o crescente uso do MPLS, um mecanismo dinâmico para dimensionamento dos LSP's que seja adaptativo, sem necessitar de uma requisição explícita de aumento de largura de banda por parte dos usuários faz-se necessário.

As requisições de alteração de largura de banda alocada passa a ser responsabilidade dos roteadores. Assim, o problema de alocação de recursos, torna-se substancialmente diferente do que é, às vezes, adotado em redes baseadas em circuitos onde a demanda de tráfego para determinado circuito é previamente conhecida.

Neste trabalho, assume-se uma rede MPLS com LSPs já dimensionados. Dadas as incertezas do tráfego, nem sempre este dimensionamento prévio é adequado para atender à variação da demanda.

O roteador de ingresso utiliza um mecanismo baseado no contador de picos [MGF91] originalmente proposto para realizar o policiamento de tráfego. Neste trabalho, entretanto, o mesmo método é utilizado para fazer verificação do tamanho médio da fila e redimensionamento dos LSPs.

Na Figura 6.1 é apresentado o mecanismo. Para um tamanho máximo de buffer, e demanda de tráfego, deseja-se idealmente alocar uma largura de banda suficiente ao LSP para que a Qualidade de Serviço (QoS), expressa em termos da taxa de perda de pacotes, desejada seja obtida. Pode-se esperar que a ocupação da fila possua um valor médio $\bar{\varphi}$, excursionando acima e abaixo deste valor médio por períodos de tempo que não comprometem a QoS desejada. A idéia seria então estabelecer limiares superior (τ_s) e inferior (τ_i) e observar por quanto tempo o comprimento da fila permanece, respectivamente, acima e abaixo destes limiares, através de um temporizador. Este processo é ilustrado na Figura 6.1(a). Caso o comprimento da fila permaneça acima do limiar superior por um período de tempo superior a uma tolerância δ_a , este é um sinal de que a largura de banda alocada ao LSP é insuficiente para manter a QoS desejada e, portanto, deve ser ampliada, como pode ser visto na Figura 6.1(b). Analogamente, a permanência do comprimento da fila

abaixo do limiar inferior por um período de tempo superior a uma tolerância δ_d , este é um sinal de que a largura de banda alocada ao LSP é superior à necessária para manter a QoS desejada e, portanto, pode ser reduzida. A utilização destes temporizadores é necessária para evitar que a rede não fique redimensionando os LSP na presença de aumento ou redução circunstancial de tráfego.

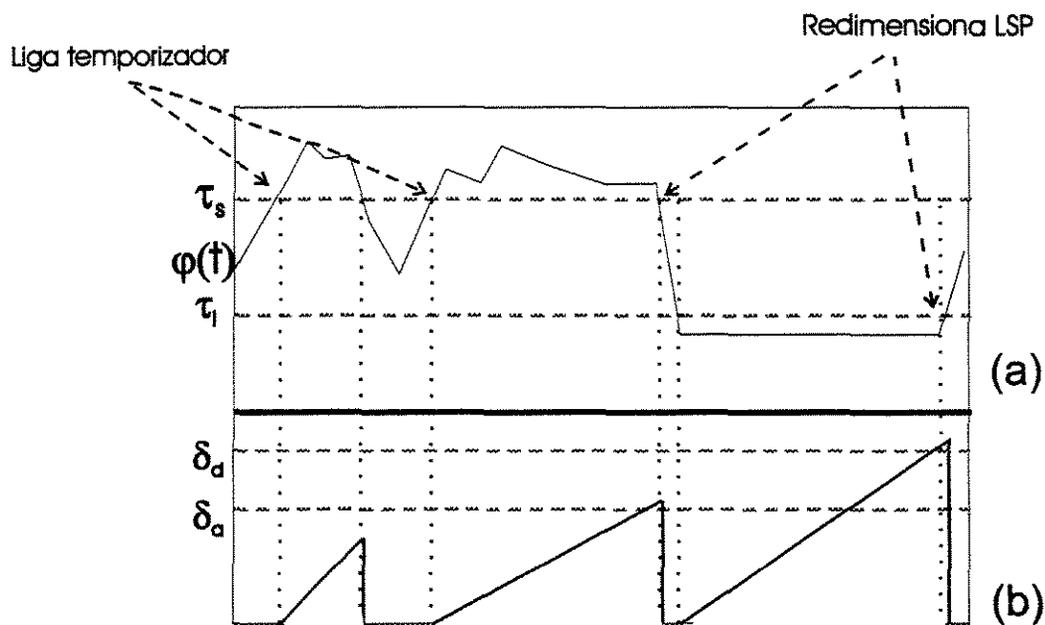


Figura 6.1: Mecanismo de redimensionamento dinâmico de LSPs

Existem então, cinco parâmetros: τ_s e τ_i que são, respectivamente, o limiar superior e o limiar inferior, que sinalizam uma condição de aumento ou diminuição de banda, conforme anteriormente descrito. δ_a e δ_d que indicam se o redimensionamento deve ser feito ou existe apenas um aumento, ou diminuição, circunstancial de tráfego. E por fim φ que é o tamanho médio da fila em um período de tempo t .

6.3 Experimentos de simulação

A topologia utilizada nas simulações é apresentada na Figura 6.2. Foram utilizadas quatro fontes ON-OFF. O período de tempo no estado “ON” segue a distribuição de Pareto, representando assim um tráfego agregado com LRD (*Long-range Dependence*) chegando no roteador de ingresso do LSP com taxa média de 10Mbps. .

A primeira fila da Figura 6.2 representa a fila do LSR de ingresso do LSP que é utilizado para transmitir um fluxo agregado de pacotes pertencentes a uma certa classe de serviço. Este LSR é responsável por fazer o monitoramento do tamanho médio da fila de saída do LSP pertencente à classe de serviço em questão. Além disso, é ele quem inicia o processo de estabelecimento dos LSPs (no início da operação da rede quando não há LSPs estabelecidos), caracterizando estabelecimento de LSPs baseados em roteamento explícito, ou redimensionamento de LSP's, depois de estourado o temporizador que contabiliza o tempo em que o tamanho médio da fila está acima do aceitável.

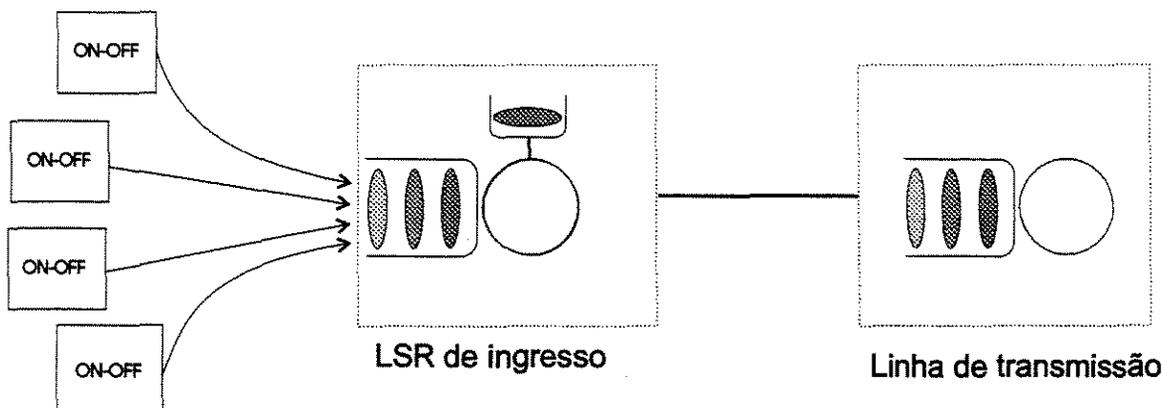


Figura 6.2: Modelo utilizado na simulação

Exceto no caso de super-dimensionamento (descrito adiante), o LSP opera a uma taxa um pouco menor do que taxa média de chegadas. Assim, espera-se um tráfego intenso no LSP e o mecanismo de realocação de banda dos LSPs é avaliado sob estas suposições. O segundo servidor da Figura 6.2 representa a transmissão na linha física e

é executado a uma taxa de 20 Mbps. Se o temporizador estourar, uma quantidade de banda é adicionada ao LSP, de forma que, tendo uma taxa de serviço maior, ele consegue diminuir o comprimento médio da fila e assim, a quantidade de perdas.

Os pacotes têm tamanho fixo de 500 bytes, o que é uma premissa razoável já que são pacotes de aplicações pertencentes à mesma classe de serviço, e conseqüentemente, requisitos bem parecidos. O *buffer* tem uma capacidade de 100KB. O número de pacotes no *buffer* é facilmente calculado com a expressão $\left\lfloor \frac{\text{Tamanho do buffer}}{\text{Tamanho do pacote}} \right\rfloor$.

Para a avaliação de desempenho do mecanismo em questão, um modelo de simulação foi desenvolvido utilizando a ferramenta TANGRAM-II[tan]. Os LSPs foram modelados, na ferramenta, como um *Leaky Bucket* com uma taxa de chegadas de fichas representando a taxa do LSP como mostra a Figura 6.2. Assim, os pacotes são transmitidos somente quando há ficha no balde.

Seis casos foram considerados: no primeiro, foi utilizado um LSP mal dimensionado em relação ao tráfego de entrada, que possui uma taxa de 6 Mbps. Obviamente este é a caracterização de um modelo instável, o que significa a fila estar sempre cheia e as perdas crescem infinitamente como indica a Figura 6.3. Este caso representa uma alocação inadequada.

Um segundo cenário corresponde a um “super-dimensionamento” cuja técnica é empregada por muitas operadoras de rede. Para tal situação foi utilizado um LSP de 12 Mbps. Como vê-se na Figura 6.3, o número de perdas é zero, porém, paga-se um preço alto pois quase 40% de banda é desperdiçada como pode ser visto na Figura 6.6.

Os outros quatro casos contemplam a utilização do mecanismo de redimensionamento dinâmico de banda dos LSPs. Nas simulações são utilizados quatro valores diferentes de τ_s e τ_i que são sumarizados na Tabela 6.1. Nos casos com realocação de banda, assume-se de modo análogo ao primeiro caso simulado, um dimensionamento inadequado de banda do LSP cuja taxa é igual a 6 Mbps. Entretanto, como descrito na Seção 6.2, ao perceber uma utilização acima de τ_s ou abaixo de τ_i os LSPs são redimensionados.

Na Figura 6.3, a quantidade de perdas experimentada pelo LSP é mostrada. Nota-se

Casos	τ_s	τ_i
3	80%	20%
4	70%	30%
5	60%	40%
6	40%	20%

Tabela 6.1: Valores dos limiares

que quanto menor for o limiar, mais cedo o dimensionamento adequado é alcançado, as perdas são eliminadas e o tempo que a rede leva pra sair de uma situação desfavorável e passar a um estado em que possa oferecer garantias em relação às perdas é menor.

A Figura 6.4 mostra a taxa de perdas experimentada pelo LSP. O intervalo em que existe um crescimento da taxa de perdas corresponde ao período em que o limiar não é atingido. Como neste período nenhuma realocação de banda foi realizada e o LSP não consegue atender a demanda, grande parte dos pacotes é perdida, o que provoca o aumento da taxa de perda. A diminuição acontece depois do redimensionamento do LSP. Percebe-se que o decréscimo mais acentuado da taxa de perda se dá quando o limiar corresponde a 40% do tamanho do *buffer* e o menos acentuado se dá quando o limiar corresponde a 80%. Isto também é explicado pelo fato de o limiar ser atingido antes dos outros casos, eliminando as perdas. Além disso, a Figura 6.5 mostra que a utilização de limiares mais altos faz com que o tamanho da fila diminua moderadamente, fazendo com que a fila esteja em utilização máxima por mais tempo como indicado na Figura 6.6.

6.4 Considerações Finais

Dimensionamento dinâmico dos LSPs pode ser uma ferramenta poderosa para o oferecimento de QoS em redes onde não se sabe *a priori* qual é demanda de tráfego.

Resultados, obtidos via simulação, indicaram que quanto menor for o limiar superior mais fácil se dá a recuperação de um cenário desfavorável. Entretanto, para que a eficácia do mecanismo seja melhor avaliada, sugere-se o uso de modelos mais realistas de tráfego,

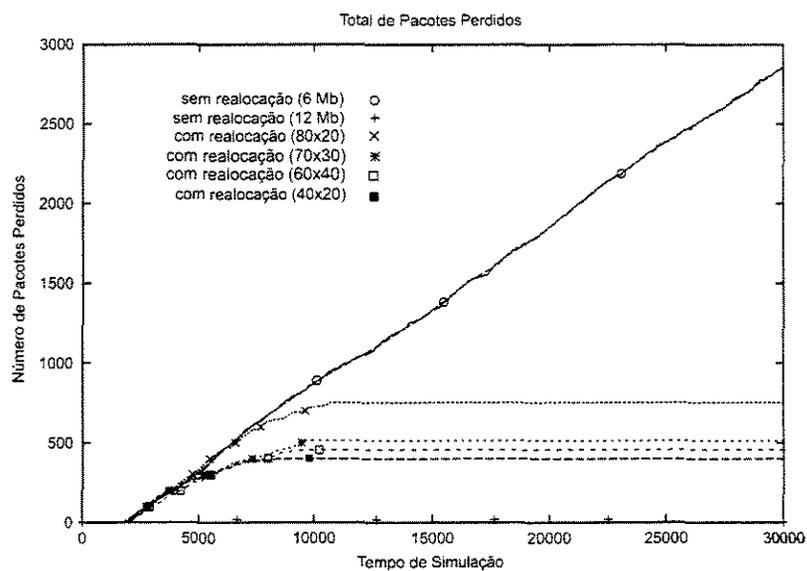


Figura 6.3: Quantidade total de perdas

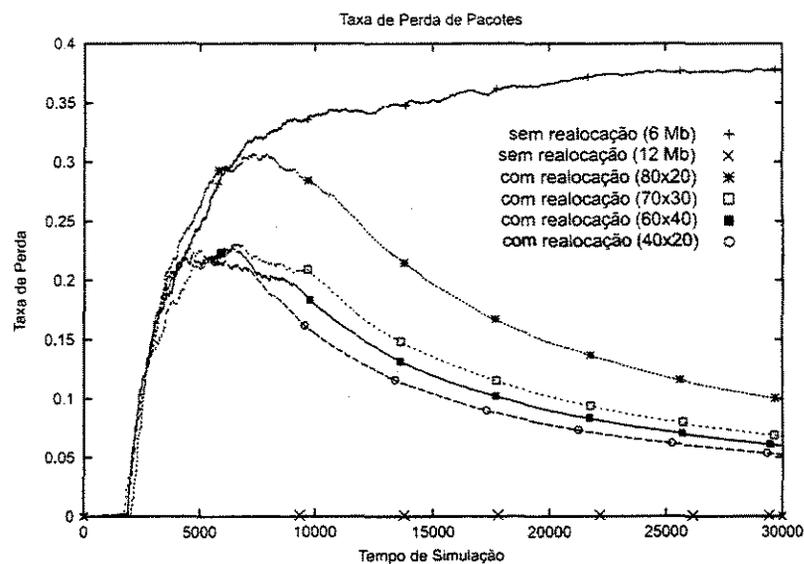


Figura 6.4: Taxa de perdas experimentada pelo LSP

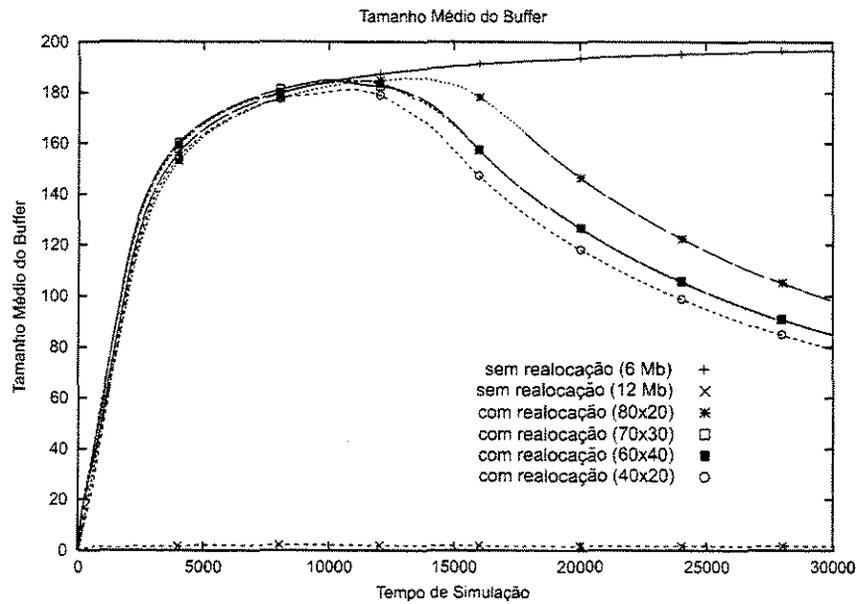


Figura 6.5: Comprimento médio do buffer

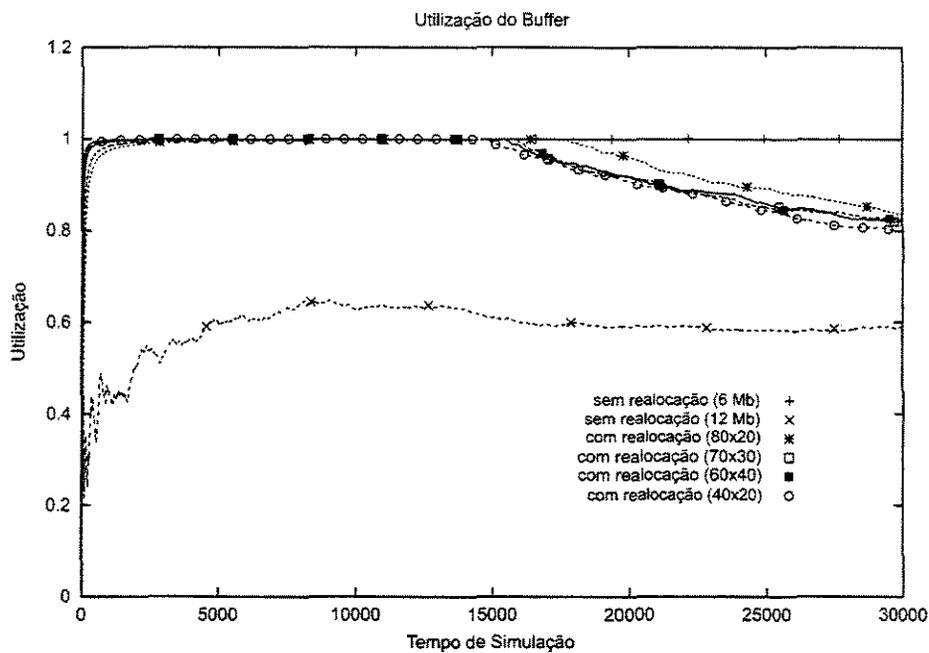


Figura 6.6: Utilização média

custos de sinalização e *overhead* de processamento nas simulações.

Capítulo 7

Conclusões

O oferecimento de novos serviços, que sejam confiáveis e garantam Qualidade de Serviço na Internet passa necessariamente pela adoção de técnicas de Engenharia de Tráfego. A não utilização de tais técnicas pode levar a uma má distribuição de tráfego e conseqüente degradação dos serviços, mesmo quando há uma alta disponibilidade de recursos.

No contexto de Engenharia de Tráfego de redes IP, o MPLS tem ganho bastante destaque devido às suas facilidades de aliar os benefícios das redes orientadas a conexão às redes não orientadas a conexão, através do uso de caminhos comutados por rótulos. Nesta dissertação, foram propostos um algoritmo de roteamento com interferência mínima e um mecanismo de dimensionamento dinâmico de LSPs.

O algoritmo de roteamento com interferência mínima, denominado LMIR, utiliza uma variação do algoritmo de Dijkstra, para realizar a detecção das arestas críticas. A sua eficácia foi analisada através de simulações comparando o LMIR a outros algoritmos de interferência mínima encontrados na literatura. Os resultados indicaram que o LMIR possui precisão similar aos outros algoritmos de roteamento com interferência mínima, porém com tempo de execução inferior.

O dimensionamento dinâmico dos LSPs pode ser uma ferramenta poderosa para o oferecimento de QoS em redes onde não se sabe *a priori* qual é demanda de tráfego. Resultados, obtidos via simulação, indicaram que quanto menor for o limiar superior

mais fácil se dá a recuperação de um cenário desfavorável.

7.1 Trabalhos Futuros

Propõe-se como trabalhos futuros a utilização de outras funções de peso como forma de penalização das arestas críticas. Além disso, outros esquemas de identificação de arestas críticas podem ser propostos e avaliados.

É sabido que a natureza do tráfego da Internet pode ser modelado como fractal se a escala de tempo for maior que um tempo de ida e volta dos pacotes (*Round Trip Time* -RTT). Dessa forma, as escalas de tempo de interesse fornecidas pelo processo envelope do Movimento Browniano Fractal [Mdf03] podem ser utilizadas como valores utilizados como limiares superior e inferior do mecanismo de dimensionamento dinâmico.

Referências Bibliográficas

- [AJ02] Daniel O. Awduche e Bijan Jabbari. Internet Traffic Engineering Using Multi-Protocol Label Switching (MPLS). *Computer Networks*, 40(1):111–129, 2002.
- [Al98] S. Slake Et Al. An Architecture of Differentiated Services. Em *RFC 2475*, Dezembro 1998.
- [Al02a] Daniel O. Awduche Et Al. Overview and Principles of Internet Traffic Engineering. Em *RFC 3272*, Maio 2002.
- [Al02b] Tricha Anjali Et Al. Optimal Policy for Label Switched Path Setup in MPLS Networks. *Computer Networks*, 39(2):165–183, 2002.
- [Awd99] Daniel O. Awduche. MPLS and Traffic Engineering in IP Networks. *IEEE communications Magazine*, 37(12):42–47, Dezembro 1999.
- [BCS94] R. Braden, D. Clark, e S. Shenker. Integrated Services in the Internet Architecture: an Overview. Em *RFC 1633*, Junho 1994.
- [Cav01] Mábia Daniel Cavalcante. Algoritmos de Balanceamento de Carga para Tráfego Tipo Melhor Esforço em Redes IP/MPLS. Dissertação de Mestrado, FEEC/UNICAMP, 2001.

- [Cha94] C. Chang. Stability, Queue Length and Delay of Deterministic and Stochastic Queuing Networks. *IEEE Transactions on Automatic Control*, 39(5):913–931, Maio 1994.
- [Che99] Shigang Chen. *Routing support for providing guaranteed end-to-end quality-of-service*. Tese de Doutorado, Engineering college – University of Illinois at Urbana-Champaign, 1999.
- [CLR90] Thomas H. Cormen, Charles E. Leiserson, e Ronald L. Rivest. *Introduction to Algorithms*. MIT Press and McGraw Hill, 1990.
- [CN98] Shigang Chen e Klara Nahrstedt. QoS Routing for Next-Generation Networks. *IEEE Network*, Dezembro 1998.
- [CW95] C. Courcoubetis e R. Weber. Effective Bandwidth for Stationary Sources. *Probability in Eng. and Info. Sci.*, 9(2):285–294, 1995.
- [DO95] N. Duffield e N. O’Connell. Large deviations and overflow probabilities for the general single server queue, with application. *Cambridge Phil. Soc.*, 118:363–374, 1995.
- [DR00] Bruce Davis e Yakov Rekhter. *MPLS: Technology and Applications*. Morgan Kaufmann Publishers, 2000.
- [eDS02] Gargi Banerjee e Deepinder Sidhu. Comparative Analysis of Path Computation Techniques for MPLS Traffic Engineering. *Computer Networks*, 40(1):149–165, 2002.
- [EM93] A. I. Elwalid e D. Mitra. Effective Bandwidth of General Markovian Traffic Sources and Admission Control of High Speed Networks. *Transactions on Networking*, 1(4):329–343, Junho 1993.
- [FdFM03] Gustavo B. Figueiredo, Nelson L. S. da Fonseca, e José A. S. Monteiro. Um Algoritmo para Roteamento com Interferência Mínima. Em *Anais do XXI*

- Simpósio Brasileiro de Redes de Computadores*, volume 2, páginas 507–518, 2003.
- [FMdFR02] Gustavo B. Figueiredo, José A. S. Monteiro, Nelson L. S. da Fonseca, e Antônio A. A. Rocha. Dynamic Sizing of Label Switched Paths in MPLS Networks. Em *Anais da conferência IEEE International Telecommunications Symposium*, páginas 593–598, 2002.
- [GJ79] M. Garey e D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Co., 1979.
- [GR98] Andrew V. Goldberg e Satish Rao. Beyond the flow decomposition barrier. Em *J. ACM* 45 (5), páginas 783–797, 1998.
- [Kel96] F. Kelly. *Notes on Effective Bandwidths*. Oxford University Press, 1996.
- [KL00] Murali S. Kodialam e T. V. Lakshman. Minimum interference routing with applications to MPLS traffic engineering. Em *INFOCOM (2)*, páginas 884–893, 2000.
- [KWC93] G. Kesidis, J. Walrand, e C. Chang. Effective Bandwidth for Multiclass Markov Fluids and Other ATM Sources. *IEEE/ACM Transactions on Networking*, 9(1):424–428, Agosto 1993.
- [MC01] Maurício F. Magalhães e Eleri Cardozo. Introdução à Comutação IP por Rótulos Através de MPLS. Livro de Minicurso do XIX Simpósio Brasileiro de Redes de Computadores, 2001.
- [MdF03] Cesar A. V. Melo e Nelson L. S. da Fonseca. An envelope process for multifractal traffic modeling. Relatório Técnico IC-03-10, IC - UNICAMP, Abril 2003.

- [MGF91] J. A. S. Monteiro, M. Gerla, e L. Fratta. Input Rate Control for ATM Networks. Em *ITC13 Queuing, Performance and Control in ATM*, páginas 117–122, 1991.
- [OSAU02] J. C. de Oliveira, C. Scoglio, I. F. Akyildiz, e G. Uhl. A New Preemption Policy for Diffserv-Aware Traffic Engineering to Minimize Rerouting. Em *INFOCOM*, Junho 2002.
- [Rab00] Peter Rabinovitch. Statistical estimation of effective bandwidth. Dissertação de Mestrado, School of Mathematics and Statistics at the Carleton University, 2000.
- [RVC01] Eric Rosen, Arun Viswanathan, e Ross Callon. MultiProtocol Label Switching Architecture. Em *RFC 3031*, Janeiro 2001.
- [SLC95] L. F. G. Soares, G. Lemos, e S. Colcher. *Das Lans, Mans e Wans às redes ATM*. Editora Campus, 1995.
- [tan] *Tangram II User's Manual*. www.land.ufrj.br/tools/tangram2/tangram2.html.
- [Tan97] Andrew S. Tanenbaum. *Redes de Computadores*. Editora Campus, 1997.
- [Wax88] Bernard M. Waxman. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 6(9):1617–1622, 1988.
- [Wes96] Douglas B. West. *Introduction to Graph Theory*. Prentice Hall, 1996.
- [WSC02] Bin Wang, Xu Su, e Philip Chen. A New Bandwidth Guaranteed Routing Algorithm for MPLS Traffic Engineering. Em *ICC 2002 - IEEE International Conference on Communications*, volume 25, páginas 1001–1005, Abril 2002.
- [ZCB96] Ellen W. Zegura, Kenneth L. Calvert, e Samrat Bhattacharjee. How to model an internet network. Em *IEEE Infocom*, volume 2, páginas 594–602, San Francisco, CA, Março 1996.