

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA
ELÉTRICA**

**Engenharia de Tráfego em Redes IP sobre Tecnologia
MPLS: Otimização Baseada em Heurísticas.**

Tese submetida à
Universidade Federal de Santa Catarina
como parte dos requisitos para a
obtenção do grau de Doutor em Engenharia Elétrica.

ROBERTO ALEXANDRE DIAS

Florianópolis, Abril de 2004.

Engenharia de Tráfego em Redes IP sobre Tecnologia MPLS: Otimização Baseada em Heurísticas.

Roberto Alexandre Dias

'Esta tese foi julgada adequada para a obtenção do título de Doutor em Engenharia Elétrica, Área de Concentração em *Sistemas Computacionais*, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina.'

Jean-Marie Farines, Dr.
Orientador

Jefferson Brum Marques
Coordenador do Programa de Pós-Graduação em Engenharia Elétrica

Banca Examinadora:

Jean-Marie Farines, Dr. Presidente

Roberto Willrich, Dr. Co-orientador - UFSC

Maurício Magalhães, Dr. UNICAMP

Eduardo Camponogara, PhD. UFSC

Joberto S. B. Martins, Dr. UNIFACS

Joni S. Fraga. Dr. UFSC

A trilogia anterior foi feita ... Uma árvore ... Um filho ... e um livro ... Agora a história continua ... Mais árvores ... mais livros ...

À minha esposa e filha...

Márcia e Nati

Agradecimentos

Agradeço inicialmente a Deus por ter me dado o dom de estudar ... Pois nem sempre estudar é uma questão de gosto ou inteligência, mas sim de perseverança.

Agradeço a minha esposa Márcia e minha filha Natália pela compreensão na subtração de tempo que lhes impingi. "O mãe eu não acredito que você foi incomodar o pai bem na hora do estudo dele ..."

*Agradeço a acolhida do meu orientador Jean-Marie num momento tormentoso de minha vida acadêmica. Com ele aprendi o significado da palavra **pesquisa**.*

Agradeço ao amigo Roberto Willrich que me recebeu no CPGCC e me encaminhou para o CPGEEL.

Agradeço ao amigo Eduardo Camponogara pelos ensinamentos, paciência e orientação. Sempre esteve disposto a nos ajudar nesta empreitada.

Agradeço ao amigo Adriano Campestrini, programador exímio, um jovem de grande futuro.

Resumo da Tese apresentada à UFSC como parte dos requisitos necessários para a obtenção do grau de Doutor em Engenharia Elétrica.

Engenharia de Tráfego em Redes IP sobre Tecnologia MPLS: otimização baseada em heurísticas.

Roberto Alexandre Dias

Abril de 2004

Orientador: Jean-Marie Farines, Dr.

Área de Concentração: Sistemas Computacionais

Palavras-chave: **Otimização, Engenharia de Tráfego, Qualidade de Serviços, MPLS.**

Número de Páginas: 100

RESUMO.

Esta tese propõe uma nova abordagem de solução de um problema de Engenharia de Tráfego dinâmico em redes IP sobre Tecnologia Multiprotocol Label Switching (MPLS).

O objetivo do problema de Engenharia de Tráfego (PET), proposto neste trabalho, visa a maximização da vazão global dos fluxos de dados injetados em uma topologia de rede de tamanho significativo. Para tanto são configurados caminhos de menor distância métrica entre origem e destino dos fluxos de dados, bem como das larguras de bandas dos Label Switched Paths (LSPs), nos quais os fluxos de dados serão encaminhados. Estas configurações devem respeitar as restrições de largura de banda dos enlaces e atraso fim-a-fim máximo, de cada fluxo de dados. Do ponto de vista da teoria de otimização, este problema é enquadrado como um *problema de caminhos mínimos sujeito a restrições*, sendo do tipo NP-completo.

Neste problema foi adotada uma política de diferenciação de serviços que depende de um controle de admissão e da configuração de largura de banda dos LSPs, baseadas em parâmetros de prioridade para cada fluxo de dados. Foram analisados cenários onde os fluxos foram classificados como sendo de *alta prioridade* e de *baixa prioridade*. Nesta política, na eminência de congestionamento da rede, os fluxos de baixa prioridade podem ter suas vazões reduzidas até zero (fluxos rejeitados) para garantir os requisitos de QoS dos fluxos de alta prioridade.

O problema foi modelado usando notação de programação linear inteira, sendo resolvido através de heurísticas simples, que consomem baixo tempo de execução e produzem soluções de qualidade satisfatória, quando comparadas às soluções exatas computadas por pacotes comerciais de programação matemática.

Abstract of Thesis presented to UFSC as a partial fulfillment of the requirements for the degree of Doctor in Electrical Engineering.

Traffic Engineering in IP Networks over MPLS: Heuristics Based Otimization

Roberto Alexandre Dias

April 2004

Advisor: Jean-Marie Farines, Dr.

Area of Concentration: Computational Systems

Keywords: **Otimization, Traffic Engineering, Quality of Service, MPLS.**

Number of Pages: 100

ABSTRACT.

This thesis proposes a new solution approach for a Traffic Engineering (TE) problem in IP based Networks over Multiprotocol Label Switching (MPLS) technology.

The goal of this work is to maximize overall throughput of data flows injected in a representative network topology, with expressive workload. To meet this objective, paths and bandwidths of Label Switched Paths (LSPs) are configured to forward the data flows. The configurations are subject to constraints such as: network link bandwidths and data-flow end-to-end delay. This is a multiple constraint shortest-path problem — an NP-Complete type problem.

In this problem a service differentiation policy is adopted, based in priority parameters. The data flows are classified in two types: high and low priority data flows.

In the eminence of network congestion, the bandwidth of low priority data flows can be reduced to zero.

The problem was modeled using integer linear programming and solved by simple heuristics, with fast and high quality solution.

Sumário

1	Introdução	1
1.1	Objetivos	2
1.2	Resultados esperados	3
1.2.1	Plano de trabalho	3
1.2.2	Organização da monografia de tese	3
2	Engenharia de tráfego e MPLS	4
2.1	Engenharia de tráfego em redes IP	4
2.2	Taxonomia de Sistema de Engenharia de Tráfego	6
2.3	Tecnologia MPLS: conceitos básicos e terminologia	7
2.3.1	Alguns conceitos e componentes da tecnologia MPLS	9
2.4	Roteamento baseado em restrições	11
2.4.1	RSVP-TE	12
2.4.2	Protocolo CR-LDP	13
2.5	MPLS e Engenharia de Tráfego	13
2.6	Conclusões do capítulo	14
3	Engenharia de tráfego e otimização	15
3.1	Modelagem de problemas	15
3.1.1	Modelos de programação matemática	16
3.1.2	Problemas NP-Completo	18
3.2	Programação Linear Inteira (PLI)	19
3.2.1	Problema geral	19
3.2.2	Problema da mochila	20
3.2.3	Métodos de solução exatos para problemas de programação linear inteira	21
3.2.3.1	Branch and Bound	21
3.2.3.2	Programação dinâmica (PD)	22
3.3	Métodos de soluções aproximados dos problemas de PLI	23
3.3.1	Emprego de relaxações	23
3.3.1.1	Relaxação Lagrangeana	24

3.3.2	Soluções Heurísticas	26
3.4	Engenharia de tráfego como um problema de otimização	26
3.4.1	O problema de caminhos mínimos (PCM)	27
3.4.2	Algoritmo de Dijkstra	28
3.4.3	O problema de caminhos mínimos sujeito a restrições	29
3.5	Conclusões do capítulo	30
4	Modelagem de um Problema de Engenharia de Tráfego (PET)	31
4.1	Problema de Engenharia de Tráfego	31
4.2	Modelagem e formulação do problema	32
4.3	Solução Lagrangeana	33
4.3.1	Algoritmo subgradiente	36
4.3.2	Heurística de recuperação de viabilidade primal (HRV)	38
4.4	Solução heurística pura	40
4.5	Experimentos e resultados	41
4.5.1	Experimentos	42
4.5.2	Topologia 1	43
4.5.2.1	Avaliação de desempenho do PHP e RLH	44
4.5.2.2	Controle de admissão do <i>PHP</i>	45
4.5.3	Topologia 2	47
4.6	Conclusões do capítulo	49
5	Uma abordagem para solução de um problema de engenharia de tráfego dinâmico (PETDIN)	51
5.1	Definição do problema	51
5.2	Proposta de solução	53
5.3	Implementação da solução	53
5.3.1	Estrutura lógica	54
5.3.2	Infra-estrutura física	54
5.3.3	Valores de prioridade de configuração e manutenção	56
5.3.4	Discussão sobre questões operacionais da proposta	57
5.4	Estratégia de avaliação da proposta	57
5.5	Experimentos e resultados	59
5.5.1	Limitações da plataforma de validação	59
5.5.2	Desempenho do algoritmo	60
5.5.3	Controle de admissão	62
5.5.3.1	Alterando a relação <i>holding/setup</i>	65
5.5.4	Garantia de QoS dos Fluxos de Alta prioridade	65
5.5.5	Conclusões sobre o controle de admissão do PHP	71

5.6	Trabalhos relacionados	72
5.7	Conclusões do capítulo	73
6	Conclusão Final e Perspectivas de Trabalhos Futuros	75
6.1	Contribuições	76
6.2	Perspectivas de Trabalhos Futuros	77
6.3	Comentários finais	78
A	Gerenciamento baseado em políticas	79
A.1	Conceitos básicos	79
A.2	Gerenciamento baseado em políticas do IETF	81
A.3	Relacionamento com a abordagem proposta	83

Lista de Figuras

2.1	O problema da espinha de peixe	5
2.2	Exemplo de roteamento convencional	8
2.3	Encaminhamento de pacotes no MPLS	9
2.4	Formato do shim header	10
4.1	Primeira topologia de rede	42
4.2	Segunda topologia de rede	43
4.3	Tempo de execução do RLH, PHP e Xpress-MP para várias cargas de trabalho	44
4.4	Grau de otimalidade do <i>RLH</i> e <i>PHP</i>	45
4.5	Percentual de fluxos rejeitados na rede pelo emprego do <i>PHP</i>	46
4.6	Taxa de perdas de pacotes global quando empregados os protocolos RIP e OSPF	47
4.7	Tempo computacional consumido pelo <i>RLH</i> e <i>PHP</i> para a topologia 2	48
4.8	Valores da função objetivo computados pelo <i>RLH</i> e <i>PHP</i>	48
5.1	Linha do tempo de um problema de ET dinâmico	52
5.2	Estrutura lógica da solução do PETDIN	55
5.3	Infra-estrutura para o procedimento dinâmico de engenharia de tráfego PETDIN	55
5.4	Valores de função objetivo ao longo do tempo para as várias cargas de trabalho experimentadas	61
5.5	Tempo de execução ao longo do tempo de simulação para a carga de trabalho de 754 fluxos de dados, em média, por rodada de otimização	62
5.6	Vazão do fluxo 299 ao longo do tempo, pelo uso do <i>PHP</i> comparado ao uso do RIP e OSPF	67
5.7	Vazão do fluxo 311 ao longo do tempo, pelo uso do <i>PHP</i> comparado ao uso do RIP e OSPF	67
5.8	Vazão do fluxos 234 ao longo do tempo, pelo uso do <i>PHP</i> comparada com RIP e OSPF	68
5.9	Atraso fim-a-fim do fluxo 299 ao longo do tempo, pelo uso do <i>PHP</i> comparado com RIP e OSPF	68
5.10	Atraso fim-a-fim do fluxo 311 ao longo do tempo, pelo uso do <i>PHP</i> comparado com RIP e OSPF	69

5.11	Atraso fim-a-fim do fluxo 234 ao longo do tempo, pelo uso do <i>PHP</i> comparado com RIP e OSPF	69
A.1	Arquitetura para gerenciamento de rede baseado em políticas do IETF	82

Capítulo 1

Introdução

A Internet tem evoluído de forma exponencial e as demandas das aplicações, em termos de desempenho e segurança, têm se tornado mais sofisticadas, exigindo assim, maior qualidade dos serviços oferecidos pelas operadoras de telecomunicações. Desta forma, se torna imperativo a adoção de novas políticas operacionais que suportem, por exemplo, estratégias de diferenciação de serviços, adicionando flexibilidade e eficiência à infra-estrutura de comunicação. O gerenciamento da Qualidade de Serviços (QoS) desponta neste cenário como uma alternativa vital ao sobre-provisionamento de recursos. Neste contexto, a Engenharia de Tráfego (ET) representa uma forma de aproveitar melhor os recursos de rede existentes, com consequente redução de custos (Awduche et al. 1999).

A *Engenharia de tráfego* é uma técnica eficiente para o adequado gerenciamento do tráfego nas redes, com o objetivo de otimizar o uso dos seus recursos. Entre os principais objetivos da engenharia de tráfego podemos destacar: redução nos congestionamentos; uso eficiente dos recursos de rede, com consequente economia de recursos financeiros; satisfação dos requisitos das aplicações e dos usuários e, finalmente, a melhoria geral de desempenho da rede.

Um elemento chave de suporte à ET em redes IP é a tecnologia **Multiprotocol Label Switching (MPLS)**, principalmente pelo fato de que esta tecnologia permite a implementação de orientação à conexão, com a possibilidade do emprego da técnica de roteamento explícito. Nesta técnica as rotas a serem seguidas pelos pacotes podem ser definidas *a priori* na origem dos fluxos de dados.

Para tanto, é necessário que o MPLS utilize protocolos de sinalização como o RSVP-TE (Awduche 2001) e o CR-LDP (Jamoussi 2002), que surgem de forma promissora para o suporte de roteamento com Qualidade de Serviços (QoS) (Wang 1999).

O presente trabalho busca uma solução eficiente que melhore a utilização dos recursos da rede, com um impacto mínimo na arquitetura da infra-estrutura existente. Muitos trabalhos relacionados buscam soluções para problemas de ET, porém necessitam do desenvolvimento de novos protocolos de rede, da atualização ou adaptação de tecnologias existentes. Além disso, a abordagem proposta neste trabalho, além de promover uma melhor utilização de recursos da rede, imbuir em seu modelamento um controle de admissão que permite a implementação de uma estratégia de diferenciação de serviços que pode atuar de forma conjunta ou complementar a outras arquiteturas de provimento de QoS, como os Serviços Diferenciados.

De forma particular, as ferramentas de ET permitem a seleção de caminhos de rede a serem configurados em caminhos virtuais da tecnologia MPLS (similares a circuitos virtuais do ATM) denominados Label Switched Paths (LSPs). A configuração destes LSPs no processo de ET deve estar sujeita às restrições de QoS, expressas em termos de métricas como: vazão mínima, atraso fim-a-fim máximo, variação de atraso máximo, taxa de perdas máxima, de cada fluxo de dados a ser admitido na rede.

1.1 Objetivos

O objetivo principal deste trabalho é o desenvolvimento de algoritmos que permitam a implementação de ET em redes IP sobre tecnologia MPLS, empregando algoritmos de roteamento heurísticos.

Os objetivos específicos são relacionados a seguir:

- Proposta de uma abordagem de implementação de engenharia de tráfego dinâmica;
- Avaliação do desempenho computacional de algoritmos e qualidade das soluções quanto à melhoria de utilização de recursos da rede;
- Avaliação do impacto da utilização das ferramentas de ET desenvolvidas nos parâmetros de desempenho como: vazão, atraso, variação de atraso e taxa de perda de pacotes, em comparação com protocolos de roteamento convencional.
- Modelamento matemático dos problemas de ET usando notação de programação matemática inteira;
- Solução dos problemas de ET utilizando relaxações e algoritmos heurísticos;

1.2 Resultados esperados

- Desenvolvimento de modelos matemáticos que representem o problema de Engenharia de Tráfego;
- Obtenção de soluções eficientes baseadas em heurísticas simples;
- Melhoria da QoS das aplicações em comparação com roteamento IP convencional.

1.2.1 Plano de trabalho

O trabalho de tese foi organizado nas seguintes etapas:

- Levantamento bibliográfico sobre engenharia de tráfego, tecnologia Multiprotocol Label Switching e teoria de otimização;
- Estudo de ferramentas e bibliotecas matemáticas de programação linear inteira;
- Modelamento matemático de problemas de ET;
- Desenvolvimento de algoritmos de relaxação e heurísticos para solução dos problemas de ET;
- codificação dos algoritmos em linguagem C;
- avaliação dos modelos e soluções pelo emprego de experimentação numérica e simulações.

1.2.2 Organização da monografia de tese

O presente trabalho tem a seguinte organização: (i) o capítulo 2 apresenta uma revisão bibliográfica sobre Engenharia de Tráfego e tecnologia Multiprotocol Label Switching; (ii) o capítulo 3 apresenta uma revisão bibliográfica sobre teoria de otimização de sistemas, com ênfase na solução de problemas de engenharia de tráfego; (iii) o capítulo 4 apresenta o modelamento e solução de um problema de engenharia de tráfego; (iv) o capítulo 5 apresenta uma abordagem de solução de problemas de engenharia de tráfego em tempo real e seu posicionamento em relação aos trabalhos relacionados; (v) o capítulo 6 apresenta as conclusões finais e propostas de trabalhos futuros.

Capítulo 2

Engenharia de tráfego e MPLS

Este capítulo apresentará uma revisão bibliográfica sobre os principais conceitos de Engenharia de Tráfego (ET). Serão enfatizadas as abordagens de implementação de ET em redes IP sobre tecnologia Multiprotocol Label Switching (MPLS).

Serão discutidos os seguintes assuntos: (i) Engenharia de Tráfego em redes IP: conceitos e classificação; (ii) Tecnologia MPLS: terminologia e conceitos básicos; (iii) Abordagens de implementação de Engenharia de Tráfego em Redes IP.

2.1 Engenharia de tráfego em redes IP

Os algoritmos de roteamento atualmente em uso em redes como a Internet são baseados na escolha de rotas que minimizem métricas, como número de saltos, entre origem e destino de um fluxo de dados. Do ponto de vista de QoS, nem sempre o caminho mais curto é o caminho que apresenta o melhor conjunto de recursos necessários a uma aplicação.

Na maioria das redes atuais, quando enlaces começam a apresentar tendência de congestionamento, a alternativa mais comum para solução deste problema é o aumento da capacidade dos mesmos. Com o crescimento das redes e o aumento da demanda de recursos, por parte das aplicações, surge a necessidade de uma abordagem de provisionamento da rede de forma mais eficiente. Neste sentido, o emprego de técnicas de engenharia de tráfego, vem a contribuir significativamente com a evolução das redes de computadores.

Segundo (Magalhães e Cardoso 2001) *”engenharia de tráfego (ET) é a tarefa de realizar o mapeamento dos fluxos de tráfego em uma infra-estrutura física de transporte, de modo a atender os critérios definidos pela operação da rede”*.

Como resultado direto da ação da engenharia de tráfego é possível estabelecer-se um balanceamento de carga nos enlaces de rede, de modo a reduzir-se os congestionamentos

e otimizar a utilização dos recursos de rede. Dentre os benefícios do emprego de técnicas de engenharia de tráfego, podemos citar:

- Redução de pontos de congestionamento, que representam gargalos na rede;
- Re-roteamento rápido de fluxos de dados em caso de falhas;
- Redução de custos pelo melhor aproveitamento dos enlaces, com uso mais eficiente da banda disponível;
- Melhoria geral na qualidade de serviço, pela redução da taxa de perdas de pacotes e redução da variação de atraso.

A figura 2.1 (problema da espinha de peixe) mostra um cenário típico onde o emprego de protocolos de roteamento IGP (Internet Gateway Protocol) como o OSPF e IS-IS pode não ser eficiente. Nestes protocolos, o roteamento se dá pela minimização do número de saltos (hops), o que pode provocar um mau aproveitamento dos recursos de rede, uma vez que a rota a ser seguida será sempre a de menor número de saltos, mesmo que caminhos alternativos estejam ociosos. A adoção de um caminho alternativo, como apontado na figura, empregando alguma técnica de engenharia de tráfego, salienta as potencialidades de melhoria no aproveitamento dos recursos de rede, proporcionadas pela ET.

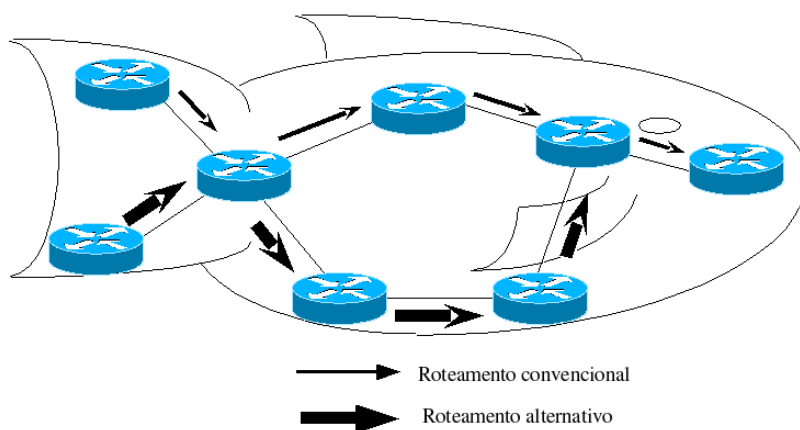


Figura 2.1: O problema da espinha de peixe

Podemos observar que a rota selecionada pelos protocolos IGP podem provocar congestionamentos, com a consequente perda de pacotes nas filas dos roteadores. A seleção de uma rota alternativa, mesmo que percorrendo um caminho mais longo, pode evitar o congestionamento, minimizar as perdas de pacotes e melhorar a QoS geral dos fluxos de dados.

A maior parte das abordagens de suporte à ET em redes IP necessitam que os protocolos convencionais de roteamento sensíveis ao estado dos enlaces (link state - LS), sejam es-

tendidos. Os protocolos LS como o OSPF só fornecem informações de topologia (enlace UP ou DOWN). Nenhuma informação adicional tal como banda máxima do enlace, banda reservável para o enlace (banda residual), uso atual da banda é fornecida pelos protocolos IGP atuais. As extensões dos protocolos LS visam distribuir, através de mensagens de notificações estas informações complementares.

No entanto, de acordo com (Kompella e Rekhter 2002) o emprego destes novos protocolos pode provocar uma inundação (flood) de mensagens de sinalização em redes de grande escala, podendo prejudicar o desempenho da rede. Estudos para aperfeiçoar as estratégias de sinalização com minimização da inundação de mensagens ainda são um grande desafio. A utilização destes protocolos deve ser criteriosa e o número de argumentos de estado deve ser reduzido, para evitar que a inundação de mensagens degrade o desempenho da rede. Além disso, estas extensões ainda são muito recentes e muitas delas são ainda propostas de *Internet Drafts*, como a proposta do protocolo OSPF-TE (Kompella 2003) e (Ishiguro 2003).

2.2 Taxonomia de Sistema de Engenharia de Tráfego

Segundo (Magalhães e Cardoso 2001) os sistemas de ET podem ser classificados de acordo com as seguintes características:

Dependente de tempo/Dependente de Estado: os sistemas de ET dependentes do tempo são aqueles onde os procedimentos empregados levam em conta informações históricas, baseadas em informações sazonais. Nos sistemas dependentes de estado, somente informações atuais são levadas em conta para a execução dos procedimentos de ET.

Online/offline: o cálculo de rotas nos procedimentos de ET são executadas de forma offline quando se tem um conhecimento prévio das informações sobre as requisições de serviço (fluxos de dados) submetidos à rede. Estes procedimentos offline estão associados a procedimentos de ET estáticos, como ações de planejamento de capacidade. Os procedimentos de ET online são aqueles onde as requisições de admissão de fluxos de dados são tratadas uma-a-uma à medida que são solicitadas à rede, sem conhecimento prévio das demandas anteriores.

Centralizado/distribuído: um procedimento de ET centralizado ocorre quando uma autoridade central coleta as informações de estado da rede, calcula as rotas e informa os resultados periodicamente aos roteadores. No caso distribuído, várias entidades recebem informações do estado da rede para processamento das tarefas de ET de forma conjunta.

Prescritivo/Descritivo: a ET prescritiva analisa as alternativas e recomenda uma ação (caminhos ou configurações a serem implementadas). A ET descritiva descreve o impacto

de diferentes ações de configuração, sem no entanto, recomendar uma.

Open-loop/close loop: um processo de ET open-loop não utiliza informações a respeito do estado atual da rede, ao passo que o close-loop utiliza.

Global/local: no sistema de ET global o cálculo da rota é feito com informações de toda a rede. A ET global é apropriada para otimização centralizada. No sistema de ET local a informação para processamento está relacionada com apenas uma porção da rede.

Além das classificações propostas na taxonomia apresentada em (Magalhães e Cardoso 2001) podemos incluir:

Dinâmico/estático: um sistema de ET dinâmico é aquele em que as requisições de serviço são continuamente submetidas à rede ao longo do tempo. Um sistema de ET estático é aquele empregado para um conjunto fixo de requisições de serviço, que podem ocorrer em um intervalo de tempo ou de forma sazonal. Em geral, procedimentos de otimização dinâmicos são executados por procedimento online e os estáticos por procedimentos offline.

2.3 Tecnologia MPLS: conceitos básicos e terminologia

Nos ambientes de roteamento convencionais os pacotes são encaminhados na rede com inspeção das tabelas de roteamento configuradas em cada roteador constituinte da topologia, numa técnica denominada salto-a-salto (hop-by-hop). Para tanto são usados protocolos de roteamento intradomínio (Interior Gateway Protocols - IGP), como o Routing Information Protocol (RIP) e o Open Shortest Path First (OSPF), bem como os protocolos de roteamento interdomínio, como o Border Gateway Protocol (BGP). Ou seja, cada pacote deve ser analisado (inspeção do cabeçalho do protocolo de camada de rede) em cada roteador, entre a origem e destino de um fluxo de dados.

Na figura 2.2, para encaminhar os pacotes do roteador A para roteador F, o roteador C referencia somente o endereço de destino do roteador F. O roteador C pode então determinar a melhor rota, de acordo com o protocolo de roteamento empregado. No caso específico do RIP, a rota selecionada será baseada na menor métrica (número de saltos) entre a origem e o destino dos dados. Se o protocolo for OSPF, a rota será baseada no custo total (normalmente as métricas usadas são o número de saltos e a largura de banda dos enlaces) entre a origem e destino dos dados. A decisão de seleção de rotas em ambos os casos é feita em cada nó de roteamento, não permitindo o estabelecimento prévio da rota a ser seguida pelos pacotes.

A fim de contornar esta limitação do protocolo IP, o Multiprotocol Label Switching (MPLS) integra recursos encontrados em tecnologias de camada 2 e 3 do modelo

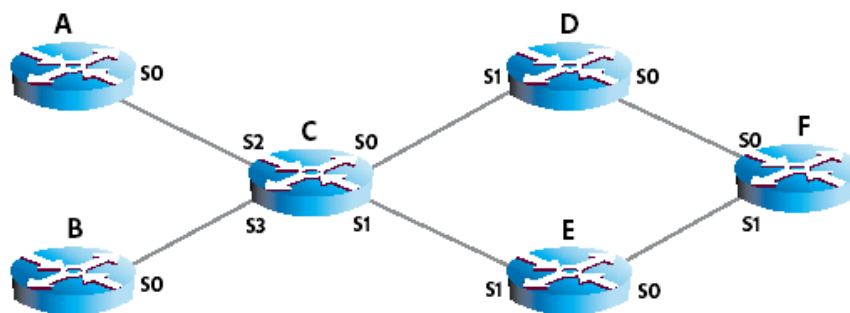


Figura 2.2: Exemplo de roteamento convencional

ISO/OSI. Na tecnologia MPLS, ao contrário dos protocolos IGP, não há inspeção de pacotes em todos os nós de roteamento (análise hop-a-hop). O MPLS combina a velocidade de encaminhamento da camada de enlace (camada 2) com a escalabilidade e inteligência da camada de rede (camada 3). Conforme a figura 2.3 os roteadores nas bordas de um domínio MPLS, os denominados *Label Edge Routers (LERs)*, inserem rótulos aos pacotes que entram no domínio, de acordo com as denominadas classes de equivalência de encaminhamento, ou *forward equivalence classes (FECs)*. Sendo assim, os pacotes são encaminhados através de uma rede MPLS baseados em sua associação com as FECs. À medida que os pacotes vão atravessando os roteadores de núcleo, denominados de *Label Switch Routers (LSRs)*, os rótulos vão sendo substituídos, de forma similar à troca de pares VPI/VCI em redes ATM, até que os mesmos cheguem ao roteador de borda de egresso, quando os rótulos são retirados e o pacote IP volta a ter sua semântica original.

Uma FEC é definida como sendo uma classe correspondente a um conjunto de pacotes encaminhados de uma mesma forma através da rede. Existem diversas formas de se definir uma FEC. Ela pode incluir todos os pacotes que têm o mesmo destino na rede. Por outro lado, uma FEC pode incluir todos os pacotes oriundos de uma mesma aplicação (porta de protocolo), ou com o mesmo conteúdo de campo TOS do cabeçalho IP. A associação do pacote à FEC é efetuada pelos dispositivos de borda (LERs). A definição das FECs permite a criação das bases de informação de rótulos ou *label information base (LIB)*, mantidas em cada LSR.

Com isso, na tecnologia MPLS, a decisão de roteamento dos pacotes, empregando roteamento convencional (hop-by-hop) só ocorre uma vez para cada FEC. Toda vez que um pacote for associado a uma FEC já existente, somente operações de encaminhamento de camada 2 são efetuadas, ou seja, ao chegar ao LSR, somente uma inspeção do rótulo, que tem tamanho curto e fixo é efetuada. O rótulo serve como índice de busca em uma tabela de encaminhamento, denominada base de informação de rótulos — *Label Information Base (LIB)*, a partir da qual é tomada a decisão de encaminhamento. Com isso, o MPLS apresenta um ganho de desempenho em relação ao roteamento IP convencional, uma vez

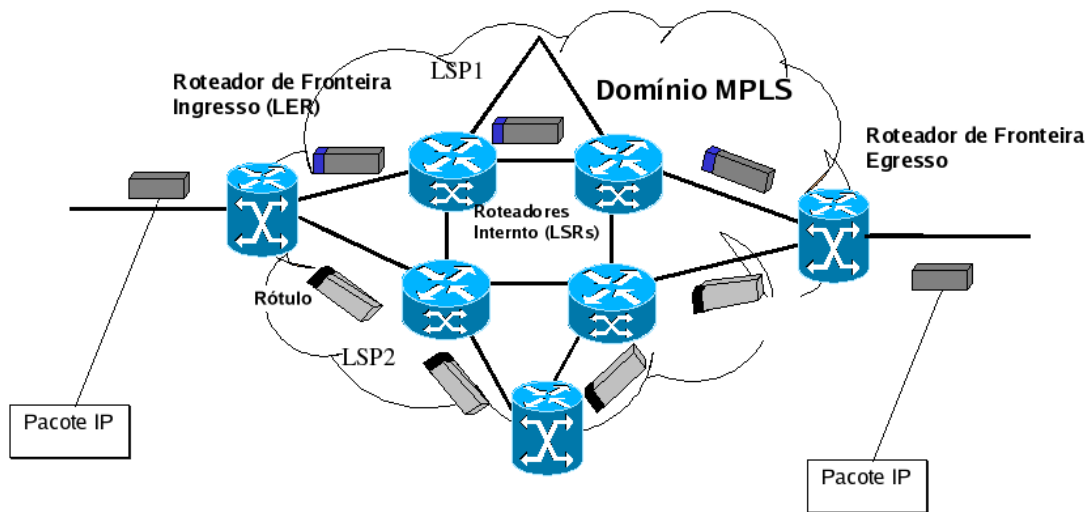


Figura 2.3: Encaminhamento de pacotes no MPLS

que o esforço de encaminhamento é feito somente na borda, quando é feita a associação do pacote à FEC, com inserção do rótulo ao mesmo.

Como a atribuição dos rótulos aos pacotes é feita na borda, ou seja, a associação dos pacotes às FECs pode ser feita previamente, no que se denomina *roteamento explícito* (Callon et al. 2001), o MPLS possibilita a implementação de ferramentas de engenharia de tráfego (ET), cujo objetivo é otimizar a utilização dos recursos de uma rede pela devida determinação das FECs e os caminhos pelos quais os pacotes associados a elas serão encaminhados.

Esta tecnologia é denominada multiprotocolo, uma vez que pode ser empregada de forma independente dos protocolos de camada de rede e enlace. A princípio, qualquer tecnologia de camada de enlace pode ser empregada (ATM, Frame Relay, Ethernet, PPP, redes ópticas). Na prática, em termos de camada de rede, o foco é o protocolo IP. No entanto, nada impede que a componente de controle do MPLS utilize outros protocolos de rede como IPX ou Appletalk.

2.3.1 Alguns conceitos e componentes da tecnologia MPLS

A tecnologia MPLS traz para as redes IP muitas características de outras tecnologias de rede como a ATM. Um dos principais benefícios é a separação entre as *componentes de encaminhamento* e *componentes de controle*. As componentes de encaminhamento são as estruturas responsáveis pela definição de qual interface de um dado LSR os pacotes serão enviados. A componente de controle, via de regra é representada por algum protocolo de roteamento convencional (RIP, OSPF, BGP, etc..), que associa uma rota a uma FEC.

A seguir são descritos alguns conceitos e componentes da tecnologia MPLS empregados ao longo do texto.

Rótulo

O rótulo é um conjunto de informações resumido e de tamanho fixo utilizado como índice de busca em uma tabela de encaminhamento do pacote para o seu destino. Ao contrário de um cabeçalho IP, o rótulo não contém um cabeçalho longo, mas simplesmente um valor numérico de tamanho fixo e curto, que identifica a qual FEC o pacote está associado. A associação do pacote a uma FEC normalmente é feita baseada no endereço da sub-rede de destino. No entanto, outros procedimentos de classificação mais complexos podem ser efetuados (campo TOS, campo de prioridade do padrão IEEE 802.1p ou algum algoritmo complexo de otimização de rede). Entretanto, o fato importante é que a complexidade de associação dos pacotes às FECs é feita somente na borda da rede.

Em algumas tecnologias de rede, estruturas pré-existentes são empregadas como rótulos MPLS. No ATM são utilizados os pares VPI/VCI. No Frame Relay o campo DLCI. Outras tecnologias como a Ethernet e PPP exigem a inclusão de uma estrutura adicional denominada *shim header*, conforme figura 2.4.

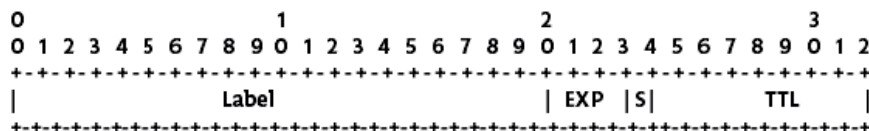


Figura 2.4: Formato do shim header

De acordo com a figura 2.4, o campo LABEL contém o valor do rótulo propriamente dito, com comprimento de 20 bits. O campo EXP (experimental) atualmente é empregado para implementação de diferenciação de serviços (mapeamento do campo TOS), com comprimento de 3 bits, permitindo até 8 classes de serviço. O campo S (stack) de 1 bit quando tem valor igual a 1 irá indicar que existe mais de um rótulo no shim header. Múltiplos rótulos no shim header são empregados para roteamento hierárquico intradomínios. O campo TTL com 8 bits de comprimento abriga um contador equivalente ao *Time To Leave* do cabeçalho IP, empregado para fins de prevenção de loop e compatibilidade retroativa com aplicações que empreguem o traçado de rotas, como o comando "traceroute" do Unix (Callon et al. 2001).

Base de informação de rótulos — Label Information Base (LIB)

A LIB é uma tabela localizada em cada roteador LSR que contém informações necessárias ao encaminhamento dos pacotes. A LIB contém basicamente um campo com o valor do rótulo de entrada, que funciona como índice de busca na LIB; um campo que define

a interface de entrada do LSR; um campo correspondente ao roteador de destino; um campo correspondente à interface de saída pela qual deverá ser encaminhado o pacote e um campo com o valor do rótulo que deverá ser substituído no pacote ao sair do LSR.

Caminhos comutados por rótulos — Label Switched Paths (LSP)

Um LSP é basicamente uma rota pré-definida, na qual um conjunto de pacotes de uma mesma FEC são encaminhados através de uma rede MPLS. Cada LSP é unidirecional. O tráfego de retorno, portanto, deve ser encaminhado em um LSP separado.

Protocolos de distribuição de rótulos

Para que as decisões de encaminhamento de pacotes sejam definidas em um domínio MPLS, é necessário que informações locais, mantidas nas LIBs de cada LSR sejam intercambiadas entre todos os LSRs vizinhos. Para tanto são necessários protocolos que distribuam as informações de rótulos. A referência (Callon et al. 2001) não especifica um protocolo de distribuição de rótulos específico, no entanto, foi desenvolvido um protocolo específico para distribuição destas informações no MPLS — o Label Distribution Protocol (LDP) (Anderson 2001). A especificação do LDP define um conjunto de procedimentos e mensagens através dos quais os LSPs são estabelecidos na rede MPLS.

Além do LDP, o Resource Reservation Protocol (RSVP), inicialmente empregado na arquitetura Serviços Integridos (Braden 1994), (Wroczowski 2001) é um protocolo de sinalização que, além de permitir a troca de informações de rótulos entre os LSRs, permite a sinalização de reservas de recursos na rede, o que o tem tornado popular entre os fabricantes. Além disso, por ser mais antigo, foi mais validado e está disponível em muitos equipamentos de mercado.

Nem o LDP, nem o RSVP, no entanto, incluem estratégias no estabelecimento de rotas explícitas, nem permitem o roteamento baseado em restrições. A implementação de roteamento explícito é o elemento chave para implementação de engenharia de tráfego. Para tanto, foram desenvolvidas as extensões dos protocolos de distribuição de rótulos. Estas extensões são o Constraint Routing - Label Distribution Protocol (CR-LDP) (Jamoussi 2002) e o Resource Reservation Protocol - Traffic Engineering (RSVP-TE) (Awduche 2001).

2.4 Roteamento baseado em restrições

Diferentemente dos algoritmos de roteamento convencionais, o roteamento baseado em restrições (RBR) é um conjunto de processos voltado ao cômputo de rotas de rede sujeito ao atendimento de uma ou mais restrições (regras). Um conjunto de regras poderia corresponder às especificações de requisitos de QoS exigidos pelas aplicações (ou configurações

especificadas pelo administrador de um domínio de rede). A partir deste conjunto de requisitos, que se apresentariam como restrições (regras) para o roteamento, poderiam ser computados os caminhos que melhor atendessem estas restrições.

Além de requisitos de QoS, outras informações poderiam ser utilizadas como restrições para o roteamento, como informações de segurança e tarifação. O cômputo das rotas baseado nestas restrições pode implicar em rotas de maior distância métrica (caminhos com maior número de saltos), porém mais adequadas do ponto de vista das restrições implementadas.

O RBR requer, entre outras coisas, capacidade de roteamento explícito (ou roteamento baseado na fonte), ou seja, as rotas serão computadas em função das necessidades de recursos (restrições exigidas pela fonte) para atingir o destino. Existem atualmente dois protocolos de distribuição de rótulos necessários à implementação de roteamento baseado em restrições sobre o MPLS. O primeiro estende o protocolo RSVP. O segundo, uma extensão do protocolo LDP, denominado CR-LDP (Constraint Based Routing Label Distribution Protocol).

2.4.1 RSVP-TE

A utilização do RSVP como protocolo de distribuição de etiquetas foi proposto em (Awduche 2001). Nesta proposta é definido o *RSVP Label Object* como extensão do RSVP para sua utilização como protocolo de troca de mensagens de informações para distribuição de etiquetas.

A utilização do RSVP estendido com o *Label Object* permite o estabelecimento do estado do LSP, computado pelos algoritmos de roteamento do componente de controle do MPLS, entre a origem e o destino. Isto porque a mensagem PATH do RSVP é controlada pelo algoritmo de encaminhamento baseado no destino, permitindo a determinação do caminho seguido pelo LSP. Quando um roteador encaminha uma mensagem PATH, o roteador utiliza a combinação das informações providas pelo algoritmos de roteamento (como RIP, OSPF, BGP) e o endereço de destino.

A fim de adequar o RSVP à característica de roteamento explícito do MPLS, é inserido um novo objeto denominado de *Explicit Route Object (ERO)*. Este objeto é enviado dentro da mensagem PATH e contém a rota explícita que a mensagem deve seguir. O encaminhamento desta mensagem pelo roteador é determinada não pelo endereço de destino contido no cabeçalho do pacote IP, mas pelo conteúdo do objeto ERO contido na mensagem PATH. O ERO consiste de uma seqüência ordenada de *saltos* que especificam a rota explícita. Cada *salto* é representado por um *nó abstrato* (Awduche 2001). Este *nó*

abstrato identifica um grupo de um ou mais roteadores cuja topologia é invisível a partir do roteador que computa a rota explícita.

2.4.2 Protocolo CR-LDP

O outro protocolo de distribuição de etiquetas, o LDP, também deve ser estendido para o suporte ao roteamento baseado em restrições. Este novo protocolo, denominado de CR-LDP (Constraint Based Routing Label Distribution Protocol), foi especificado em (Jamoussi 2002). O CR-LDP introduz um novo objeto na estrutura do LDP denominado *explicit route (ER)*. Sua estrutura é idêntica ao objeto ERO do RSVP, dentro do formato "type, length, value"(TLV) (Anderson 2001) suportado pelo LDP. O ER é transmitido juntamente com as mensagens de *LABEL REQUEST* do LDP (Jamoussi 2002)

2.5 MPLS e Engenharia de Tráfego

O MPLS favorece a implementação da ET em redes IP, uma vez que introduz o paradigma de orientação a conexão e roteamento explícito com o emprego dos protocolos RSVP-TE e CR-LDP. Com isso, os LSPs podem ser configurados seguindo um procedimento de ET que os configure (rotas, larguras de banda) de acordo com objetivos de ET como maximização de banda, minimização de atrasos, seguindo restrições de QoS.

De acordo com (Magalhães e Cardoso 2001) as vantagens do emprego do MPLS para ET são:

- O MPLS permite esquemas sofisticados de roteamento baseados na capacidade de estabelecimento de LSPs explicitamente roteados (não baseados em destino como os protocolos de caminhos mínimos);
- Possibilidade de mapeamento de troncos de tráfego em LSPs;
- Suporte ao roteamento baseado em restrições.

Etapas de um procedimento de ET sobre MPLS

As etapas básicas de um procedimento de ET podem ser descritas como abaixo:

1. Gerenciamento de caminhos:

Seleção de caminhos: Determina a rota explícita no nó de origem do LSP. Pode ser empregado o roteamento baseado em restrições para computação de rotas ou um algoritmo de otimização.

Estabelecimento de caminho: um protocolo de sinalização (que também serve para distribuir os rótulos) é utilizado para estabelecer o caminho de um LSP na estrutura física da rede. Atualmente são empregados o RSVP-TE e o CR-LDP.

Manutenção do caminho: manutenção e finalização de LSPs, incluindo os eventuais re-roteamentos.

2. **Atribuição do tráfego ao LSP:** Nesta etapa são classificados os vários perfis de tráfego (agregação) e alocação de LSPs existentes.
3. **Disseminação de informação do estado da rede:** Esta etapa é normalmente necessária para os procedimento de ET online. Ela pode empregar extensões dos protocolos de roteamento IGP convencionais, como o OSPF-TE.
4. **Gerencianmento da rede:** Nesta etapa é executado o monitoramento de desempenho e contabilização, bem como o gerenciamento de configuração e falhas.

2.6 Conclusões do capítulo

O objetivo deste capítulo foi apresentar uma breve revisão bibliográfica sobre os principais conceitos sobre engenharia de tráfego, com foco em redes IP sobre tecnologia MPLS. As definições e a taxonomia da engenharia de tráfego dão base à contextualização da proposta de trabalho. Além disso, foram apresentados os principais conceitos, protocolos e terminologias da tecnologia MPLS, que serão empregados ao longo do texto. Neste trabalho, como será descrito a seguir, será empregada uma estratégia de ET para solução de problemas dinâmicos. A abordagem utilizada nas soluções propostas será baseada no desenvolvimento de um modelo matemático, formulado numa notação de programação matemática e resolvido empregando algoritmos heurísticos. O próximo capítulo apresentará uma breve revisão bibliográfica sobre a teoria de otimização aplicada à engenharia de tráfego.

Capítulo 3

Engenharia de tráfego e otimização

Em geral os objetivos dos problemas de TE são relacionados com aproveitamento otimizado dos recursos de rede, sem violar os requisitos das aplicações.

A fim de atender a estes objetivos, o problema deve ser modelado, formulado e resolvido pelo emprego de técnicas que produzam soluções de boa qualidade e que sejam eficientes. O presente capítulo visa apresentar uma revisão bibliográfica sobre os fundamentos da teoria de otimização de sistemas, focando o modelamento e solução de problemas de engenharia de tráfego pelo emprego de programação linear inteira e soluções aproximadas através de relaxações e heurísticas.

Neste capítulo serão apresentados os seguintes assuntos: (i) abordagens para modelagem de problemas; (ii) modelos de programação matemática, com enfoque em programação linear inteira (PLI); (iii) abordagens de solução de problemas de PLI, com enfoque em métodos de aproximação, com o uso de relaxação Lagrangeana e heurísticas; (iv) problemas de engenharia de tráfego como problemas de otimização.

3.1 Modelagem de problemas

A representação formal da realidade é uma necessidade histórica da humanidade. Na impossibilidade de lidar diretamente com a realidade, seja por aspectos econômicos, seja pela sua complexidade, necessitamos criar uma visão bem estruturada dela, através da formulação de modelos representativos. De acordo com (Goldbarg e Luna 2000) "modelos são representações simplificadas da realidade que preservam, para determinadas situações e enfoques, uma equivalência adequada".

A Geometria Euclidiana, transformadas de Laplace, Teoria Quântica e o modelo atômico de Bohr, são exemplos típicos de modelos representativos. Ainda de acordo com

(Goldbarg e Luna 2000) o poder da representatividade do modelo é sua característica mais desejável. A capacidade de simplificação de um modelo lhe confere factibilidade operacional. Existem vários critérios de medida ou adequação de um modelo à realidade. Em várias situações a representatividade de um modelo é obtida de forma interativa. O processo de verificação da representatividade do modelo, denominada de *validação do modelo* é uma etapa indispensável para a qualificação do processo científico.

O processo de modelagem de problemas pode de uma forma geral ser resumido nas seguintes etapas: (i) definição do problema; (ii) formulação; (iii) solução e validação.

A definição do problema é uma das fases mais importantes do processo. O problema deve ser traduzido em elementos concretos, entre eles:

- objetivos;
- variáveis de decisão ou controle;
- níveis de detalhamento e;
- restrições.

A etapa da formulação do modelo consiste na tradução do problema em termos de uma linguagem de especificação, normalmente expressa de forma matemática, podendo trazer em seu bojo, fórmulas, equações e expressões matemáticas. A formulação pode ter um enfoque altamente quantitativo, porém o sucesso nesta etapa, em geral, depende de fatores mais subjetivos como: intuição, experiência, criatividade e poder de síntese. A formulação representa, via de regra, a etapa mais difícil no processo de modelagem de problemas.

A etapa de solução e validação implica na concepção de uma abordagem de resolução da formulação, pela adoção de técnicas adequadas. Para validação podem ser empregadas soluções analíticas, simulações ou comparações com protótipos. Em geral é um processo iterativo e interativo, onde as soluções apontadas em um dada iteração podem ser empregadas para refinamento do modelo.

3.1.1 Modelos de programação matemática

A formulação e resolução de problemas de otimização faz emprego de modelos quantitativos expressos matematicamente. Tal abordagem é amplamente utilizada num ramo do conhecimento denominado *Pesquisa Operacional (PO)*. A Pesquisa Operacional visa a otimização de um processo, seja pela maximização ou minimização de um dado critério, como por exemplo: maximização da produção de um dado insumo, sujeito a determinadas condições de operação, as quais denomina-se genericamente de restrições.

Os principais modelos de PO podem ser expressos segundo modelos de *Programação Matemática*, onde o termo programação aparece com o sentido de planejamento, estruturação. Por outro lado, é inevitável o emprego de ferramentas computacionais para solução dos problemas formulados em notação de programação matemática, tendo em vista que na maioria dos problemas realmente interessantes, as instâncias a serem analisadas implicam em grande número de variáveis de decisão e restrições.

Um problema de otimização, expresso em notação de programação matemática tem, em geral, a seguinte notação:

(P) = Maximizar ou minimizar uma função objetivo $f(x)$

Sujeito a (restrições):

$$h_i(x) = 0, i = 1, \dots, p$$

$$g_j(x) \leq 0, j = 1, \dots, q$$

$$x \in \mathbb{R}^n$$

Uma notação compacta do problema é dada por:

(P) = Maximizar $\{f(x) \mid h_i(x) = 0, g_j(x) \leq 0, i = 1, \dots, p, j = 1, \dots, q, x \in \mathbb{R}^n\}$,

Entre as principais áreas de formulação e solução de problemas de modelos de programação matemática podemos citar: *Programação Linear*, *Programação Não Linear*, *Programação Linear Inteira*.

A *Programação Linear* é um caso particular dos modelos de programação matemática onde a função objetivo e as restrições são lineares.

A *Programação Não-Linear* é um caso particular da programação matemática quando houver algum tipo de não-linearidade, seja na função objetivo ou nas famílias de restrições. A questão da não-linearidade é bem tratada no caso das funções convexas. Problemas não lineares com funções não convexas são mais difíceis de tratar e são muito comuns no estudo da teoria econômica e muitas outras áreas.

A *Programação Linear Inteira* é um caso especial da programação linear, onde as variáveis só podem assumir valores inteiros (discretos). Os problemas formulados em notação inteira de maior interesse tem, em geral, grande complexidade computacional, sendo normalmente resolvidos por processos aproximativos, como heurísticas e relaxações.

3.1.2 Problemas NP-Completos

Os problemas formulados em notação de programação matemática, invariavelmente são resolvidos a partir de algoritmos processados em computadores. De acordo com (Ziegelmann 2001) a teoria da complexidade nos permite classificar os problemas em duas grandes categorias:

- *Problemas fáceis*, solúveis por algoritmos executáveis em tempo polinomial;
- *Problemas difíceis* que nem sempre podem ser resolvidos de forma direta em tempo polinomial e onde os algoritmos conhecidos têm tempo de execução exponencial.

Definição 3.1 *Define-se na teoria da complexidade um problema de tomada de decisão (D) aquele que pode ser respondido com sim ou não. Esta definição simplifica bastante todo o restante da teoria. Muitos problemas podem ser facilmente convertidos em problemas de decisão. Por exemplo, ao invés de procurar pelo tamanho máximo em um dado gráfico, questiona-se a existência de um valor $> k$. Resolvendo-se o problema de decisão, pode-se resolver o problema original.*

A partir desta definição, diz-se que um problema é da classe de problemas polinomiais — problemas do tipo P , se existe um algoritmo conhecido que consegue resolver o problema D em tempo polinomial. Um exemplo típico de problema de decisão do tipo P é o problema de escolha de caminhos mínimos em um grafo.

Um problema de decisão D é classificado como NP (*não-deterministicamente polinomial*), se a cada instância I com resposta *sim*, sua solução a princípio tem ordem exponencial, pois não se conhece nenhum algoritmo que possa fornecer esta resposta em tempo polinomial. Observe que o termo *não-determinístico* não implica aleatoriedade, mas que apenas não se pode afirmar a existência de um algoritmo de complexidade polinomial para o problema em consideração.

Um problema de decisão D é enquadrado na classe dos problemas NP -difíceis, se e somente se, qualquer problema do tipo NP pode ser reduzido a D . Um problema de decisão D é do tipo NP -completo se (i) $D \in NP$ e (ii) D é NP -difícil.

Um típico exemplo de problemas NP -completos é o problema da mochila (Dantzig 1957), bem como o problema de caminhos mínimos sujeitos a restrições. A implicação deste fato é que

os problemas desta natureza fatalmente consomem um tempo de solução exponencial e são intratáveis para instâncias de maior porte pelo consumo excessivo de tempo. Como ainda não se conseguiu provar que existe algum algoritmo que possa resolver os problemas *NP*-completos em tempo polinomial, em geral, eles são resolvidos de forma aproximada, empregando-se relaxações ou heurísticas.

Maiores detalhes sobre a definição, certificação, teoremas sobre complexidade dos algoritmos e problemas *NP* podem ser obtidos em Cormen (Cormen et al. 1990), Wolsey (Wolsey 1998), Papadimitriou (Papadimitriou 1998) e (Garey e Johnson 1979)

3.2 Programação Linear Inteira (PLI)

Uma variedade de problemas reais podem ser formulados e resolvidos empregando-se notação de Programação Linear Inteira (*PLI*). Entre eles podemos citar: o agendamento de trens; agendamento de tripulações de aviões; alocação de recursos em sistemas de telecomunicações. Todos eles visam resolver as mesmas categorias de problemas: otimização da utilização dos recursos, onde as variáveis utilizadas são inteiras, pois por exemplo, não se pode dividir um trem em frações; nem alocar meio piloto em uma viagem. Os recursos de um sistema de telecomunicações podem ser especificados com valores inteiros, por exemplo: um enlace de rede com capacidade de transmissão expressa sob forma de números inteiros, quantidades inteiras de nós de redes, etc.

3.2.1 Problema geral

(PL) Maximizar $c^T x$

Sujeito a:

$$Ax \leq b$$

$$x \geq 0$$

Onde A é uma matriz $m \times n$; c um vetor linha n -dimensional; b um vetor coluna m -dimensional e x um vetor coluna n -dimensional.

Se adicionarmos às restrições algumas variáveis inteiras, temos a seguir, um típico problema de programação inteira mista (PIM).

(PIM) Maximizar $cx + hy$

Sujeito a:

$$Ax + Gy \leq b$$

Se todas as variáveis forem inteiras, temos um exemplo típico de problema de programação linear inteira (PLI).

(PLI) Maximizar cx

Sujeito a:

$$Ax \leq b$$

$$x \geq 0$$

x e b são inteiros

3.2.2 Problema da mochila

O chamado problema da mochila ou "*knapsack problem*" é um dos problemas mais representativos no estudo da teoria de programação matemática. Em suma, o problema da mochila pode ser entendido como o desafio de encher uma mochila sem ultrapassar um limite de peso, maximizando o valor agregado dos produtos carregados. O problema é uma metáfora e se caracteriza como um problema representativo de otimização na alocação de recursos sujeito a restrições. Foi reportado pela primeira vez na literatura por Dantzig (Dantzig 1957) e constitui um marco das técnicas de programação inteira.

Suas aplicações mais típicas aparecem nas áreas de: investimento de capital; problemas de corte e empacotamento de materiais; carregamento de veículos, orçamento, e recentemente, na área de alocação de recursos de telecomunicações. O problema da mochila (PM) tem a formulação a seguir:

(PM) Maximizar $\sum_{j=1}^n c_j x_j$

Sujeito a:

$$\sum_{j=1}^n w_j x_j \leq b$$

$$x_j \in Z_+ \text{ para } j = 1 \dots, n$$

Onde x_j representa o número de objetos do tipo j a serem incluídos na mochila, selecionado de acordo com o critério de restrição do modelo. Para tanto, sabe-se que a mochila tem capacidade b e c_j é o custo do objeto do tipo j (seu valor econômico) e w_j é o peso (kg) do objeto. Os coeficientes da restrição podem ser aproximados para números inteiros, sem perda de generalidade.

Um problema similar é a alocação de recursos em um sistema de telecomunicações, onde o objetivo é admitir o maior número de fluxos de dados em uma rede, com o menor custo administrativo (capacidade dos enlaces), sem que a capacidade destes enlaces seja violada.

3.2.3 Métodos de solução exatos para problemas de programação linear inteira

Dentre as principais técnicas de solução dos problemas de programação linear inteira (PLI) podemos citar: (i) o método de *branch-and-bound* (BB); (ii) *programação dinâmica* (PD); (iii) emprego de relaxações em procedimentos de solução e (iv) soluções heurísticas. Os dois primeiros (BB e PD) são métodos de solução exatos e serão apresentados a seguir.

3.2.3.1 Branch and Bound

O método de *branch-and-bound* (BB) baseia-se num processo de enumeração de pontos candidatos à solução ótima inteira. É portanto um método exato. É o método do tipo "divisão e conquista", onde o problema a ser resolvido pode ser decomposto pelo particionamento do espaço de soluções, motivo do nome *branch*. O nome *bound* ressalta que a prova da otimalidade da solução utiliza os limites calculados ao longo da enumeração.

O método BB pode ser definido matematicamente da seguinte forma:

Seja:

$$P = \text{Maximizar } \{cx \mid Ax = b, x \geq 0, x \in Z^+\}$$

$$P' = \text{Maximizar } \{cx \mid Ax = b, x \geq 0, x \in \mathfrak{R}^+\}$$

Definindo ainda $V^*(P)$ e $V^*(P')$ os valores da função objetivo nos pontos ótimos (P) e (P') respectivamente, temos que:

$$V^*(P) \leq V^*(P')$$

Se \tilde{x} é uma solução viável de (P), então:

$$V(\tilde{x}) \leq V^*(P) \Rightarrow C^T \tilde{x} \leq V^*(P)$$

Seja $V^*(P')$ um limite superior para (P) e qualquer de suas soluções viáveis e seja \hat{x} um vetor de soluções ótimas de (P') tal que \hat{x}_j é não inteiro, temos:

Se $\tilde{x}_j \geq \hat{x}_j + 1$ ou $\tilde{x}_j \leq \hat{x}_j$ para toda solução viável de (P) .

O problema (P) pode ser dividido em dois novos problemas (P_1) e (P_2) , de forma que a envoltória convexa - $Conv\{(P_1) \cup (P_2)\}$ está estritamente contida em $Conv\{(P)\}$.

Com isso, a solução dos sub-problemas (P_1) e (P_2) fica simplificada e a união destas soluções comporá a solução de (P) . Cada sub-problema, por sua vez, poderia ser subdividido, e assim por diante, gerando uma árvore de partição do BB , tornando cada sub-problema nos ramos extremos da árvore, cada vez mais fáceis de resolver.

Um exemplo detalhado da aplicação do método BB pode ser vista em (Goldbarg e Luna 2000).

O problema do emprego do método BB em problemas NP-completos é a explosão combinatoria provocada na árvore de partição, tornando o método inviável computacionalmente de ser aplicado em instâncias grandes.

3.2.3.2 Programação dinâmica (PD)

A programação dinâmica (PD) é uma técnica de solução exata para problemas de programação matemática do tipo otimização de processos multiestágios. Um processo multiestágio é aquele em que pode ser desdobrado segundo um certo número de etapas sequenciais (estágios).

Basicamente, o processo de programação dinâmica consiste em se tomar uma *decisão* para um determinado estágio baseado na decisão (estado) do estágio anterior. Com isso, o processo de otimização também é decomposto, simplificando a solução. Neste processo, o objetivo do analista é encontrar uma política ótima (chamada trajetória ótima) em relação ao retorno auferido com as decisões anteriores. Este processo pode ser determinístico se o resultado de cada decisão for conhecido exatamente.

A técnica de programação dinâmica pode se tornar complexa para o caso de problemas NP-difíceis ou NP-completos.

3.3 Métodos de soluções aproximados dos problemas de PLI

Nesta seção serão discutidos dois métodos aproximativos de solução de problemas de programação linear inteira: (i) pelo emprego de relaxações e (ii) pelo emprego soluções heurísticas.

3.3.1 Emprego de relaxações

Como vimos, nem sempre os PLI podem ser resolvidos de forma eficiente por métodos exatos como o *BB* e a *PD*, seja por motivos de viabilidade, do ponto de vista da otimização, ou de eficiência computacional.

Uma alternativa interessante às soluções exatas é a adoção de um procedimento que permita a busca de um conjunto de soluções para um problema de PLI que seja o mais próximo possível das soluções ótimas, o qual é definido como relaxação. Normalmente os procedimentos de relaxação podem ser resolvidos por processos iterativos. Para tanto devem ser determinados os limiares superiores (*upper bounds*) ou os limiares inferiores (*lower bounds*) respectivamente para os problemas de maximização e minimização, que são empregados para estabelecer os limites máximos e mínimos que podem ser assumidos pelas funções objetivo, balizando estes procedimentos iterativos.

A principal vantagem das relaxações é de permitir ao analista estabelecer uma relação de compromisso entre a *qualidade da solução* (o quão próximo da solução ótima estão as soluções encontradas) e o *tempo computacional*. Muitas vezes é possível encontrarmos soluções ótimas para problemas NP-difícil ou NP-completo, utilizando-se algum método de relaxação, com tempo computacional viável, mesmo em problemas difíceis.

Matematicamente poderíamos expressar o problema de relaxação da seguinte forma, dado o seguinte problema de programação:

Sendo $f(x)$ uma função objetivo a maximizar; x a variável de decisão pertencente ao domínio X e Z^n é o espaço de soluções, temos:

$$z = \max \{f(x) : x \in X \subseteq Z^n\}$$

Como definir que um dado ponto x^* é um ponto ótimo ?

Definindo um limite superior (upper bound) $\bar{Z} \geq z$ e um limite inferior (lower bound) $\underline{Z} \leq z$

e z_s a solução ótima, estará sem sombra de dúvida entre estes dois limiares.

Podemos encontrar uma solução aproximada Z_t , de forma iterativa, utilizando como critério de parada a expressão:

$$|\bar{Z} - \underline{Z}| \leq \epsilon$$

Onde ϵ é uma constante pequena e não negativa.

Definição: Um problema de relaxação (PR): $z^R = \max \{f(x) : x \in T \subseteq \mathbb{R}^n\}$ é uma relaxação de um problema linear inteiro (PLI): $z = \max \{c(x) : x \in X \subseteq \mathbb{R}^n\}$ se:

(i) o espaço de variáveis de solução de (PLI) $X \subseteq T$ está contido no espaço de decisão de (PR).

(ii) $f(x) \geq c(x), \forall x \in X$.

Proposição 3.1 Se (RP) é uma relaxação de (PLI), então: $z^R \geq z$.

Note que sempre $\bar{Z} \geq Z_t$, pois \bar{Z} é um limite superior.

Na seção seguinte será abordado o método de relaxação Lagrangeana para solução de problemas de PLI.

3.3.1.1 Relaxação Lagrangeana

Considere o seguinte problema de programação inteira (PLI)

(PLI) $z = \max c^T \cdot x$

sujeito a:

$$A \cdot x \leq b$$

$$D \cdot x \leq d$$

$$x \in Z_+^n$$

Suponha que a família de restrições $Ax \leq b$ são restrições fáceis de serem computadas. Suponha, por outro lado, que a família de restrições $Dx \leq d$ são restrições difíceis. Sendo assim, relaxar a segunda família de restrições torna a solução do problema original (PLI) mais eficiente.

A relaxação Lagrangeana de (PLI) consiste em incluir a família de restrições difíceis em sua função objetivo multiplicada por um vetor μ , denominado *multiplicador Lagrangeano*, transformando o problema (PLI) em um problema de relaxação Lagrangeana.

$$\text{(PRL)} \quad z(\mu) = \max \{c^T \cdot x + \mu^T (d - Dx)\}$$

sujeito a:

$$Ax \leq b$$

$$x \in Z_+^n$$

Como (PRL) é uma relaxação de (PLI) $z(\mu) \geq z$ é um limite superior para (PLI) se $\mu \geq 0$. Para encontrar o menor (i.e o melhor) limite superior de (PLI), que tende a se aproximar da solução ótima, devemos resolver o problema Lagrangeano Dual (LD).

$$\text{(LD)} \quad Z_{LD} = \min\{z(\mu) : \mu > 0\}$$

A solução da relaxação Lagrangeana pode muitas vezes levar à solução ótima do problema de (PLI).

Algoritmo subgradiente

Via de regra, a solução de (LD) consiste na escolha adequada dos valores do vetor μ . Estes correspondem a fatores de penalidade, evitando que as restrições do problema primal sejam violadas. Mesmo que o problema LD seja côncavo, porém não diferenciável, caso típico de problemas do tipo da mochila, soluções que empreguem gradientes, como o método de Newton, são inviáveis, uma vez que (LD) é de natureza não diferenciável (?). Sendo assim, o *algoritmo subgradiente* desponta como uma solução atraente para o problema de relaxação Lagrangeana Dual.

De uma maneira simples o *algoritmo subgradiente* usa o multiplicador Lagrangeano como função de penalidade para reduzir ou até mesmo eliminar a violação das restrições relaxadas, como dito anteriormente. Admitindo-se um valor inicial para o vetor multiplicador Lagrangeano μ_0 , o valor de μ é decrementado de um valor correspondente ao subgradiente da solução inicial se as restrições relaxadas não são violadas, caso contrário incrementa-se. Uma nova iteração é executada e o processo continua até obter-se a convergência. Ao final do processo, a solução de (LD), que é um limite superior de (PLI) pode ser uma solução ótima, se e somente se, não violar as restrições do problema original.

Um dos problemas do algoritmo subgradiente é que não existe garantia que a solução ótima de LD seja uma solução viável para o problema primal, em decorrência de erros de aproximação entre as soluções primal e dual.

Portanto, em geral, a solução da relaxação Lagrangeana deve ser complementada por um procedimento heurístico que recupere a viabilidade do problema primal.

3.3.2 Soluções Heurísticas

O trabalho de Garey e Johnson (Garey e Johnson 1979) é um marco para o entendimento da dificuldade de solução de problemas NP-completos.

Uma vertente dos algoritmos aproximativos são as relaxações. A outra vertente são os denominados algoritmos heurísticos. O termo heurística tem origem na palavra grega *heuriskein*, que significa "descobrir", "achar".

De acordo com (Goldbarg e Luna 2000): "Uma heurística é uma técnica que busca alcançar uma boa solução, utilizando um esforço computacional considerado razoável, sendo capaz de garantir a viabilidade ou otimalidade da solução encontrada, ou ainda, em muitos casos, ambas, especialmente nas ocasiões em que essa busca parte de uma solução viável próxima ao ótimo."

Normalmente os procedimentos heurísticos podem ser antecidos por procedimentos de relaxação, permitindo obter-se soluções ótimas ou sub-ótimas. No presente trabalho será empregada uma abordagem de relaxação Lagrangeana para solução de um problema de engenharia de tráfego, com auxílio de uma heurística que verifica e, quando necessário, recupera a viabilidade do problema primal. Além disso, o cerne da proposta de trabalho foi o desenvolvimento de um procedimento puramente heurístico para solução de problemas de ET em redes IP sobre tecnologia MPLS.

3.4 Engenharia de tráfego como um problema de otimização

Esta seção abordará de forma sucinta os principais conceitos e problemas de otimização relacionados com alocação de recursos em sistemas de telecomunicações, mais especificamente nos problemas de localização de caminhos mínimos em grafos, sujeitos a restrições, que é a base para modelamento e solução de problemas de Engenharia de Tráfego em redes de comunicação de dados.

O conceito de grafo é simples e intuitivo. Grafo é uma representação gráfica da inter-

dependência entre elementos que são denominados *vértices* ou *nós*. Esta relação de interdependência é representada por um arco que une estes vértices.

Matematicamente, define-se grafo como um conjunto V de vértices interligados por A arcos, tendo sua notação $G = (V, A)$.

Um *grafo direcionado* $G = (V, A)$ é aquele que consiste de V vértices cuja interligação é ordenada. Um arco com origem em u e término em v é diferente do arco com origem em v e término em u . Um arco com origem no vértice u e término no vértice v é representada pela notação $e = (u, v)$. A cada arco podem ser atribuídos atributos, como custo, capacidade, etc. Note que $e' = (v, u)$ é um vértice diferente de e , podendo ter atributos distintos.

O emprego de grafos é muito comum em vários problemas de otimização e sobretudo nos problemas relacionados com fluxos em redes. Sua utilização na solução de problemas de Engenharia de Tráfego é inquestionável.

3.4.1 O problema de caminhos mínimos (PCM)

Sendo u e v dois vértices no grafo $G = (V, A)$, o caminho mais curto entre u e v é uma sequência de arestas que, passando por vértices distintos, liga u a v de forma a acumular o menor comprimento (sequência de vértices).

A formulação do problema é dada a seguir:

$$\text{(PCM) Minimizar } z = \sum_{(i,j) \in A} c_{ij} x_{ij}$$

sujeito a:

$$\sum_{\{j:(i,j) \in A\}} x_{ij} - \sum_{\{j:(j,i) \in A\}} x_{ji} = b_i$$

$$x_{ij} \in \{0, 1\} \forall (i, j) \in A$$

Onde: $b_i = 1$ se $i = o$; $b_i = 0$ se $i \neq o$ ou $i \neq d$; $b_i = -1$ se $i = d$.

Os vértices o e d representam respectivamente a origem e o destino do caminho. O coeficiente c_{ij} representa o custo da aresta e x_{ij} indica se aresta (i,j) faz parte ou não do caminho a ser selecionado. A restrição garante que o caminho selecionado é contínuo, ou seja, que não hajam caminhos paralelos para o mesmo destino.

3.4.2 Algoritmo de Dijkstra

O algoritmo mais popular e um dos mais eficientes para solução do problema de caminhos mínimos foi sugerido por Dijkstra em 1957. Ele é válido para grafos em que os arcos tem pesos não negativos e utiliza o método da indução e ajuste, com fácil implementação computacional. Para tanto, utiliza as seguintes definições:

- *Lista F* (lista de nós fechados): é um conjunto de vértices para os quais já se conhece um caminho mínimo;
- *Lista A* (lista de nós abertos): um conjunto de nós para o qual ainda não se conhece um caminho mínimo;
- *t*: contador de iterações;
- *V*: representando o conjunto dos nós rotulados de *G*;
- *r*: índice do vertice fechado na iteração *t*;
- $C = [c_{ij}]$: matriz de custos, representando a distância métrica entre os vertices ligados diretamente;
- d_{ij} : distância entre os vertices x_i e x_j
- d_{ij}^t : distância calculada entre os vertices x_i e x_j na iteração *t*;
- $rot(i)$: vetor que armazena o vértice que deu origem à distância calculada para o vértice de índice *i*;
- $\Gamma^+(r)$: conjunto de índices vizinhos do vértice de índice *r*.

O algoritmo de Dijkstra, abaixo, calcula a distância do vértice x_1 aos demais vértices.

Algoritmo de Dijkstra:

Ler os dados de $G = (N, A)$

Inicialização de variáveis:

$d_{11} \leftarrow 0$;

$d_{1i} \leftarrow \infty; \forall i \in N - \{x_1\}$;

$V \leftarrow x_1$;

$A \leftarrow N$;

$F \leftarrow \emptyset$;

$rot(i) \leftarrow 0; \forall i \in N$

Para $t = 1$ até n faça

$r \leftarrow x_i$ tal que $d_{1i} \leftarrow \min\{d_{1i} : i \in A\}$;


```

 $F \leftarrow F \cup \{r\};$ 
 $A \leftarrow A - r;$ 
 $V \leftarrow A \cap \Gamma^+(r);$ 
Para  $i \in V$  fazer
     $p \leftarrow \min\{d_{1i}, (d_{1r} + c_{ri})\}$ 
    Se  $p < d_{1i}$ , então  $d_{1i} \leftarrow p$  e  $rot(i) \leftarrow r$ 
FimPara

```

FimPara

Empregando-se uma fila com prioridade baseada em uma sequência de Fibonacci para determinação dos mínimos (ver maiores detalhes em (Cormen et al. 1990)), a complexidade do algoritmo de Dijkstra é do tipo $O(|N| \log |N| + |A|)$, portanto solúvel em tempo polinomial, pois como esperado, o problema de caminhos mínimos é do tipo P .

O algoritmo de Dijkstra é largamente empregado em problemas de roteamento em rede de comunicação, sendo a base dos protocolos Interior Gateway Protocol (IGP) como o Routing Information Protocol (RIP) e o Open Short Path First (OSPF).

3.4.3 O problema de caminhos mínimos sujeito a restrições

Como discutido no capítulo anterior, o advento da tecnologia MPLS possibilitou a extensão das funcionalidades do protocolo IP, favorecendo a implementação da engenharia de tráfego, pela possibilidade de execução do roteamento explícito.

A fim de garantir os requisitos das aplicações e satisfazer a expectativa dos usuários, a seleção de rotas entre origem e destino dos fluxos de dados deve ser definida de acordo com um conjunto de restrições, especificadas em termos de parâmetros de QoS como: vazão mínima, atraso máximo fim-a-fim, etc.

Esta modalidade de roteamento está relacionada com uma família de problemas denominados "problema de caminhos mínimos sujeitos a restrições (PCMSR)" que é uma extensão do problema de caminhos mínimos (PCM).

Ao contrário do PCM, que tem complexidade polinomial e pode ser resolvido eficientemente pelo algoritmo de Dijkstra, o PCMSR é difícil de resolver pois é um problema NP-completo (Girish et al. 2000).

Considerando uma rede representada por um grafo direcionado, onde cada aresta possui

parâmetros de custo administrativo e atraso de transmissão, podemos definir um *problema de caminhos mínimos sujeito a restrições* da seguinte forma:

Definição 3.2 Dado $G = (V, A)$ um grafo com $|V| = n$ e $|A| = m$, tal que cada aresta $a \in A$ é associada com um custo c_a e um atraso d_a . Dados o e $d \in V$ como sendo os nós de origem e destino e sendo T o valor do atraso máximo, representado por um inteiro positivo. O **problema de caminhos mínimos sujeito a restrições** é definido como o problema de encontrar um caminho p do grafo G partindo da origem o até o destino d com atraso total menor que T , num custo mínimo. Matematicamente:

$$(PCMSR) z = \min \sum_{(i,j) \in A} c_{ij} x_{ij}$$

Sujeito a:

$$\sum_{\{j:(i,j) \in A\}} x_{ij} - \sum_{\{j:(j,i) \in A\}} x_{ji} = b_i$$

$$x_{ij} \in \{0, 1\} \forall (i, j) \in A$$

Onde: $b_i = 1$ se $i = o$; $b_i = 0$ se $i \neq o$ ou $i \neq d$; $b_i = -1$ se $i = d$.

Esta é a formulação básica do problema a ser resolvido ao longo deste trabalho.

3.5 Conclusões do capítulo

O objetivo deste capítulo foi proporcionar uma breve revisão bibliográfica sobre a teoria de otimização a fim de subsidiar a leitura do próximo capítulo.

O emprego de programação matemática para modelamento de problemas de engenharia de tráfego (ET) tem se tornado uma alternativa promissora. Os problemas de ET, em geral, podem ser formulados como problemas de programação linear inteira, sendo variantes do problema de caminhos mínimos sujeito a restrições, que como visto, é do tipo NP-completo. A solução destes problemas pode ser obtida em tempo hábil, através de métodos aproximativos, como as relaxações e heurísticas. No presente trabalho, foram adotados os métodos de relaxação Lagrangeana, com o emprego do algoritmo subgradiente para obtenção de uma solução viável, em tempo polinomial, para um problema de engenharia de tráfego em redes IP sobre MPLS.

Capítulo 4

Modelagem de um Problema de Engenharia de Tráfego (PET)

Neste capítulo serão apresentados: (i) Problema de Engenharia de Tráfego; (ii) modelagem e formulação do problema; (iii) abordagem de solução Lagrangeana; (iv) abordagem de solução heurística pura; (v) experimentos e seus resultados; (vi) conclusões do capítulo.

4.1 Problema de Engenharia de Tráfego

O objetivo deste problema de ET visa a maximização da vazão global dos fluxos de dados injetados em uma topologia de rede de tamanho significativo, através da seleção de caminhos mínimos e larguras de bandas a serem configurados em LSPs nos quais os fluxos de dados serão roteados, respeitando-se as restrições de largura de banda dos enlaces e atraso fim-a-fim máximo de cada fluxo de dados. Este problema é enquadrado como um *problema de caminhos mínimos sujeito a restrições* (Girish et al. 2000), sendo do tipo NP-completo (Papadimitriou 1998).

Na modelagem deste problema será adotada uma política de diferenciação de serviços a partir do controle de admissão e da configuração de largura de banda dos LSPs, baseadas em parâmetros de prioridade de cada fluxo de dados. Os fluxos serão classificados em *alta prioridade e baixa prioridade*. Seguindo esta política, na eminência de congestionamento da rede, os fluxos de baixa prioridade poderão ter suas vazões reduzidas até zero (fluxos rejeitados) para garantir os requisitos de QoS dos fluxos de alta prioridade.

A vazão será modelada em níveis, ou seja, no final do procedimento, cada LSP será confi-

gurado com um dos possíveis níveis de largura de banda, que pode variar de um nível de largura de banda igual a zero até um valor máximo desejado.

4.2 Modelagem e formulação do problema

A modelagem e formulação do problema de ET proposto tem como objetivo definir as características topológicas e lógicas da rede, a função objetivo, as variáveis de decisão, e as famílias de restrições.

As características topológicas são modeladas por um grafo direcionado $G = (V, E)$, onde $V = \{1, \dots, n\}$ corresponde aos nós da rede e $E = \{1, \dots, m\}$ aos enlaces de transmissão. A capacidade de transmissão de um enlace (i, j) é μ_{ij} Kbps, e seu custo administrativo c_{ij} corresponde ao atraso de transmissão provocado pelo mesmo.

As características lógicas da rede são: o número de LSPs a serem configurados é K . O número de níveis de largura de banda a serem configurados no k -ésimo LSP é l_k . O procedimento de otimização escolhe somente um nível para configurar cada LSP onde serão encaminhados os fluxos. λ_k^l é a taxa de transmissão em Kbps do k -ésimo LSP, onde o índice l corresponde ao nível a ser configurado no LSP. O objetivo em se discretizar a vazão é permitir a formulação do problema em termos de programação matemática inteira, o que significa um tratamento computacional menos complexo do que num problema não-linear (Wolsey 1998). O máximo atraso fim-a-fim é h_k , permitido para o k -ésimo LSP, medido em ms. Além disso, um parâmetro δ_k será introduzido na função objetivo para indicar a importância do k -ésimo fluxo de dados, que será encaminhado no k -ésimo LSP, denominado parâmetro de prioridade de admissão, o qual atua como elemento de diferenciação de serviços.

A função objetivo visa maximizar a vazão global dos fluxos de dados injetados na rede, segundo uma política de priorização.

As variáveis de decisão permitem a seleção do nível de largura de banda a ser configurado em cada LSP, além da definição do caminho entre a fonte e o destino do fluxo de dados correspondente. As variáveis adotadas são: y_k^l denominada seletor de nível e x_{ij}^{kl} que determina se o LSP_k está ou não roteado no enlace (i, j) para o l -ésimo nível de transmissão.

A família de restrições corresponde à limitação de capacidade de transmissão μ_{ij} dos enlaces e atraso máximo fim-a-fim de cada fluxo de dados.

A formulação matemática do PET é dada a seguir:

$$\text{PET : } z = \text{Maximize } \sum_{k=1}^K \sum_{l=1}^{l_k} \delta_k \lambda_k^l y_k^l \quad (4.1)$$

Sujeito a :

$$\sum_{l=1}^{l_k} y_k^l = 1 \quad k = 1, \dots, K \quad (4.2)$$

$$\sum_{k=1}^K \sum_{l=1}^{l_k} \lambda_k^l x_{ij}^{kl} \leq \mu_{ij} \quad \forall (i, j) \in E \quad (4.3)$$

$$\sum_{(i,j) \in E} \sum_{l=1}^{l_k} c_{ij} x_{ij}^{kl} \leq h_k \quad k = 1, \dots, K \quad (4.4)$$

$$\sum_{\{j:(i,j) \in E\}} x_{ij}^{kl} - \sum_{\{j:(j,i) \in E\}} x_{ji}^{kl} = b_i^k y_k^l \quad \forall i \in V \quad (4.5)$$

$$k = 1, \dots, K$$

$$l = 1, \dots, l_k$$

$$x_{ij}^{kl} \in \{0, 1\} \quad \forall (i, j) \in E \quad (4.6)$$

$$k = 1, \dots, K$$

$$l = 1, \dots, l_k$$

$$y_k^l \in \{0, 1\} \quad k = 1, \dots, K \quad (4.7)$$

$$l = 1, \dots, l_k$$

Onde: $b_i^k = 1$ se $i = o_k$, $b_i^k = -1$ se $i = d_k$, e $b_i^k = 0$ caso contrário. Sendo o_k o nó de origem do LSP_k e d_k o nó de destino do LSP_k .

A expressão (4.2) garante que cada LSP será configurado precisamente em um único nível de transmissão. A expressão (4.3) impõe os limites de largura de banda dos enlaces. A expressão (4.4) modela os limites máximos de atraso de cada LSP. A expressão (4.5) garante que os LSPs serão roteados em caminhos contínuos e não paralelos entre os nós de origem e destino dos fluxos de dados correspondentes. Finalmente as expressões (4.6) e (4.7) definem as restrições booleanas para as variáveis de decisão.

4.3 Solução Lagrangeana

A aplicação de algoritmos exatos para resolver o PET é inviável devido ao árduo esforço computacional necessário em cenários típicos de operação de rede.

No entanto, com o uso de algoritmos heurísticos (Michalewicz e Fogel 2000) é possível obter-se uma grande redução no tempo computacional, sem degradação significativa da qualidade de solução. Entenda-se aqui por solução de qualidade, valores das variáveis de decisão que garantem um valor ótimo, ou próximo do ótimo, para a função objetivo.

Uma alternativa às soluções exatas foi proposta em trabalho anterior (Dias et al. 2003a) onde foi desenvolvido um procedimento de relaxação Lagrangeana (Wolsey 1998) (Nemhauser e Wolsey 1988) que utiliza uma heurística para converter soluções duais em soluções primais viáveis. Esta solução foi denominada Relaxação Lagrangeana com Heurísticas (*RLH*).

O procedimento *RLH* possui as seguintes características:

1. Uso de dois multiplicadores Lagrangeanos para relaxar respectivamente as famílias de restrições (4.3) e (4.4), similares às restrições encontradas em problemas do tipo *problema da mochila* e correspondentes às restrições de capacidade dos enlaces e atraso máximo fim-a-fim dos fluxos de dados. O resultado desta relaxação é a conversão do problema PET em K sub-problemas de *caminhos mínimos* desacoplados;
2. aplicação do algoritmo subgradiente para obter uma solução aproximada do problema Lagrangeano dual e
3. finalmente, pós-processamento desta solução candidata, obtida no passo anterior, com uma heurística, com a finalidade de verificar e recuperar a viabilidade das soluções para o PET.

No final do processo, as soluções obtidas definem as rotas e as larguras de banda que devem ser configuradas para os LSPs, empregando-se a tecnologia MPLS.

A seguir é descrito com detalhes o procedimento *RLH*.

Relaxando as famílias de restrições (4.3) e (4.4) com os multiplicadores Lagrangeanos $v = [v_{ij} : (i, j) \in E]$ e $w = [w_k : k = 1, \dots, K]$, com $v \geq 0$ e $w \geq 0$, onde v atua como penalidade para a violação de largura de banda dos enlaces e w , como penalidade para a violação de atraso máximo fim-a-fim dos fluxos de dados, obtém-se o seguinte problema de relaxação Lagrangeana:

$$\text{PRL}(v, w) : z(v, w) = \text{Max} \sum_{k=1}^K \sum_{l=1}^{l_k} \delta_k \lambda_k^l y_k^l + \sum_{(i,j) \in E} v_{ij} \left(\mu_{ij} - \sum_{k=1}^K \sum_{l=1}^{l_k} \lambda_k^l x_{ij}^{kl} \right) + \sum_{k=1}^K w_k \left(h_k - \sum_{(i,j) \in E} \sum_{l=1}^{l_k} c_{ij} x_{ij}^{kl} \right) \quad (4.8)$$

Sujeito a :

$$\sum_{l=1}^{l_k} y_k^l = 1, \quad k = 1, \dots, K \quad (4.9)$$

$$\sum_{\{j:(i,j) \in E\}} x_{ij}^{kl} - \sum_{\{j:(j,i) \in E\}} x_{ji}^{kl} = b_i^k y_k^l, \quad k = 1, \dots, K, \quad l = 1, \dots, l_k, \quad \forall i \in V \quad (4.10)$$

$$x_{ij}^{kl} \in \{0, 1\}, \quad \forall (i, j) \in E, \quad k = 1, \dots, K, \quad l = 1, \dots, l_k \quad (4.11)$$

$$y_k^l \in \{0, 1\}, \quad k = 1, \dots, K, \quad l = 1, \dots, l_k \quad (4.12)$$

Onde: $b_i^k = 1$ se $i = o_k$, $b_i^k = -1$ se $i = d_k$, e $b_i^k = 0$ caso contrário.

É possível identificar pela formulação acima que as variáveis dos diferentes LSPs são desacopladas. O acoplamento entre elas era produzido pelas restrições, que aparecem agora na função objetivo. Desta forma, podemos resolver de forma independente e paralela os K subproblemas de localização de caminhos mínimos para cada LSP.

Mais especificamente, fixando-se um $y_k^l = 1$ para um dado nível de transmissão l e $y_k^l = 0$ para os demais, de um determinado LSP k , o problema de encontrar uma rota para o mesmo é reduzido a uma instância de solução de um problema de caminhos mínimos.

Pela repetição deste problema para cada nível de transmissão de cada LSP o PRL pode ser resolvido em tempo $O(KL(n \log n + m))$ onde $L = \text{Max}\{l_k : k = 1, \dots, K\}$, n o número de nós e m o número de enlaces da rede. Para tanto, pode ser usado o algoritmo de Dijkstra e uma sequência de Fibonacci como fila de prioridades, para resolver o problema de caminhos mínimos. O algoritmo de Dijkstra pode ser aplicado porque os coeficientes das variáveis x_{ij}^{kl} não são negativos (Cormen et al. 1990). Note que $\max -f(x) \equiv \text{minf}(x)$.

Considerando-se que a nova função objetivo relaxada — $z(v, w)$ estabelece um limite superior, podemos obter um problema Lagrangeano dual (Bertsekas 1995), do tipo :

$$\text{LD} : z_{LD} = \text{Minimize } z(v, w) \\ v, w \geq 0$$

Caso o problema primal fosse resolvido empregando-se uma relaxação linear, poderiam ser obtidos valores de limites inferiores para o PET (Wolsey 1998). Sendo assim, o valor de z_{LD} deve ser maior que o limite inferior obtido pela relaxação linear de PET, isto é, o problema obtido se as variáveis x_{ij}^{kl} e y_k^l pudessem assumir valores não inteiros, fazendo com que o problema Lagrangeano dual produza tipicamente soluções que compartilham as características de uma solução ótima.

Para solução deste problema Lagrangeano dual foi usado o algoritmo subgradiente (Wolsey 1998), (Nemhauser e Wolsey 1988) e uma heurística que recupera a viabilidade do problema primal. Estes dois algoritmos serão detalhados nas sub-seções a seguir.

4.3.1 Algoritmo subgradiente

O problema LD é convexo porém não diferenciável, isto impossibilita a aplicação de procedimentos de solução eficientes baseados em gradientes, como o algoritmo de Newton (Wolsey 1998). Isto significa que LD nem sempre tem um único subgradiente para cada ponto viável, possuindo tipicamente um número infinito de gradientes em cada ponto de descontinuidade.

A fim de resolver o problema Lagrangeano dual foi aplicado o algoritmo subgradiente, que pode ser visto como uma adaptação do algoritmo de passos decrescentes para funções não diferenciáveis.

De forma simples, o algoritmo subgradiente usa os multiplicadores Lagrangeanos v_{ij} e w_k como fatores de penalidade, a fim de desencorajar a seleção de caminhos que violem as restrições.

Desta forma, se a restrição de largura de banda de um enlace (i, j) não é violada, então seu multiplicador Lagrangeano v_{ij} é decrementado de um valor que depende do valor de seu subgradiente, ou então o valor do multiplicador é incrementado, se a restrição for violada.

Um processo similar é executado para os multiplicadores w_k : se o valor da restrição de atraso máximo fim-a-fim do fluxo de dados associado ao LSP k não é violada, então o valor de w_k é reduzido de acordo com o valor de seu subgradiente. Caso contrário, ele é incrementado.

Várias iterações são executadas até a convergência ser obtida. No final do processo o algoritmo subgradiente produz um limite superior z_{\min} para z , o qual corresponde a uma solução candidata (x_{\min}, y_{\min}) .

Se esta tentativa de solução não viola nenhuma restrição, então ela é uma solução ótima de PET.

A complexidade de solução do algoritmo subgradiente é $O(IKL(n \log n + m))$, onde $L = \text{Max}\{l_k : k = 1, \dots, K\}$ e I é o número de iterações do algoritmo subgradiente.

A seguir é apresentado o algoritmo subgradiente empregado.

Algoritmo subgradiente

Seja $t = 0$ o numero de iterações

Seja I o numero maximo de iterações

Seja $\epsilon > 0$ uma constante pequena e positiva

Seja $\delta^t > 0$ o passo inicial de decremento

Seja $v^t \in \mathbb{R}_+^m$ e $w^t \in \mathbb{R}_+^K$ multiplicadores Lagrangeanos arbitrarios

Seja $z_{min} = \infty$ o menor limite superior para z

Para $t = 1$ ate I

Resolva o PRL (v^t, w^t) obtendo uma solução tentativa: $\Psi^t = \{x_{ij}^{kl}, y_k^l\}$

Se $z(v^t, w^t) < z_{min}$ **então:** $z_{min} \leftarrow z(v^t, w^t)$ e $\Psi_{min} \leftarrow \Psi^t$

Calcular π^t um subgradiente para v^t , computado de acordo com:

$$\pi_{ij}^t \leftarrow \left(\mu_{ij} - \sum_{k=1}^K \sum_{l=1}^{l_k} \lambda_k^l x_{ij}^{kl} \right) \quad \forall (i, j) \in E$$

calcular θ^t um subgradiente para w^t , computado de acordo com:

$$\theta_k^t \leftarrow \left(h_k - \sum_{(i,j) \in E} \sum_{l=1}^{l_k} c_{ij} x_{ij}^{kl} \right) \quad k = 1, \dots, K$$

Seja v^{t+1} o proximo multiplicador Lagrangeano

para as restrições de largura de banda:

$$v_{ij}^{t+1} \leftarrow \max\{0, v_{ij}^t + \delta^t \pi_{ij}^t\} \quad \text{for all } (i, j) \in E$$

Seja w^{t+1} o proximo multiplicador Lagrangeano

para as restrições de delay maximo:

$$w_k^{t+1} \leftarrow \max\{0, w_k^t + \delta^t \theta_k^t\} \quad \text{para } k = 1, \dots, K$$

Decremente o valor de δ^t para obter δ^{t+1}

Se $\delta^{t+1} < \epsilon$ então pare

$$t \leftarrow t + 1$$

FimPara

4.3.2 Heurística de recuperação de viabilidade primal (HRV)

Mesmo que o algoritmo subgradiente possa encontrar uma solução (x_{LD}, y_{LD}) para LD, não há nenhuma garantia que a mesma seja uma solução viável para o problema PET, devido a uma folga de aproximação entre as soluções primais e duais.

Sendo assim, foi desenvolvida uma heurística que visa verificar e recuperar a viabilidade da solução para o problema primal PET. Esta heurística utiliza uma solução $\Psi_{LD} = (x_{LD}, y_{LD})$ como ponto de partida para obter uma nova solução $\Psi_H = (x_H, y_H)$ que não viole as restrições.

A idéia básica desta heurística é ordenar os fluxos de dados (LSPs) em ordem decrescente do produto dos respectivos parâmetros de prioridade e taxa de transmissão e utilizar um procedimento simples de alocação de recursos na rede, seguindo e verificando a solução apontada pelo procedimento de Relaxação Lagrangeana. Se, à medida que os fluxos são alocados em LSPs na rede, a capacidade do enlace for violada, o nível de transmissão é reduzido. A seleção das rotas a serem configuradas nos LSPs, que encaminharão os fluxos, são obtidas pela execução do algoritmo de Dijkstra.

Esta heurística foi denominada de *Heurística de Verificação e Recuperação de Viabilidade (HRV)* cujos passos são descritos a seguir.

Heurística de Verificação e Recuperação de Viabilidade (HRV)

Seja $\Psi_H = \{x_{ij}^{kl}, y_k^l\}$ uma solução tentativa inicial para o PET

onde $x_{ij}^{kl} = 0$ e $y_k^l = 0$ para todo $k = 1, \dots, K, (i, j) \in E, l=1, \dots, l_k$

Seja $g = (g_1, \dots, g_K)$ uma permutação de $\{1, \dots, K\}$

tal que $\delta_{g_k} \lambda_{g_k}^{l_{g_k}} \geq \delta_{g_{k+1}} \lambda_{g_{k+1}}^{l_{g_{k+1}}}$ para $k = 1, \dots, K - 1$

Seja $t = 1$ o número de iterações

Seja $G^t = G$ uma rede residual (a rede com a capacidade de transmissão remanescente)

Para $t = 1, \dots, K$ **faça**

Seja $k \leftarrow g_t$

Seja p_k o caminho para o k -ésimo LSP dado por Ψ_{LD}

Se p_k pode ser roteado em G^t sem violação de restrições **então**:

aloque os recursos ao longo de p_k ; atualize Ψ_H ; reduza a capacidade de G^t de λ_k^l

unidades ao longo de p_k para obter G^{t+1} , sendo $y_k^l = 1$ a solução de Ψ_{LD} ;

Senão

$G^{t+1} \leftarrow G^t;$

FimSe

$t \leftarrow t + 1;$

FimPara

$G^1 \leftarrow G^K;$

Para $t = 1, \dots, K$

Seja $k \leftarrow g_t$

Se o k -ésimo LSP não pode ser alocado, então:

Para $l = l_k, \dots, 1$ **faça**

Use o algoritmo Dijkstra para encontrar um caminho alternativo p_k em G^t ;

Se $\mu_{ij}^t < \lambda_k^l$ então:

O custo do arco $(i, j) = c_{ij}^t = \infty$;

caso contrario $c_{ij}^t = c_{ij}$;

Se um caminho p_k for encontrado e $\sum_{(i,j) \in p_k} c_{ij}^t \leq h_k$ então:

aloque os recursos ao longo de p_k ;

atualize Ψ_H ; reduza a capacidade de G^t de λ_k^l unidades ao longo de p_k obtendo G^{t+1} ;

FimSe

FimPara

Senão

$G^{t+1} \leftarrow G^t;$

FimSe

$t \leftarrow t + 1;$

FimPara

O custo computacional da execução da HRV é $O(KL(n \log n + m))$. O custo computacional total para obter a solução final do PET, usando o algoritmo subgradiente para produzir Ψ_{LD} com a heurística HRV para verificar e recuperar a viabilidade da solução primal Ψ_H é por sua vez $O(IKL(n \log n + m))$.

4.4 Solução heurística pura

Apesar do procedimento *RLH* produzir soluções com complexidade polinomial, o tempo computacional e a qualidade das soluções são diretamente relacionados com seus parâmetros de configuração, tais como o passo de decremento do algoritmo subgradiente δ_t , a constante ϵ e o número de iterações I do algoritmo subgradiente. Os experimentos apresentados na seção seguinte demonstram que o tempo computacional pode ser excessivo para alguns casos representativos da operação de uma rede de computadores, sobretudo se a heurística *RLH* for empregada para solução em um problema de ET dinâmico.

Um procedimento de otimização foi desenvolvido, denominado Procedimento Heurístico Puro (PHP), que não considera *a priori* os procedimentos de relaxação Lagrangeana. Apesar do *PHP* encontrar uma solução viável para o PET, não existe garantia de sua otimalidade. A ideia básica desta heurística é a mesma da *HRV*, somente não utiliza *a priori* um procedimento de Relaxação Lagrangeana como ponto de partida, reduzindo o tempo de execução.

Apesar disto, uma série de experimentos demonstrou que, com os parâmetros inicialmente estabelecidos para o algoritmos subgradiente (a saber δ_t , ϵ e I) o desempenho computacional e também a qualidade de solução de *PHP* foi superior à obtida pela abordagem *RLH* em todas os casos experimentadas.

Este fato evidencia que o uso do *PHP* é uma alternativa eficiente e eficaz para a solução de tarefas de ET. O algoritmo do *PHP* é mostrado abaixo.

Procedimento Heurístico Puro (PHP)

Seja $g = (g_1, \dots, g_K)$ uma permutação de $\{1, \dots, K\}$ tal que

$$\delta_{g_k} \lambda_{g_k}^{l_{g_k}} \geq \delta_{g_{k+1}} \lambda_{g_{k+1}}^{l_{g_{k+1}}} \text{ para } k = 1, \dots, K-1$$

Seja $t = 1$ o numero de iterações

Seja $G^t = G$ uma rede residual, ou seja, uma rede com

a capacidade de transmissão remanescente

Seja $\Psi_H = \{x_{ij}^{kl}, y_k^l\}$ uma solução inicial para PET

onde $x_{ij}^{kl} = 0$ e $y_k^l = 0$ para todo $k = 1, \dots, K$, $(i, j) \in E$, $l=1, \dots, l_k$

Para $t = 1, \dots, K$

Seja $k \leftarrow g_t$

Para $l = l_k, \dots, 1$

Use o algoritmo de Dijkstra para encontrar um caminho

p_k em G^t para o k -ésimo LSP, tal que

$c_{ij}^t = \infty$ se $\mu_{ij}^t < \lambda_k^l$ e, caso contrário, $c_{ij}^t = c_{ij}$

Se um caminho p_k foi encontrado e $\sum_{(i,j) \in p_k} c_{ij}^t \leq h_k$ **então**:

aloque os recursos ao longo de p_k e atualize Ψ_H

reduza a capacidade de G^t de λ_k^l unidades ao longo de

p_k obtendo G^{t+1}

FimSe

FimPara

$t \leftarrow t + 1$

FimPara

A complexidade de tempo do procedimento heurístico puro (*PHP*) é $O(KL(n \log n + m))$.

4.5 Experimentos e resultados

De acordo com (Girish et al. 2000) alguns dos principais critérios para avaliação de desempenho de algoritmos são: i) *grau de otimalidade*, que mede a capacidade do algoritmo em encontrar uma solução de alta qualidade (com valores iguais ou próximos de valores ótimos); (ii) *tempo de execução*, que se refere ao tempo necessário para obter-se a solução.

A estratégia de avaliação deste procedimentos compreenderá as seguintes análises:

- Avaliação de desempenho dos algoritmos: onde serão contrastados o grau de otimalidade e os tempos de execução para obtenção de soluções sub-ótimas para o PET, pelo uso dos procedimentos heurísticos *PHP* e *RLH*, com as soluções exatas, obtidas por um pacote comercial de alto desempenho para solução de problemas de programação linear inteira — o Xpress-MP (Das 2002);
- Análise do controle de admissão: onde são analisados o número de rejeições de fluxos de dados quando o PET é resolvido com o emprego do *PHP*.

Duas topologias de rede foram analisadas. A primeira topologia é muito similar às adotadas em (Rabbat 2000a), (Ricciato e Angelo 2002), (Kodialam e Lakshman 2000) e (Banerjee e Sidhu 2002). Esta topologia é apresentada na figura 4.1. A segunda topologia, com maior dimensão e complexidade, atingindo 50 nós e 200 enlacs, pode ser vista na figura 4.2. Ela

foi gerada com o gerador aleatório de topologias de rede chamado BRITE (Boston University Internet Topology Generator) (Medina et al. 2001).

O objetivo do uso da segunda topologia foi avaliar o comportamento dos procedimentos heurísticos propostos para uma topologia extremamente aleatória e complexa, a fim de verificar a eficiência e o grau de otimalidade das soluções (auferido pelo valor da função objetivo), sobretudo para as soluções produzidos pelo *PHP*. Os experimentos tendem demonstrar que mesmo em condições de grande carga de trabalho e complexidade da topologia, o *PHP* produz bons resultados, superiores aos produzidos pelos procedimentos Lagrangeanos (*RLH*).

Os algoritmos foram implementados em C++, executados em um computador Pentium IV de 1.2 GHz e 256 Mbytes de RAM e sistema operacional Linux.

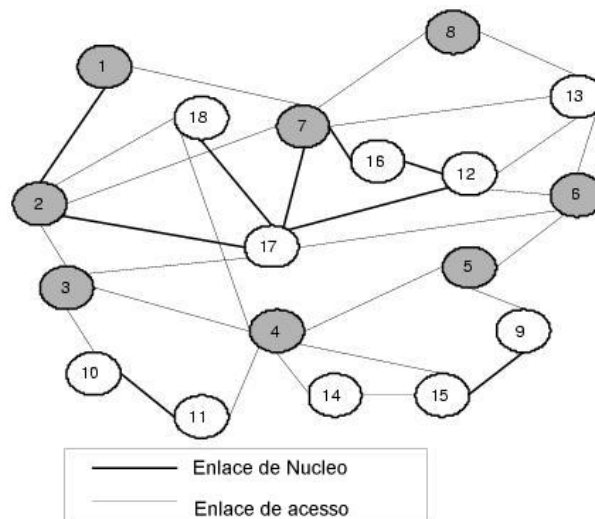


Figura 4.1: Primeira topologia de rede

4.5.1 Experimentos

A fim de avaliar o desempenho dos algoritmos propostos, foram gerados dois tipos básicos de fluxos de dados, ambos do tipo CBR (Constant Bit Rate). O nível mais baixo é configurado obrigatoriamente com nível igual a zero Kbps, o que corresponde a rejeição do fluxo correspondente na rede. Os dois tipos de fluxos de dados são: (i) *fluxos de baixa prioridade*: com taxa de transmissão variando aleatoriamente entre 20 e 150 Kbps e restrições de atraso de transmissão fim-a-fim variando entre 100 and 150 ms; e (ii) *fluxos de alta prioridade*: com taxa de transmissão variando entre 380 e 870 Kbps, com restrições de atraso fim-a-fim variando entre 30 e 60 ms.

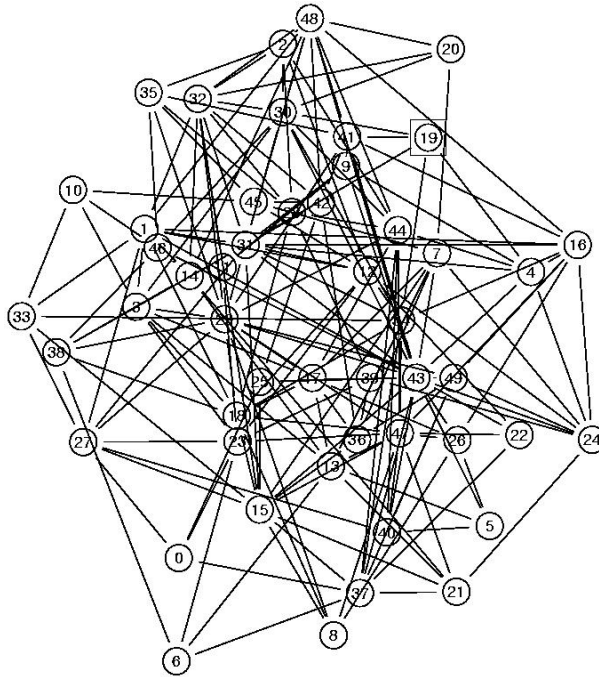


Figura 4.2: Segunda topologia de rede

Cada fluxo pode assumir, nestes experimentos, 5 possíveis níveis de transmissão, variando de 0 (zero) até um valor máximo desejado dentro das faixas correspondentes. Para a topologia apresentada na figura 4.1 foram gerados até 500 fluxos de dados. Para a topologia apresentada na figura 4.2 foram gerados até 20.000 fluxos de dados. Para ambos cenários, do total de fluxos injetados na rede, 1,5 % são fluxos de alta prioridade.

Com estes experimentos procurou-se simular um cenário aproximado de uma operadora de telecomunicações que provê serviços de transmissão de vídeo, com alta prioridade no acesso aos recursos da rede, juntamente com serviços básicos (melhor esforço), de baixa prioridade. Os experimentos foram executados com somente duas classes de serviço, porém o modelo permite o emprego de várias classes.

4.5.2 Topologia 1

Os resultados obtidos nesta sub-seção são relacionados à topologia 1. Como apresentado anteriormente, foram empregadas duas abordagens de solução para PET: *RLH* que emprega procedimentos de relaxação Lagrangeana e o *PHP*, que é um procedimento heurístico puro.

4.5.2.1 Avaliação de desempenho do PHP e RLH

Nos experimentos, as soluções exatas, computadas pelo Xpress-MP, foram obtidas com cargas de trabalho de até 200 fluxos de dados. Acima desta quantidade de fluxos, o tempo de execução cresce exponencialmente e nos casos de 250 e 300 fluxos, o Xpress-MP computa somente valores de limites superiores para o PET, através de relaxações lineares (Wolsey 1998).

Seguindo as estratégias de avaliação, a figura 4.3 apresenta o tempo de execução para os vários casos experimentados, usando os procedimentos *PHP*, *RLH* e soluções exatas computadas pelo Xpress-MP.

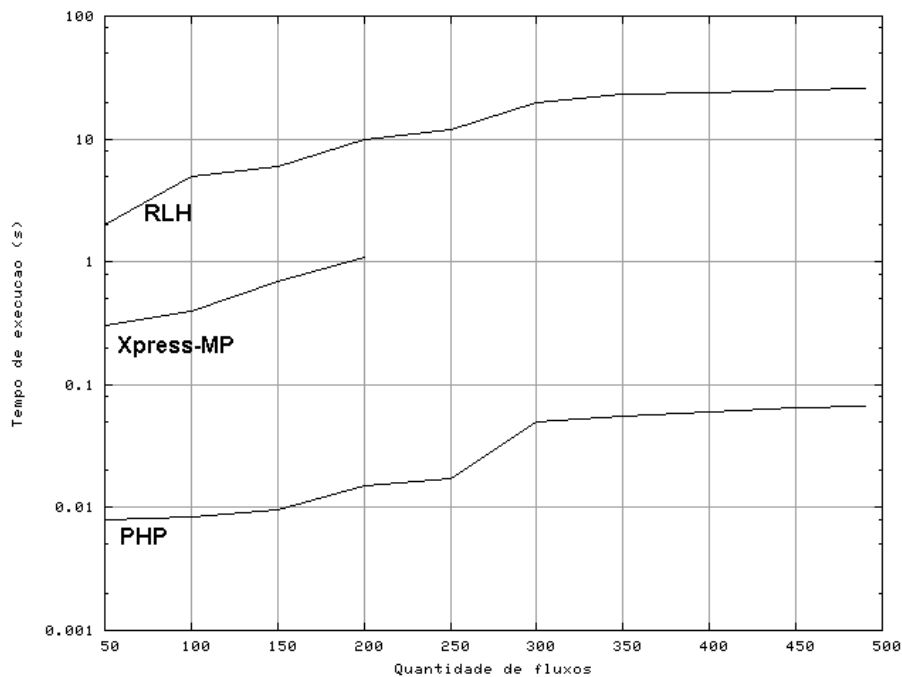


Figura 4.3: Tempo de execução do RLH, PHP e Xpress-MP para várias cargas de trabalho

Como é possível observar na figura 4.3, o tempo computacional consumido pelo *PHP* é relativamente pequeno, no pior caso consumindo menos que 0,06s para rotar 490 fluxos de dados. O *RLH* por sua vez, ultrapassa 25s para o mesmo caso. Além disso, nos casos onde é possível obter uma solução para o PET em tempo viável, o tempo consumido pelo Xpress-MP também é maior do que o obtido com o *PHP*, levando cerca de 1s para obter a solução, com uma carga de 200 fluxos de dados, enquanto o *PHP* consome em torno de 0,02s.

A figura 4.4 representa o grau de otimalidade do *PHP* e do *RLH* pela comparação dos valores de função objetivo obtidos pelo *PHP* e *RLH* com os valores de função objetivo obtidos pelo uso do Xpress-MP na solução do PET.

A partir da observação da figura 4.4 pode-se concluir que as soluções produzidas por *PHP*

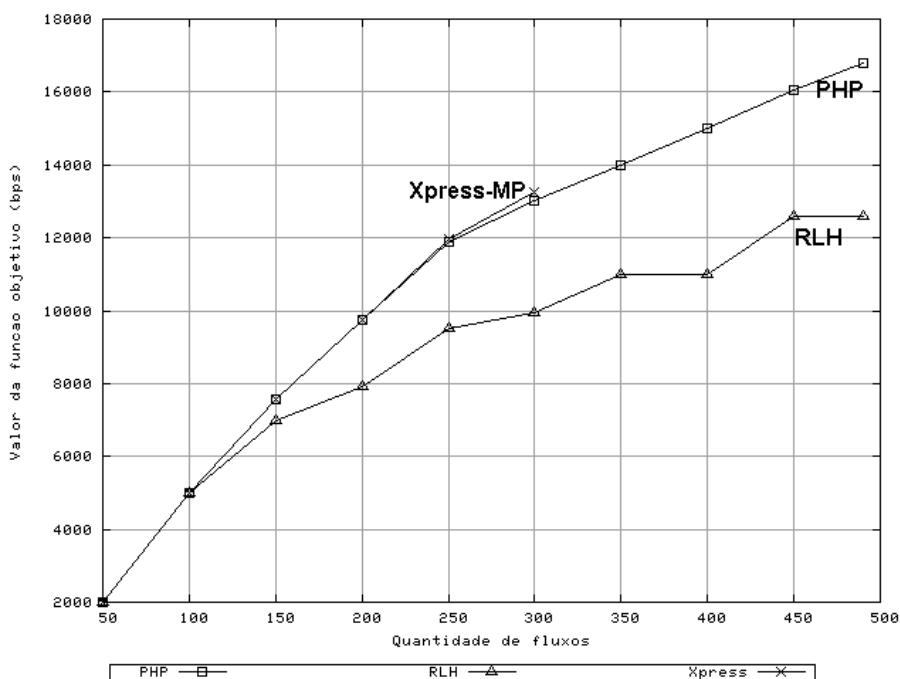


Figura 4.4: Grau de otimalidade do *RLH* e *PHP*

são de alta qualidade. Nota-se que *PHP* encontra soluções ótimas ou próximas das ótimas ou dos limites superiores nas instâncias correspondentes às computadas pelo *Xpress-MP*. O *RLH* por sua vez produz soluções de qualidade inferior.

Para obter soluções de qualidade superior, o *RLH* deveria ter seus parâmetros alterados, como o passo de decremento e principalmente o número de iterações. Invariavelmente estas mudanças produzem um incremento no tempo de execução do *RLH*.

4.5.2.2 Controle de admissão do *PHP*

Para garantir a QoS dos fluxos de alta prioridade o *PHP* incorpora uma política de controle de admissão dos fluxos de dados. Nesta política, os fluxos de alta prioridade têm acesso privilegiado aos recursos da rede. Na eminência de um congestionamento, os fluxos de baixa prioridade podem ter a taxa de transmissão reduzida até zero, condição de rejeição do fluxo.

A figura 4.5 mostra o percentual de fluxos de baixa prioridade que foram rejeitados na rede pelo uso do *PHP*, nos vários casos experimentados. Nota-se que todos os fluxos de alta prioridade foram admitidos com vazão máxima desejada.

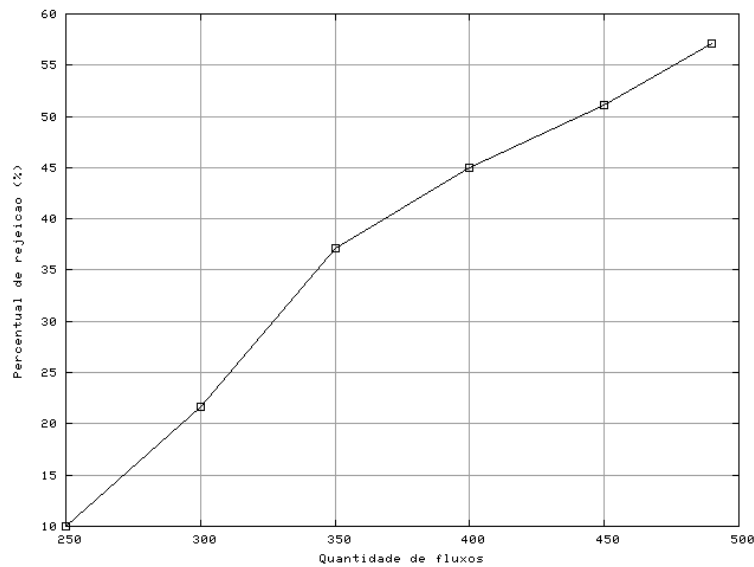


Figura 4.5: Percentual de fluxos rejeitados na rede pelo emprego do *PHP*

Com o objetivo de demonstrar que o carregamento da rede é elevado, foi analisada a taxa de perdas de pacotes quando são utilizados protocolos de roteamento do tipo Interior gateway Protocol (IGP) convencionais, como o Routing Information Protocol (RIP) e o Open Short Path First (OSPF). Para tanto, foi empregada a ferramenta de simulação de redes Network Simulator (NS) (VINT 2003).

A figura 4.6 mostra que a taxa de perda global de pacotes (diferença percentual entre o número total de pacotes transmitidos e o número total de pacotes recebidos) quando se emprega protocolos IGP é alta. Observa-se que acima de 250 fluxos a taxa de perda é muito elevada, nestes casos, superando 30 %, tanto para o RIP como para o OSPF, evidenciando uma operação com grande congestionamento.

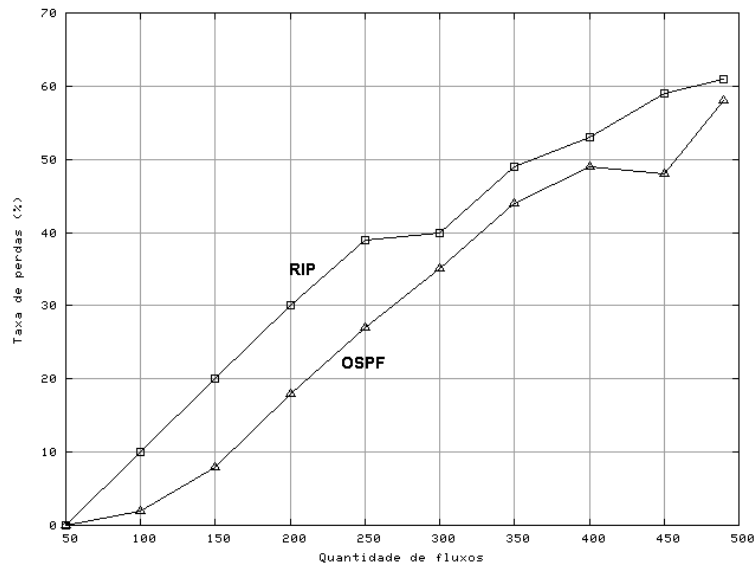


Figura 4.6: Taxa de perdas de pacotes global quando empregados os protocolos RIP e OSPF

Os resultados dos experimentos com a topologia 1, demonstram que o procedimento *PHP* tem boa eficiência e ótima qualidade de solução para um cenário significativo. A solução de Relaxação Lagrangeana tende a produzir resultados com maior custo computacional. A qualidade do procedimento *RLH* poderia ser melhorada se os parâmetros da heurística de recuperação de viabilidade e algoritmo subgradiente forem alterados.

Visando-se verificar se o comportamento do procedimento *PHP* segue a tendência em outros casos, buscou-se empregar uma topologia de rede mais complexa, gerada de forma aleatória, com um carga de trabalho extremamente elevada. Os resultados desta segunda topologia são apresentados na seção seguinte.

4.5.3 Topologia 2

Os resultados apresentados nesta sub-seção estão relacionados à topologia apresentada na figura 4.2, que foi sinteticamente gerada pelo gerador aleatório de topologias de rede chamado BRITE.

Para esta topologia foram gerados até 20.000 fluxos de dados de acordo com a estratégia de geração de fluxos. O propósito destes experimentos foi verificar que o método *PHP* mantém o mesmo comportamento em uma topologia complexa, aleatória e correspondente a uma elevada carga de trabalho.

A figura 4.7 mostra o tempo de execução dos procedimentos *RLH* e *PHP* para a topologia 2, para as várias cargas de trabalho experimentadas. Ela confirma que o *PHP* é substancialmente mais rápido que o *RLH*, que chega a levar 11.000s (acima de três horas) para computar

as rotas para os 20.000 fluxos, enquanto o *PHP* leva menos de 10s.

A figura 4.8 mostra os valores das funções objetivo obtidos com o uso do *PHP* e *RLH*, evidenciando que a qualidade de solução do *PHP* é superior.

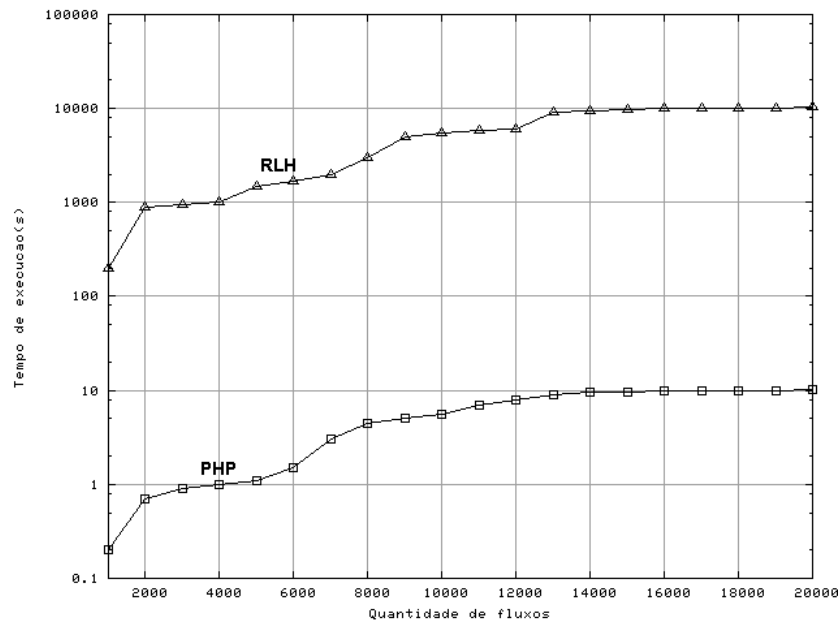


Figura 4.7: Tempo computacional consumido pelo *RLH* e *PHP* para a topologia 2

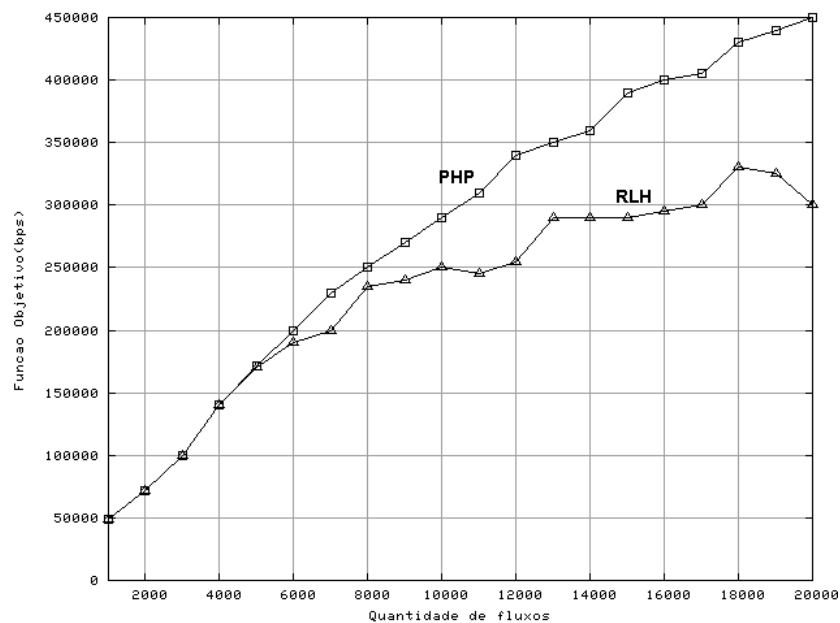


Figura 4.8: Valores da função objetivo computados pelo *RLH* e *PHP*

Como a topologia é complexa (grande número de nós e enlaces) e a carga de trabalho elevada, não é possível resolver o problema com o emprego do Xpress-MP. Não foi estimado, neste caso, o grau de otimalidade das soluções obtidas com o uso do *PHP* e *RLH*. O intuito foi mostrar que o uso de um procedimento heurístico puro é muito rápido. O grau de otimali-

dade do *RLH* poderia ser melhorado se os parâmetros do algoritmo *HRV* fossem ajustados, como discutido anteriormente. No entanto, estas alterações implicarão em substancial aumento do tempo de execução que já é muito elevado (cerca de 10.000 vezes maior que o consumido pelo *PHP* para este experimento).

A principal conclusão dos resultados para esta segunda topologia é que o *PHP* segue o comportamento anterior, tendo um bom desempenho, mesmo em condições de operação mais severas, indicando que o mesmo pode ser empregado em cenários de operação de rede significativos. Como seu tempo de execução é polinomial, o uso de um hardware com grande poder de processamento e/ou a adaptação do procedimento *PHP* para operação com processamento paralelo, torna-se uma opção viável para sua utilização em tempo real.

4.6 Conclusões do capítulo

Neste capítulo foi apresentado um problema típico de Engenharia de Tráfego em redes IP sobre tecnologia MPLS e duas abordagens de solução. O problema de ET consiste na maximização de vazão dos fluxos de dados que são injetados em um domínio MPLS, sujeito a restrições de capacidade dos enlaces e limitações de atraso máximo fim-a-fim deste fluxos de dados, de acordo com uma política controle de admissão que proporciona diferenciação de serviços.

Foi proposta uma abordagem de solução baseada em um procedimento de Relaxação Lagrangeana complementado por uma heurística, que apesar de encontrar boas soluções em tempo polinomial, apresentou-se lenta para os casos estudados. Uma segunda abordagem de solução, por sua vez, puramente heurística, proporcionou ótimos resultados e com eficiência satisfatória.

Uma das contribuições diretas do capítulo foi o desenvolvimento de algoritmos que permitam o desenvolvimento de ferramentas de planejamento de capacidade de redes. A partir desta ferramenta seria possível prever os limites de operação da rede, sendo útil no adequado dimensionamento das capacidades dos enlaces e previsão da capacidade de carga de trabalho, de forma rápida e simples.

Apesar do cenário proposto para os experimentos contemplar somente duas classes de serviços, o *PHP* permite operação com um maior número de classes, bastando apenas configurar diferentes níveis de prioridade para os fluxos.

A discretização da vazão foi feita com cinco níveis nos experimentos realizados, no entanto, o número de níveis pode ser selecionado de acordo com a granularidade desejada. O aumento

do número de níveis aumenta o tempo de execução do *PHP*, porém de forma linear, não comprometendo significativamente o desempenho do *PHP*.

Tendo em vista o ótimo desempenho computacional do *PHP*, o capítulo seguinte mostra como este procedimento pode ser usado em cenários de Engenharia de Tráfego Dinâmicos, melhorando a QoS dos fluxos de dados de alta prioridade, com a rede operando em tempo-real.

Capítulo 5

Uma abordagem para solução de um problema de engenharia de tráfego dinâmico (PETDIN)

No capítulo anterior foi demonstrado que o Procedimento Heurístico Puro (*PHP*) é eficiente e tem qualidade de solução.

Como o tempo de execução deste procedimento é polinomial, sendo proporcional ao tempo de execução do algoritmo de Dijkstra, o presente capítulo visa demonstrar a viabilidade do emprego de um procedimento de sequencialização do *PHP* para solução de problemas de engenharia de tráfego dinâmicos, onde requisições de admissão de fluxos de dados são submetidas à rede ao longo do tempo.

Neste capítulo serão discutidos os seguintes assuntos: (i) definição do problema de engenharia de tráfego dinâmico; (ii) proposta de solução; (iii) implementação da solução; (iv) experimentos e resultados; (v) trabalhos relacionados; (vi) conclusões do capítulo.

5.1 Definição do problema

Considerando-se uma rede delimitada por um domínio administrativo, representada por um grafo $G = (V, E)$, onde $|V| = n$ é o número de nós; $|E| = m$, é o número de enlaces. Considerando-se a_i como sendo o número de nós de acesso (origem ou destino dos fluxos de dados), com $i = 1, \dots, n_a$, onde n_a é o número de nós de acesso. Neste cenário as requisições

de serviço são submetidas continuamente ao longo do tempo, o que configura um problema de engenharia de tráfego dinâmico.

Em suma, o problema de engenharia de tráfego dinâmico consiste então em atender de forma dinâmica novas requisições de serviço submetidas a um domínio administrativo de rede.

A figura 5.1 mostra em forma de linha do tempo como se processa este problema de engenharia de tráfego dinâmico.

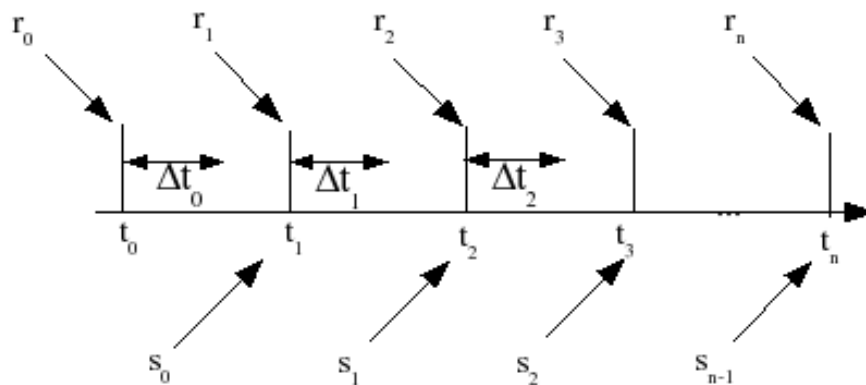


Figura 5.1: Linha do tempo de um problema de ET dinâmico

Considerando-se que os atributos topológicos e as demandas dos fluxos são conhecidas, temos:

1. Num instante t_0 , r_0 requisições de serviço aguardam admissão na rede. A partir das informações lógicas das r_0 requisições de serviço (demanda de taxa de bits, atraso máximo fim-a-fim e demais restrições de QoS), é executado um procedimento de ET;
2. No instante t_1 , após um intervalo de tempo Δt_0 , correspondente ao processamento do procedimento de ET referente ao conjunto r_0 de requisições de serviço, as configurações da rede, representadas pelo conjunto de soluções s_0 são aplicados à rede. Um conjunto de fluxos Δr_0 pode ter a admissão rejeitada após a obtenção desta solução.
3. No tempo t_1 , r_1 requisições de serviço aguardam admissão na rede. O conjunto de requisições de serviço r_1 é a soma do conjunto de requisições Δr_0 com o conjunto r'_1 que representa os fluxos que pediram admissão entre o tempo t_0 e t_1 .
4. No instante t_2 , após um intervalo de tempo Δt_1 , suficiente para o processamento do procedimento de ET referente ao conjunto r_1 , as configurações dos fluxos de dados, repre-

sentados pelo conjunto de soluções s_1 são aplicados à rede. Um conjunto Δr_1 pode ter a admissão rejeitada na rede.

5. Em t_2 , r_2 requisições de de serviço aguardam admissão na rede. O procedimento de ET se repete de forma similar aos itens anteriores ao longo do tempo.

Deve-se salientar que as requisições que não forem atendidas num determinado instante de tempo são submetidas ao procedimento de otimização num instante posterior, podendo ser retiradas da fila de admissão após um determinado intervalo de tempo (*time out*).

5.2 Proposta de solução

Como mostrado na figura 5.1 várias rodadas de otimização são executadas em pequenos intervalos de tempo. Enquanto a otimização é processada, novas requisições de admissão de fluxos de dados ficam aguardando para serem encaminhadas, após a execução de uma nova rodada.

Para resolver este problema, a solução proposta é um processo simples de sequencialização de um procedimento de otimização *PHP* que foi descrito no capítulo anterior. Esta proposta é viável devido ao ótimo desempenho computacional do *PHP*, mesmo em instâncias de operação significativas.

Esta solução para o problema de ET dinâmico, de acordo com a taxonomia utilizada para Engenharia de Tráfego (capítulo 2) é classificado como:

- Um procedimento *centralizado*, pois cabe a uma entidade centralizada executar as várias rodadas de otimização, a partir de informações lógicas (demandas de largura de banda e limites de atraso máximo fim-a-fim) de cada fluxo de dados, bem como de informações topológicas;
- Um processo *prescritivo*, pois o processo de ET encontra uma solução ótima ou sub-ótima para a distribuição do tráfego na rede e tem como saída, informações de configuração dos LSPs (rotas e larguras de banda);

5.3 Implementação da solução

Esta seção apresenta a estrutura lógica e a infra-estrutura física necessária à implementação do "Procedimento de Engenharia de Tráfego Dinâmico"(PETDIN). Serão discutidas

também estratégias de implementação e questões operacionais da solução proposta.

5.3.1 Estrutura lógica

A solução do problema de engenharia de tráfego dinâmico emprega a mesma formulação matemática do PET, adotando a mesma estratégia de diferenciação de serviços, classificando os fluxos em alta e baixa prioridade.

A solução do problema de engenharia de tráfego dinâmico (PETDIN) tem a seguinte estrutura:

- **Entrada de dados:** obtém informações sobre a topologia da rede (que pode ser obtida a partir de arquivos de configuração e/ou protocolos do tipo "link-state", como o OSPF), bem como sobre as informações lógicas (demanda de taxa de bits e restrições de atraso fim-a-fim dos fluxos de dados). As informações lógicas são fornecidas pelos dispositivos de borda da rede (LERs).
- **Componente de otimização:** utiliza o *PHP* para configuração das rotas e larguras de bandas dos LSPs;
- **Componente de monitoramento e geração de logs:** verifica se os fluxos de dados levam mais do que um tempo limite (time-out) para admissão no domínio; registra em arquivos informações para análise (logs) como: número de rejeições, distribuição de renivelamentos (alterações de configurações nas larguras de banda dos LSPs); tempo de espera de admissão por fluxo; número de re-roteamentos ao longo do tempo; número de exclusões de fluxos de dados (quando o time-out é atingido os fluxos são excluídos da lista de pendência de admissão).

A figura 5.2 mostra a estrutura lógica da solução do PETDIN

5.3.2 Infra-estrutura física

A figura 5.3 mostra um diagrama representando uma proposta de infra-estrutura para execução do procedimento *PHP* para solução de um problema dinâmico.

Como se pode observar da figura 5.3, o modelo proposto é similar ao Common Open Police Service (COPS) (Durham 2000) do IETF, empregado no gerenciamento de redes baseado

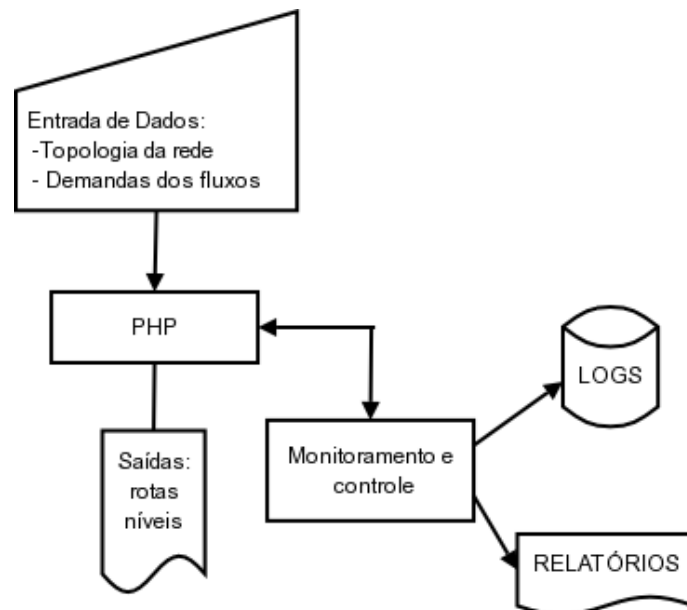


Figura 5.2: Estrutura lógica da solução do PETDIN

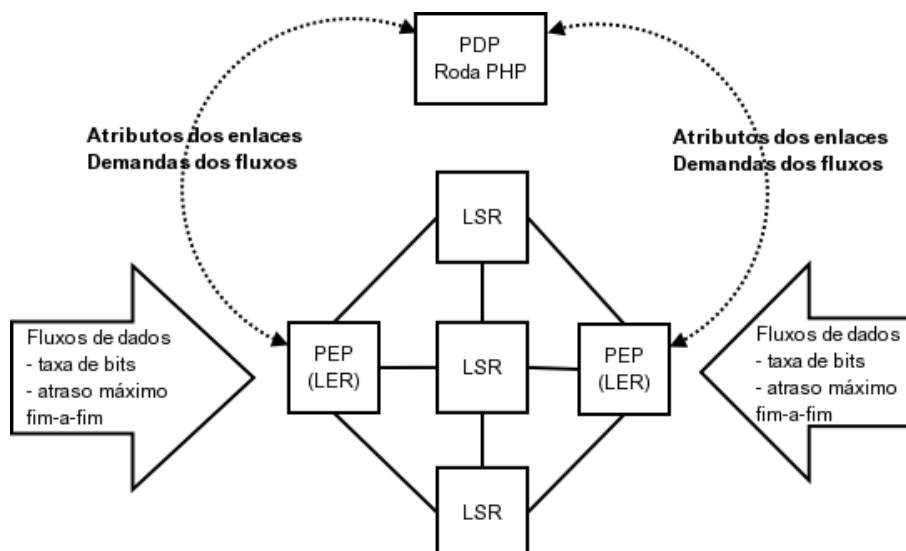


Figura 5.3: Infra-estrutura para o procedimento dinâmico de engenharia de tráfego PETDIN

em políticas. Neste modelo, descrito no anexo A, a entidade centralizada de processamento do *PHP* representa o PDP (Police Decision Point) do COPS. Dentro do modelo COPS, o PDP é uma entidade que, a partir de informações de gerenciamento baseado em políticas, expressas em notação de alto nível, define as ações a serem implementadas na rede. Nos dispositivos de borda do domínio (LERs) devem estar instalados agentes similares aos PEPs (Police Enforcement Points) do COPS. Os PEPs são entidades que se encarregam de mapear as decisões de gerenciamento definidas pelos PDPs em ações de configuração dos elementos de rede. Neste cenário, cabe aos LERs repassar à entidade centralizada, as informações lógicas para o processamento do *PHP*, a partir de algum protocolo de sinalização. Desta forma, a infra-estrutura de rede para transmissão de informações de controle pode ficar separada da rede de dados.

5.3.3 Valores de prioridade de configuração e manutenção

Como o processo de ET ocorrerá de forma dinâmica, foi adotada uma estratégia que empregasse dois valores para o parâmetro de prioridade. Um valor de configuração, quando o fluxo ainda não foi admitido na rede, o qual será denominado de valor *setup* e outro valor de manutenção, quando o fluxo já foi admitido previamente na rede, o qual será denominado de valor *holding*.

Uma prioridade com valor *setup* elevado indica que o fluxo tem alta prioridade em ser admitido na rede. Uma prioridade com valor *holding* alto indica que o fluxo tem alta prioridade em manter a sua configuração (rota e largura de banda).

A idéia de utilizar dois valores para o parâmetro de prioridade, em condições de configuração inicial e de manutenção, foi inspirada no protocolo CR-LDP (Jamoussi 2002). Com a adequada configuração da razão entre estes valores é possível diminuir o efeito de re-configuração dos fluxos de dados, o que poderia provocar a instabilidade na rede ou a degradação da QoS destes fluxos ao longo do tempo de transmissão. A adequada seleção da razão entre os dois parâmetros, permite por exemplo, evitar a interrupção de um fluxo de alta prioridade já admitido na rede, em detrimento de novos fluxos de baixa prioridade. Para tanto, basta que se configure uma relação *holding/setup* elevada para os fluxos de alta prioridade em relação aos de baixa prioridade.

5.3.4 Discussão sobre questões operacionais da proposta

A solução do problema PETDIN necessita de uma plataforma de hardware eficiente, do ponto de vista de velocidade de processamento, para o processamento de um grande número de fluxos de dados simultâneos. Como apresentado no capítulo 4, com uma topologia complexa e com grande número de fluxos de dados, a exigência de processamento pode ser alta. Entretanto, a complexidade de tempo é polinomial e o tempo de execução pode ser reduzido pelo aumento da capacidade do hardware (maior poder de processamento) ou adaptando os algoritmos para processamento paralelo.

Como observado na descrição do *PHP*, a heurística desenvolvida é baseada num algoritmo de caminhos mínimos sujeitos a múltiplas restrições. A idéia básica deste procedimento é a seleção de rotas e configurações de níveis de largura de banda para os LSPs que não violem as restrições de atraso fim-a-fim dos fluxos e larguras de bandas dos enlaces. Sendo assim, esta abordagem visa a eliminação dos congestionamentos, pelo controle de admissão dos fluxos de dados. Entretanto numa rede com elevada carga de trabalho, a rejeição de alguns fluxos de dados seria inevitável.

Nesta proposta de trabalho é considerado que o sistema de gerenciamento de ET tem controle sobre a taxa de transmissão dos fluxos de dados. Do ponto de vista da implementação poderia ser considerada uma limitação importante, uma vez que um mecanismo de sinalização deveria ser estabelecido entre o dispositivo de borda e a fonte de dados, a fim de controlar sua taxa de bits. No entanto, o modelo poderia ser empregado sem nenhuma alteração na rede e nas aplicações, se as taxas de bits dos fluxos de dados não fossem alteradas de acordo com a solução do *PHP*. Neste caso, a solução poderia configurar os LSPs com larguras de banda inferiores às demandas dos fluxos de dados, implicando na existência de perdas de pacotes nas filas dos roteadores.

5.4 Estratégia de avaliação da proposta

A estratégia de avaliação da proposta foi dividida em três etapas, de forma similar ao que foi apresentado no capítulo 4:

1. **Desempenho do algoritmo:** nesta etapa, foram experimentadas instâncias com grande carga de trabalho (grande número de fluxos por rodada de otimização). Com esta etapa, foi

verificada a aplicabilidade do *PHP* para solução de problemas dinâmicos. Os parâmetros avaliados nesta etapa foram:

- **Tempo de execução ao longo do tempo de simulação:** com este parâmetro é verificado o custo de processamento para obtenção das soluções para o *PETDIN*;
- **Valor da função objetivo ao longo do tempo:** com este parâmetro se verifica o aproveitamento dos recursos empregando-se o *PETDIN* nas várias instâncias experimentadas.

2. **Controle de Admissão:** nesta etapa foi verificado o comportamento da solução proposta no que diz respeito aos parâmetros de controle de admissão executados pelo *PHP* durante a obtenção das soluções do *PETDIN*. Os parâmetros avaliados nesta etapa foram:

- **Tempo de espera de admissão dos fluxos:** este parâmetro permitiu verificar o tempo que os fluxos de dados, divididos nas classes de alta e baixa prioridade levam para ser admitidos na rede;
- **Número de rejeições de fluxos de dados ao longo do tempo:** com este parâmetro procurou-se avaliar o número médio de rejeições ao longo do tempo;
- **Número de interrupções de transmissão de fluxos de dados ao longo do tempo:** este parâmetro avalia a distribuição da frequência de interrupções de transmissão de fluxos de dados ao longo do tempo de simulação, de acordo com prioridade dos fluxos. Entende-se por interrupção, o processo no qual um determinado fluxo de dados, previamente admitido em uma rodada de otimização, teve sua transmissão interrompida numa rodada posterior;
- **Número de re-roteamentos ao longo do tempo:** este parâmetro permite avaliar a distribuição de frequência de re-roteamentos sofridas pelos fluxos de dados ao longo das várias rodadas de otimização;
- **Número de renivelamentos dos fluxos de dados:** este parâmetro permite avaliar a distribuição de frequência de renivelamentos ao longo das rodadas de otimização. Os renivelamentos podem ocorrer como consequência dos resultados das rodadas de otimização (larguras de banda a serem configuradas nos LSPs);
- **Número de exclusões de fluxos de dados:** este parâmetro mede o número de fluxos que após um tempo determinado (time out) não conseguiram admissão na rede. Este *timeout* foi adotado pois em condição de grande carregamento da rede, um fluxo de

baixa prioridade e baixa demanda de taxa de transmissão, pode não obter admissão na rede, após muitas rodadas de otimização. Assim, um fluxo de dados é excluído da fila de espera de admissão após um tempo limite.

3. **QoS dos fluxos:** nesta etapa foram avaliados os parâmetros de desempenho da rede para o encaminhamento dos fluxos baseados no roteamento *PHP* em comparação com o roteamento IGP convencional. Os parâmetros de desempenho dos fluxos de dados avaliados foram: vazão, atraso fim-a-fim, variação de atraso fim-a-fim.

5.5 Experimentos e resultados

A fim de se avaliar a proposta de solução do problema PETDIN, foi realizada uma série de experimentos numéricos e de simulações. Os procedimentos foram executados com a mesma plataforma de hardware e software descritos no capítulo 4: procedimentos implementados em C++, executados em um computador Pentium IV de 1.2 GHz e 256 Mbytes de RAM. Também foi utilizada a mesma estratégia de geração de carga de trabalho, com fluxos de alta e baixa prioridade, com as mesmas características, ou seja, mesmas faixas de variação de taxa de transmissão, mesma faixa de variação de limite de atraso máximo fim-a-fim. O percentual de fluxos de alta prioridade também foi de 1,5 %.

Nestes experimentos a taxa de transmissão dos fluxos de dados foi dividida em 7 diferentes níveis de transmissão, a fim de proporcionar uma maior granularidade na seleção dos níveis de largura de banda dos LSPs.

A avaliação dos parâmetros de desempenho de rede como vazão, atraso de transmissão, taxa de perda de pacotes foi obtida com o emprego do simulador de rede Network Simulator (VINT 2003). Para implementar o roteamento explícito foi empregado o protocolo CR-LDP, disponível no módulo de extensão do NS denominado MPLS Network Simulator (MNS) (Ahn e Chun 2002). A topologia adotada na validação é a mesma empregada no capítulo 4 (topologia 1).

5.5.1 Limitações da plataforma de validação

Algumas limitações foram encontradas na realização dos experimentos:

- **Limitações de memória:** o *PHP* é um procedimento extremamente rápido e eficiente,

consumindo pouca memória. No entanto, para validação do procedimento dinâmico é necessário gerar-se uma série de informações complementares de análise, tais como: análise do número de rejeições, renivelamentos e re-roteamentos, tempo de espera de admissão dos fluxos de dados, entre outros. Para rodadas de otimização significativas (milhares de fluxos, com tempos de simulação na faixa de horas) são executadas centenas de milhares de rodadas de otimização. Sendo assim, um conjunto grande de dados de informação para análise é gerado, podendo ser atingido o limite de memória RAM do sistema empregado, em casos complexos. Desta forma, a carga de trabalho (número de fluxos) teve de ser limitada nos experimentos executados.

- **Limitações de capacidade da ferramenta de simulação de rede (NS + MNS):** a limitação aqui é a capacidade do MNS em manipular um grande número de fluxos, no intervalo de simulação desejado. O MNS suporta o roteamento explícito de algumas centenas de fluxos. Isto significa que o número máximo de fluxos por rodada de otimização fica reduzido a algumas dezenas de fluxos simultâneos.

5.5.2 Desempenho do algoritmo

Esta seção apresenta os resultados da análise de desempenho do procedimento *PHP* usado para solução do problema dinâmico. Como descrito na seção anterior, são analisados o tempo de execução e o valor da função objetivo, como medida do grau de otimalidade, para diversos valores de carga de trabalho (quantidade de fluxos simultâneos por rodada de otimização).

A figura 5.4 mostra a variação dos valores da função objetivo para as várias cargas de trabalho experimentadas, respectivamente 549, 623, 754 e 798 fluxos, em média, por rodada de otimização.

Pela inspeção da figura 5.4, pode-se observar um crescimento dos valores da função objetivo proporcional às cargas de trabalho até a instância de 754 fluxos de dados em média por rodada de otimização. No caso de 798 fluxos de dados em média por rodada de otimização, os valores da função objetivo decaem, o que evidencia que esta instância corresponde a um ponto de operação dentro de uma região de saturação da rede. Ou seja, para cargas de trabalho experimentadas, acima de 754 fluxos em média por rodada de otimização, a rede não dispõe de recursos suficientes e muitos fluxos de dados têm seus níveis de transmissão reduzidos ou até mesmo são rejeitados na rede, evidenciando que a PETDIN não consegue alocar de forma

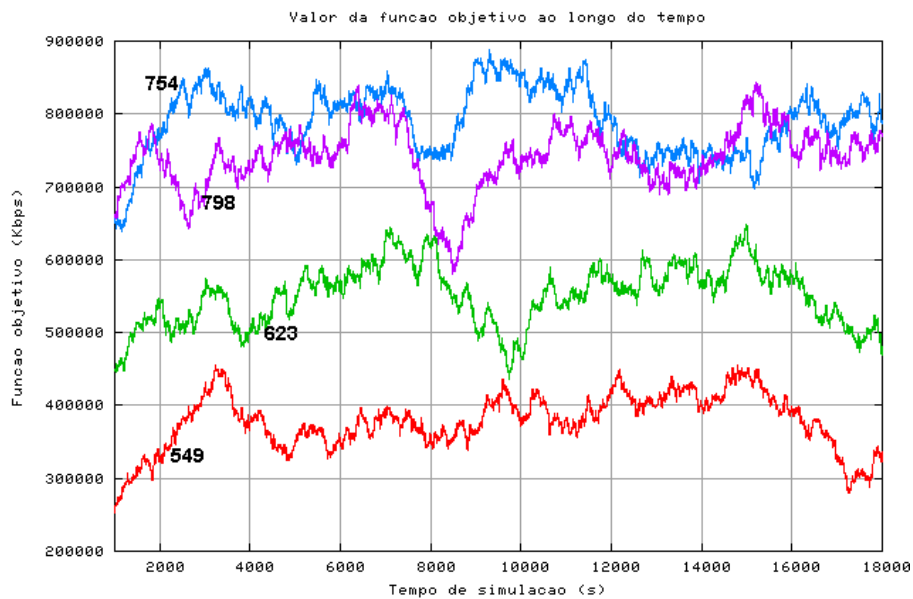


Figura 5.4: Valores de função objetivo ao longo do tempo para as várias cargas de trabalho experimentadas

eficiente os recursos de rede. Com isso, o *PHP* rejeita muitos fluxos de dados, o que justifica o comportamento não linear.

Tendo em vista que, nos casos experimentados, acima da instância de 754 fluxos de dados por rodada de otimização, estar fora do limite de linearidade dos experimentos realizados, este caso será considerado para análise dos demais parâmetros, como o limite operacional dos experimentos.

A figura 5.5 mostra a variação do tempo de execução do *PHP* ao longo do tempo de simulação, para a carga de trabalho de 754 fluxos de dados em média por rodada de otimização (pior caso).

Como se pode observar pela inspeção da figura 5.5, o tempo execução do *PHP* oscila entre 25 ms e 100 ms, com média de 41 ms, ao longo do tempo da simulação, para o pior caso considerado.

A conclusão desta sub-seção é que o uso do *PHP* para processamento de uma solução para um problema de ET dinâmico é viável, uma vez que o tempo de execução é relativamente pequeno, mesmo em casos com grande número de fluxos simultâneos a serem processados em cada rodada de otimização, como também foi verificado no capítulo anterior.

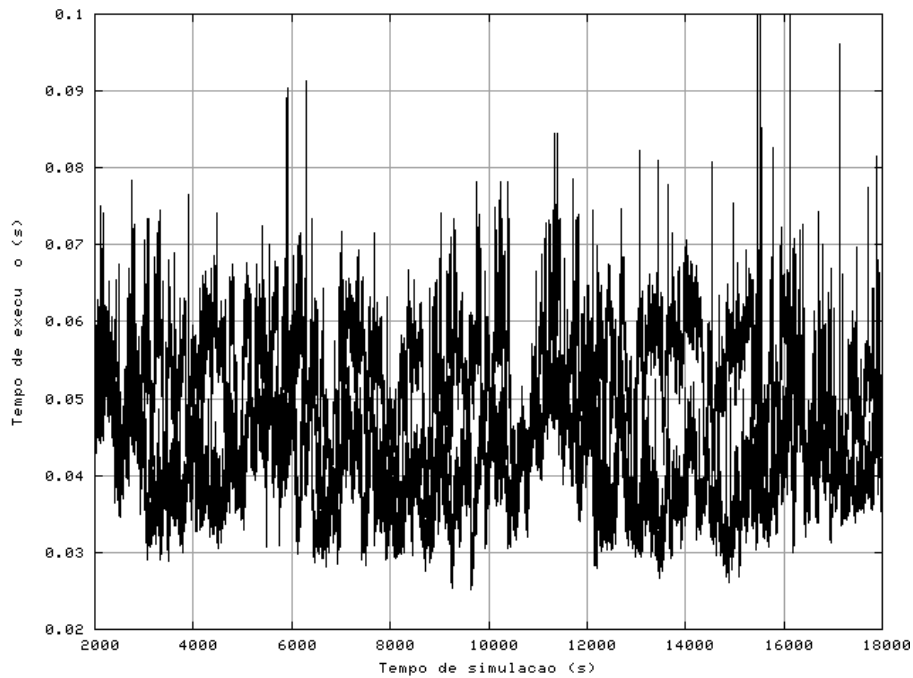


Figura 5.5: Tempo de execução ao longo do tempo de simulação para a carga de trabalho de 754 fluxos de dados, em média, por rodada de otimização

5.5.3 Controle de admissão

Esta seção apresenta os resultados das análises dos parâmetros de controle de admissão para os mesmos cenários da subseção anterior. São mostrados aqui os resultados dos diversos parâmetros de admissão: tempo de espera de admissão; distribuição dos níveis de largura de banda dos LSPs, nos quais são alocados os fluxos de dados; número de rejeições de fluxos de dados; número de interrupções; número de exclusões; número de re-roteamentos.

A tabela 5.1 mostra a frequência de distribuição de tempo de espera de admissão de fluxos de alta prioridade. O tempo é limitado em 60 s a partir do parâmetro de "time out".

Como se pode observar na tabela 5.1, a grande maioria dos fluxos de alta prioridade aguarda um tempo relativamente curto de admissão, concentrando-se na faixa entre 0.01 e 0.05 segundos. Por outro lado, dentro da estratégia de priorização do *PHP*, os fluxos de baixa prioridade, por sofrerem rejeições de admissão em algumas rodadas de otimização, aguardam tempos de admissão maiores. Praticamente um terço deles, aguardam a mesma faixa de tempo dos fluxos de alta prioridade. Uma parcela significativa dos fluxos de baixa prioridade (24,49 %) é excluído da fila de admissão por não conseguir ser admitido na rede em tempo inferior a 60 segundos ("time out"), enquanto nenhum fluxo de alta prioridade é excluído durante o tempo de

Tabela 5.1: Distribuição de frequência de tempo de espera de admissão dos fluxos de alta prioridade

Faixas de tempo (s)	% fluxos	
	Alta prioridade	Baixa Prioridade
De 0 a 0.001	1.41	1.88
De 0.001 a 0.005	10.33	7.60
De 0.005 a 0.01	13.42	9.73
De 0.01 a 0.05	72.26	32.92
De 0.05 a 0.1	2.12	1.59
De 0.1 a 0.5	1.36	2.41
De 0.5 a 1	1.42	6.28
De 1 a 5	1.47	3.35
De 5 a 10	1.62	6.65
De 10 a 60	0.36	3.06
Excluídos em 60	0.00	24.49

simulação.

A tabela 5.2 mostra a distribuição de frequência dos níveis de larguras de banda dos LSPs obtidas pelo *PHP*, para a carga de trabalho experimentada. O nível zero corresponde a rejeição do fluxo na rede. O nível 7 corresponde ao máximo nível de transmissão da rede.

Tabela 5.2: Distribuição de frequência de nível de largura de banda dos LSPs

Níveis	% Fluxos	
	Alta Prioridade	Baixa Prioridade
0	0.000	11.220
1	1.690	2.660
2	1.830	1.300
3	1.380	0.680
4	1.022	0.450
5	0.890	0.390
6	1.006	0.3400
7	91.980	82.920

A tabela 5.2 mostra que nenhum fluxo de alta prioridade foi rejeitado (nível de transmissão igual a zero). Além disso, a maioria absoluta dos fluxos de alta prioridade foi alocado com nível de transmissão máximo. O preço pago por este desempenho satisfatório dos fluxos de alta prioridade é um percentual relativamente alto de rejeições de fluxos de baixa prioridade — nível zero (11,22%). No entanto, a maioria dos fluxos de baixa prioridade admitidos na rede, foi alocado no nível de transmissão máximo — nível 7 (82,92%).

A tabela 5.3 mostra a distribuição de frequência de re-roteamentos dos fluxos de dados ao longo do tempo de simulação, para o caso experimentado.

Tabela 5.3: Distribuição de frequência de re-roteamentos dos LSPs

Número de Reroteamentos	% Re-roteamentos	
	Alta Prioridade	Baixa Prioridade
0	47.58	34.84
1	22.51	10.34
2	6.59	4.06
3	5.18	2.74
4	3.69	2.24
5 e 6	4.39	3.19
7 e 8	2.28	2.56
9 e 10	1.05	2.13
> 10	5.71	37.85

A tabela 5.3 mostra que para proporcionar o melhor aproveitamento dos recursos da rede, as rotas dos LSPs onde são encaminhados os fluxos de dados vão sofrendo algumas alterações de rotas, ao longo do tempo de simulação. Observamos que a maioria dos fluxos de alta prioridade sofre um menor número de re-roteamentos ao longo da simulação. Por outro lado, um pouco mais de um terço dos fluxos de baixa prioridade sofrem mais de 10 re-roteamentos ao longo do tempo de simulação. Outro terço dos fluxos de baixa prioridade não sofre nenhum re-roteamento, e o restante, entre 1 e 10 re-roteamentos ao longo do tempo de simulação.

A tabela 5.4 mostra a distribuição de frequência de renivelamentos dos fluxos de dados ao longo do tempo de simulação, para diversas cargas de trabalho experimentadas.

Tabela 5.4: Distribuição de frequência de renivelamentos dos LSPs

Faixas de Renivelamento.	% Renivelamentos	
	Alta Prioridade	Baixa Prioridade
0	90.06	55.53
1	4.48	9.10
2	1.58	2.94
3	0.35	1.46
4	0.43	1.75
5 e 6	0.70	2.31
7 e 8	0.35	1.77
9 e 10	0.17	1.49
> 10	1.75	23.61

Como podemos observar na tabela 5.4 os fluxos de alta prioridade sofrem poucos renivela-

mentos, sendo que a maioria absoluta (90,06 %) não sofre nenhum renivelamento ao longo do tempo de simulação. A maior parte dos fluxos de baixa prioridade (55,53 %) também não sofre nenhum renivelamento, no entanto, um percentual relativamente elevado de fluxos de baixa prioridade (23,61 %) sofre mais de 10 renivelamentos.

5.5.3.1 Alterando a relação *holding/setup*

Apesar dos resultados apresentados na sub-seção anterior não apontarem nenhum caso de interrupção de fluxos de alta prioridade foi avaliado um cenário de operação com maior carregamento da rede. Para este cenário foi aumentado o percentual de fluxos de alta prioridade de 1.5% para 25%. Neste caso, foram injetados na rede 549 fluxos de dados, em média por rodada, o que produziu uma elevada carga de trabalho, acarretando a interrupção de alguns fluxos de alta prioridade. A relação entre os valores *holding/setup* dos valores do parâmetro de prioridade foram configurados com valor igual a 2.

Nestas condições, 10 interrupções, em diferentes fluxos de alta prioridade, ocorreram durante o tempo de simulação, o que seria inaceitável se os mesmos correspondessem a uma aplicação tempo-real, como uma transmissão de vídeo.

Para tentar diminuir este efeito, foram configurados uma relação maior entre os valores *holding/setup* do parâmetro de prioridade dos fluxos de alta prioridade, elevando a relação para um valor bem maior, igual a 100. Feita esta configuração, neste caso específico, não ocorreu nenhuma interrupção de fluxos de alta prioridade.

Desta forma, se pode concluir que a seleção de uma relação entre os valores *holding/setup* dos parâmetros de prioridade podem interferir positivamente na redução das interrupções dos fluxos de alta prioridade, melhorando a QoS geral destes tipos de fluxos.

5.5.4 Garantia de QoS dos Fluxos de Alta prioridade

Nesta etapa procurou-se avaliar o impacto da solução do PETDIN nos parâmetros de QoS analisados neste trabalho: vazão, atraso fim-a-fim e variação de atraso fim-a-fim. Para tanto foi utilizado o simulador de rede Network Simulator (NS) (VINT 2003), com o módulo de extensão do NS denominado MPLS Network Simulator (MNS) (Ahn e Chun 2002).

Foram executados vários casos de simulação com uma menor quantidade de fluxos, do que nos casos apresentados na seção 4, devido as restrições do módulo MNS, citadas anteriormente.

As características dos fluxos e a topologia adotada são similares às da seção 4, com a redução da capacidade dos enlaces de núcleo para 2 Mbps e dos enlaces de acesso para 500 Kbps, para simular uma rede altamente carregada.

Foram injetados na rede 407 fluxos de dados no intervalo de simulação, que foi de 1 hora. Destes, 9 fluxos são do tipo alta prioridade, com duração de transmissão variando de 5 a 45 minutos. A cada rodada de otimização foram processados simultaneamente, em média, 50 fluxos de dados. Com esta configuração os parâmetros de desempenho da rede puderam ser avaliados pela comparação dos resultados para roteamento e controle de admissão baseado em *PHP* com os resultados obtidos pelo uso de protocolos do tipo IGP — RIP e OSPF.

A partir dos resultados, os fluxos de alta prioridade foram divididos em três categorias, de acordo com o grau de satisfação da QoS, avaliada a partir dos valores de vazão obtidos: *ótimo*, onde a vazão desejada (maior nível de transmissão do fluxo) foi alcançada; *bom*, onde a vazão média obtida alcançou mais que 70% do valor máximo desejado; e *regular*, onde a vazão média obtida alcançada ficou entre 51 % e 53 % do valor máximo desejado. Constata-se que, a partir dos resultados, o atraso observado com o emprego do *PHP* é sempre inferior ao atraso máximo desejado, para todos os fluxos de alta prioridade.

A tabela 5.5 mostra, além dos valores médios de vazão e do grau de satisfação de QoS, os atrasos fim-a-fim computados no intervalo de transmissão de cada fluxo de alta prioridade, comparados com os respectivos valores máximos desejados.

Tabela 5.5: Valores dos parâmetros de QoS analisados com o emprego do algoritmo *PHP*, para os fluxos de alta prioridade

Identificador do Fluxo	Vazão Desejada (Kbps)	Vazão Média Obtida (Kbps)	Satisfação de QoS	Atraso Desejado (ms)	Atraso Médio Obtido (ms)
102	567	567	Ótimo	50	20
142	404	404	Ótimo	40	20
299	447	447	Ótimo	40	10
014	608	450	Bom	60	50
311	527	390	Bom	40	25
391	680	500	Bom	40	15
154	753	390	Regular	60	50
234	764	400	Regular	60	50
295	737	380	Regular	40	25

Selecionou-se três fluxos para análise da vazão e atraso fim-a-fim ao longo do tempo, o

fluxo 299 da categoria definida como "ótimo", o fluxo 311 da categoria "bom" e o fluxo 234 da categoria "regular". Para todos os fluxos de alta prioridade não houve violações de atraso.

A figuras 5.6, 5.7 e 5.8 mostram as vazões dos fluxos usados como amostra, obtidas com o emprego do *PHP*, comparadas com a vazões obtidas pelo uso dos protocolos IGP.

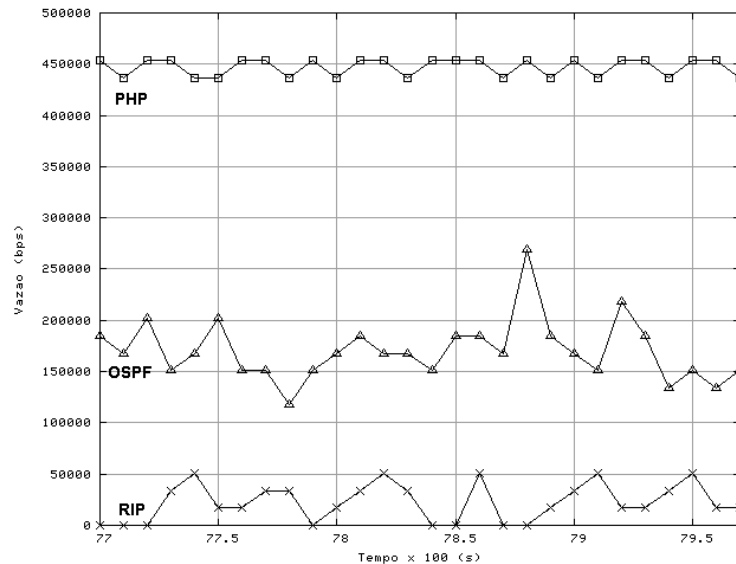


Figura 5.6: Vazão do fluxo 299 ao longo do tempo, pelo uso do *PHP* comparado ao uso do RIP e OSPF

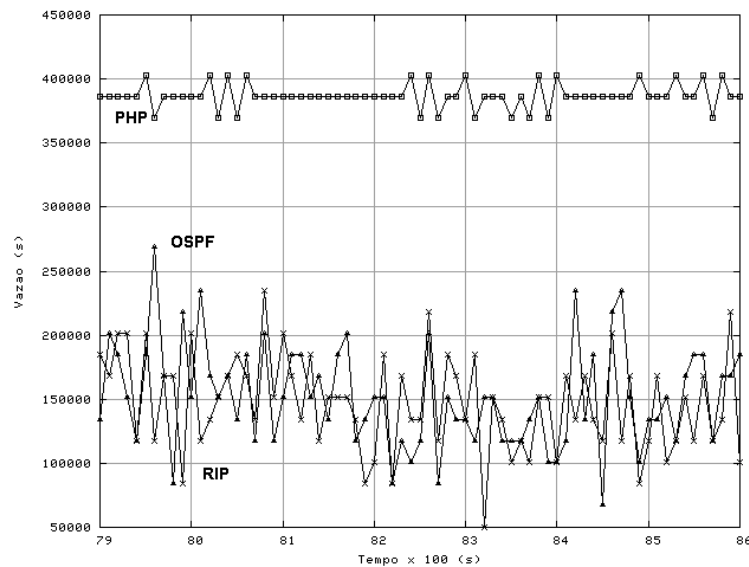


Figura 5.7: Vazão do fluxo 311 ao longo do tempo, pelo uso do *PHP* comparado ao uso do RIP e OSPF

Pela observação das figura 5.6 a 5.8 podemos concluir que para todas as amostras, a vazão obtida pelo uso do procedimento *PHP* é muito maior que a obtida pelos protocolos de roteamento convencional. Como a solução do *PHP* implica na eliminação dos congestionamentos

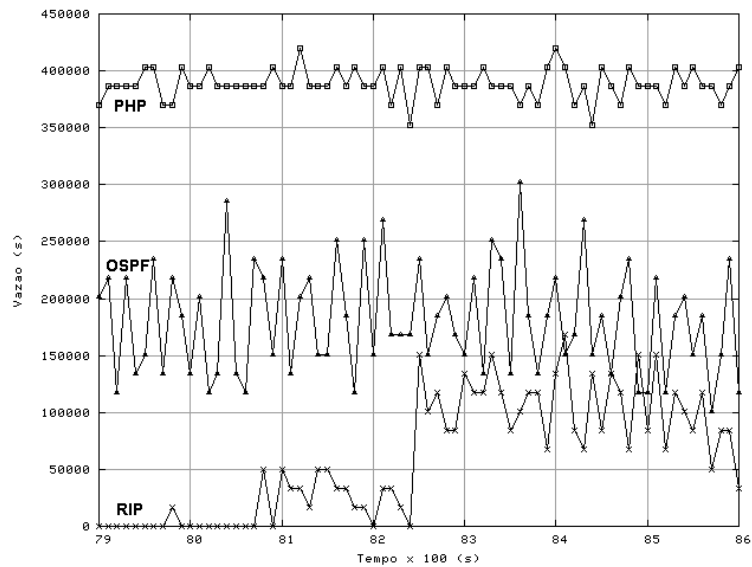


Figura 5.8: Vazão do fluxos 234 ao longo do tempo, pelo uso do *PHP* comparada com *RIP* e *OSPF*

e priorização de admissão dos fluxos de alta prioridade, a vazão tende a ser praticamente constante ao longo do tempo, no nível selecionado. Mesmo no pior caso, fluxo 234, a vazão obtida pelo *PHP* é superior a obtida pelo emprego do *RIP* e *OSPF* e tem pouca oscilação na sua amplitude.

As figuras 5.9, 5.10 e 5.11 mostram os atrasos fim-a-fim dos fluxos amostra, obtidos com o emprego da abordagem *PHP*, *RIP* e *OSPF*.

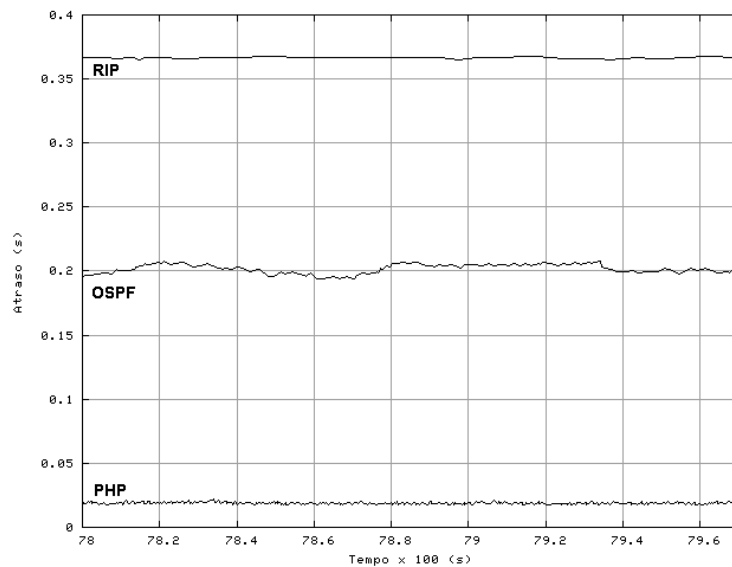


Figura 5.9: Atraso fim-a-fim do fluxo 299 ao longo do tempo, pelo uso do *PHP* comparado com *RIP* e *OSPF*

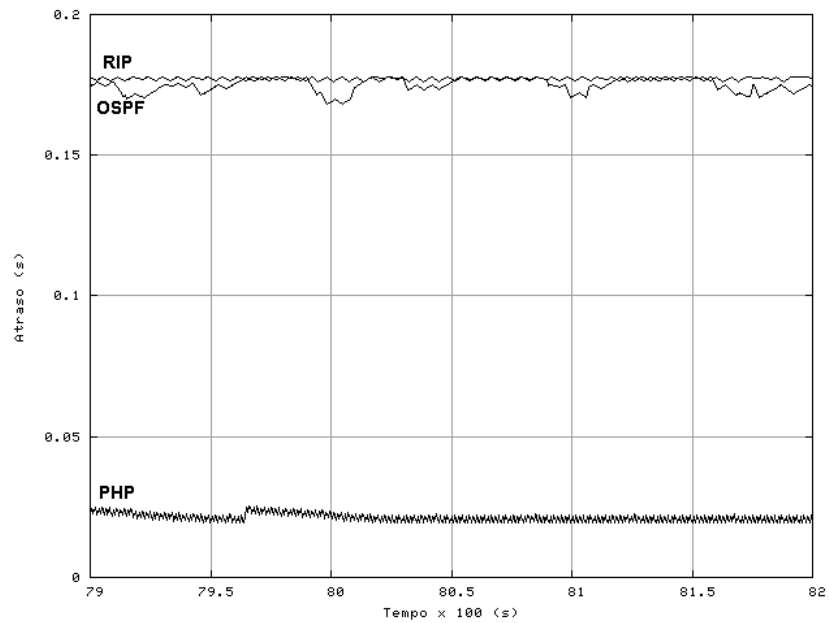


Figura 5.10: Atraso fim-a-fim do fluxo 311 ao longo do tempo, pelo uso do *PHP* comparado com RIP e OSPF

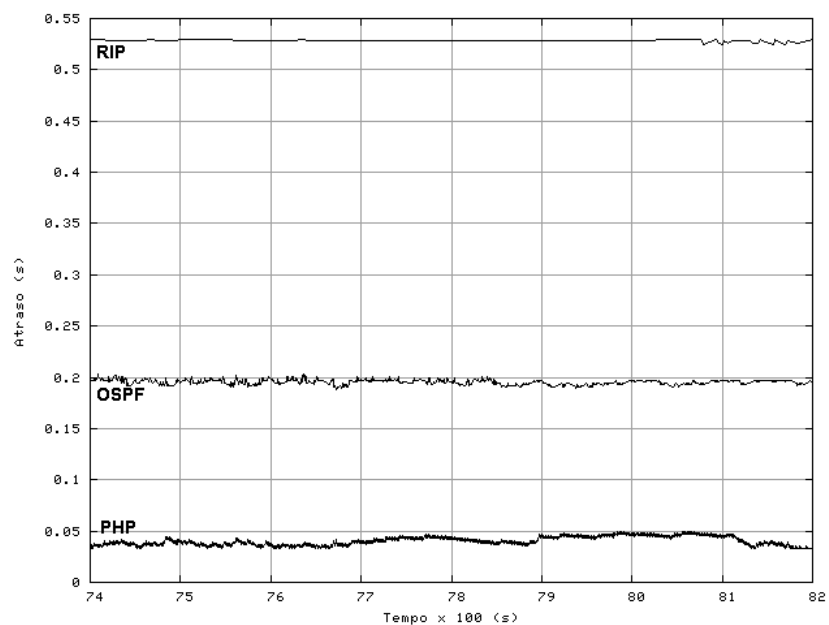


Figura 5.11: Atraso fim-a-fim do fluxo 234 ao longo do tempo, pelo uso do *PHP* comparado com RIP e OSPF

A observação das figuras 5.9 a 5.11 mostram que o atraso fim-a-fim obtido pelo emprego do *PHP* é pelo menos 4 vezes menor que os atrasos obtidos pelo roteamento convencional em todas as amostras. Além disso, com o uso do RIP a restrição de atraso do fluxo 234 é violada (maior que 50 ms).

Na proposta de solução atual para o PET não está prevista nenhuma restrição, no que diz respeito à limitação da variação de atraso dos fluxos de dados. No entanto, a adoção de uma relação *holding/setup* diferenciada para os fluxos de alta prioridade, diminui a probabilidade dos fluxos de alta prioridade sofrerem mudanças de configuração (alteração de rotas e/ou níveis). No entanto, dependendo do cenário a ser analisado, mesmo os fluxos de alta prioridade podem sofrer alterações de nível de transmissão e/ou mudanças nas rotas, ao longo das rodadas de otimização.

Tendo isto em vista, a tabela 5.6 mostra os valores médios de variação de atraso-fim-a-fim, computados no intervalo de transmissão de todos os fluxos de alta prioridade, obtidos com o uso do *PHP* e dos protocolos de roteamento IGP (RIP e OSPF).

Tabela 5.6: Valores de variação de atraso de transmissão médio dos fluxos de alta prioridade, medidos em segundos, pelo uso do PHP, RIP e OSPF

Identificador de Fluxo	Variação de atraso dos fluxos de alta prioridade (s)				
	PHP	RIP	OSPF	PHP > RIP ?	PHP > OSPF ?
102	0,000509	0,001366	0,000939	Não	Não
142	0,001021	0,001039	0,001511	Não	Não
299	0,000796	0,008074	0,001652	Não	Não
014	0,000448	0,001132	0,001428	Não	Não
311	0,001340	0,000721	0,000889	Sim	Sim
391	0,000012	0,001148	0,001191	Não	Não
154	0,001371	0,001493	0,001997	Não	Não
234	0,001295	0,005152	0,001333	Não	Não
295	0,001430	0,001492	0,002336	Não	Não

A observação da tabela 5.6 mostra que de uma maneira geral, a variação de atraso é menor com o emprego do PHP, quando comparado ao RIP e OSPF. Para a maior parte dos fluxos de alta prioridade, o número de re-configurações dos LPSs correspondentes é pequeno, como verificado anteriormente, fazendo com que a variação de atraso seja reduzida. O motivo para a variação de atraso ser maior para o RIP e OSPF, é função do alto congestionamento verificado nos casos experimentados, onde o efeito de enfileiramento dos pacotes nas filas dos roteadores,

provoca uma maior variação de atraso.

No entanto, o fluxo de alta prioridade, com número de identificação igual a 311, teve um ligeiro acréscimo de variação de atraso, quando compara-se o uso do *PHP* com os protocolos IGP. A razão para este fato, está no efeito dos eventuais renivelamentos e re-roteamentos sofridos pelo mesmo, devido ao fato de que durante a sua transmissão muitos outros fluxos de alta prioridade também estavam sendo transmitidos, o que exigiu duas mudanças de rotas para este fluxo, durante a sua transmissão para melhor alocar os recursos na rede.

De uma maneira geral, o procedimento *PHP* não degrada a variação de atraso dos fluxos de alta prioridade, apesar de não haver nenhuma garantia explícita para este parâmetro. Uma forma de diminuir a variação de atraso, seria aumentar o valor holding do parâmetro de prioridade dos fluxos de alta prioridade, fazendo com que os fluxos já admitidos tivessem menor probabilidade de ter suas configurações alteradas.

5.5.5 Conclusões sobre o controle de admissão do PHP

Com relação ao cenário analisado nesta seção, alguns dados sobre o controle de admissão do *PHP* são comentados a seguir.

Os fluxos de alta prioridade, como observado nos experimentos, aguardam somente um tempo de admissão curto, suficiente para processamento do *PHP*, não ultrapassando os 5 ms (tempo equivalente ao atraso dos enlaces de núcleo). Em torno de 3% dos fluxos de baixa prioridade aguardam de 5 a 10 s para serem admitidos na rede, o que é um tempo relativamente alto. Em torno de 12 % são excluídos da fila de admissão por atingirem um tempo limite (time out) de 60 segundos, para garantir uma maior alocação de recursos para os fluxos de alta prioridade.

Estes dados indicam que a degradação da qualidade dos fluxos de baixa prioridade é tolerável, tendo em vista o cenário de alto congestionamento experimentado. A taxa de perdas global (razão percentual entre o número de pacotes de dados transmitidos e pacotes de dados efetivamente recebidos), quando é empregado o protocolo de roteamento RIP é de 52,14 % quando é usado o roteamento OSPF é de 36.85 %, evidenciando a alta carga de trabalho para a topologia experimentada. Podemos concluir que o *PHP*, por empregar um controle de admissão, não apresenta perdas de pacotes, pois rejeita alguns fluxos de dados na eminência do congestionamento.

5.6 Trabalhos relacionados

Para a solução de problemas de ET estáticos, onde o processo de otimização ocorre num intervalo de tempo definido ou de forma sazonal, uma das abordagens de solução baseia-se no uso da Relaxação Lagrangeana (RL), que pode encontrar soluções ótimas ou sub-ótimas, como apresentadas em (Dias et al. 2004) (Dias et al. 2003a), (Dias et al. 2003b) e (Cardoso et al. 2001). Estes trabalhos propõem o uso da RL para solução de problemas de planejamento de capacidade de redes. Apesar de poderem ser resolvidos em tempo polinomial, seu emprego para solução de problemas de ET dinâmicos são limitados a pequenas topologias de rede, pelo elevado tempo de execução obtido com esta abordagem.

Uma das abordagens para os problemas de ET dinâmicos emprega o algoritmo da mínima interferência — Minimal Interference Routing Algorithm (MIRA) descrito em (Kodialam e Lakshman 2000). O MIRA busca a seleção de caminhos de rede que maximizem os fluxos dos dados, de forma que a rota selecionada interfira minimamente na maximização dos fluxos que serão encaminhados entre outros pares origem/destino. Para tanto, ele usa o algoritmo *maxflow* (Cormen et al. 1990). Em (Suri et al. 2001), o MIRA é empregado para implementação de um controle de admissão que evite a violação da largura de banda dos enlaces e atenda as especificações de vazão definidas em um acordo de nível de serviço — Service Level Agreement (SLA). A principal limitação do MIRA é a complexidade do algoritmo *maxflow*, que limita a escalabilidade da solução. Em geral, a solução do MIRA tem complexidade proporcional ao quadrado do número de enlaces das topologias (Kodialam e Lakshman 2000), enquanto o *PHP* varia logaritmicamente. Os problemas resolvidos pelo uso do MIRA tem somente uma única família de restrições a ser verificada (capacidade de transmissão dos enlaces), nem emprega um controle de admissão. O *PHP*, por outro lado, além das restrições de largura de banda, respeita as restrições de atraso fim-a-fim dos fluxos de dados.

Uma outra abordagem para solução de problemas de ET dinâmicos se baseia no balanceamento de carga entre os enlaces de rede, de forma que não viole a capacidade dos enlaces, sem contudo, otimizar algum parâmetro de interesse, como a vazão dos fluxos de dados. Entre os trabalhos deste tipo de abordagem podemos citar (Salvadori e Batiti 2003). Nenhuma restrição adicional, além da capacidade dos enlaces é verificada na solução do problema proposto pelos autores, enquanto na abordagem proposta neste trabalho, se objetiva a maximização de vazão sujeita a duas restrições (largura de banda dos enlaces e atraso fim-a-fim dos fluxos de dados).

Por fim, pode-se citar abordagens onde se busca otimizar o uso dos recursos da rede, respeitando-se uma ou mais restrições. Nesta categoria de problemas podemos citar o trabalho de (Banerjee e Sidhu 2002), onde são apresentadas heurísticas para solução de um problema de ET dinâmico, multi-objetivo (visando a minimização da redução das taxas de transmissão dos fluxos; minimização dos custos dos enlaces — i.e atraso de transmissão; minimização de caminhos críticos que possam levar ao congestionamento), sujeito a múltiplas restrições (largura de banda dos enlaces e atraso fim-a-fim dos fluxos de dados). O *PHP* se enquadra nesta categoria de problemas, apesar de ser mono-objetivo (maximização da vazão global). Já está em estudo, como continuação deste trabalho, um novo modelo de solução para o PET, com um objetivo de ET adicional — minimização das rejeições de fluxos de dados.

A presente proposta de trabalho surge como uma alternativa viável para solução de problemas de ET dinâmicos, sujeito a múltiplas restrições de QoS, com eficiência e escalabilidade superior às abordagens que empregam Relaxação Lagrangeana. Além disso, uma propriedade não encontrada nos trabalhos citados é a abordagem de controle de admissão que se baseia na configuração da largura de banda e a seleção das rotas dos LSPs, de acordo com uma estratégia de diferenciação de serviços, baseada no parâmetro de prioridade de admissão. Neste controle de admissão, a especificação da vazão em níveis, favorecendo o uso de notação de programação linear inteira, também é um aspecto relevante e inédito.

5.7 Conclusões do capítulo

Os resultados obtidos indicam que os fluxos de alta prioridade apresentaram uma grande melhoria da QoS quando é empregado o procedimento *PHP* para solução do PETDIN, em comparação ao uso de roteamento convencional IGP. Os comportamentos da vazão e atraso fim-a-fim de todos os fluxos de alta prioridade, que não foram apresentados, seguem os resultados apontados para os fluxos amostra. Verificando-se que quando se usa o *PHP* para encaminhar os fluxos de dados, o valor da variação de atraso da maioria dos fluxos de alta prioridade foi melhor que a obtida pelo roteamento IGP.

No entanto, o preço pago pela melhoria de QoS dos fluxos de alta prioridade, em uma rede altamente carregada, é uma redução na QoS de alguns fluxos de dados de baixa prioridade, devido a ação do controle de admissão do *PHP*, que provoca, entre outros, os seguintes efeitos: (i) rejeição de fluxos de baixa prioridade em determinadas rodadas do *PHP*, pela insuficiência

de recusos de rede; (ii) aumento do tempo de espera de admissão dos fluxos de baixa prioridade, que são rejeitados durante algumas das rodadas de otimização e (iii) redução do nível de transmissão de alguns fluxos de dados, quando a carga da rede é muito elevada.

Diante do exposto nas seções anteriores, podemos concluir que o uso do algoritmo *PHP* é eficiente para solução de problemas de ET dinâmicos, podendo ser empregado sem grandes alterações na infra-estrutura de redes pré-existentes. O seu emprego melhora a qualidade de serviço dos fluxos de alta prioridade, seguindo uma estratégia de diferenciação de serviços, em detrimento dos fluxos de baixa prioridade, sendo útil em cenários de operadoras de telecomunicações que desejam oferecer contratos de nível de serviço diferenciados para os seus clientes.

Mostrou-se ainda, neste capítulo, que o *PHP* proporciona um bom grau de otimalidade, com baixo custo computacional e que a complexidade do *PHP* é inferior ou similar à complexidade dos principais algoritmos de solução de problemas de ET. Além disso, o *PHP*, ao contrário das outras abordagens, emprega um controle de admissão que proporciona uma possibilidade de diferenciação de serviços.

Capítulo 6

Conclusão Final e Perspectivas de Trabalhos Futuros

O objetivo do problema de Engenharia de Tráfego (PET), proposto neste trabalho, visa a maximização da vazão global dos fluxos de dados injetados em uma topologia de rede de tamanho significativo. Para tanto são configurados caminhos de menor distância métrica entre origem e destino dos fluxos de dados, bem como das larguras de bandas de LSPs, nos quais os fluxos de dados serão encaminhados. Estas configurações devem respeitar as restrições de largura de banda dos enlaces e atraso fim-a-fim máximo, de cada fluxo de dados. Do ponto de vista da teoria de otimização, este problema é enquadrado como um *problema de caminhos mínimos sujeito a restrições*, sendo do tipo NP-completo. Para resolver este problema foi proposto um algoritmo heurístico denominado Procedimento Heurístico Puro (PHP).

Na abordagem proposta foi adotada uma política de diferenciação de serviços que depende de um controle de admissão e da configuração de largura de banda dos LSPs, baseadas em parâmetros de prioridade para cada fluxo de dados. Foram analisados, neste trabalho, cenários onde os fluxos foram classificados como sendo de *alta prioridade* e de *baixa prioridade*. Nesta política, na eminência de congestionamento da rede, admite-se que os fluxos de baixa prioridade podem ter suas vazões reduzidas até zero (fluxos rejeitados) para garantir os requisitos de QoS dos fluxos de alta prioridade.

A largura de banda a ser configurada nos LSPs foi modelada em níveis, ou seja, cada LSP será configurado com um desses possíveis níveis necessários à alocação dos fluxos de dados. Estes níveis podem variar de um nível de largura de banda igual a zero até um valor máximo

desejado. A discretização das larguras de banda a serem configuradas nos LSPs permite a formulação do problema em termos de programação linear inteira, o que significa um tratamento computacional menos complexo do que em um problema não-linear.

Segundo um dos objetivos específicos do trabalho, foi possível estabelecer os fundamentos para o desenvolvimento de uma ferramenta de planejamento de capacidade em redes, utilizando procedimentos de relaxação Lagrangeana e/ou heurísticos. Destaca-se também entre os objetivos propostos a solução de problemas de ET dinâmicos. O uso do procedimento PHP permite o seu emprego em cenários de operação dinâmica, devido ao seu ótimo desempenho e qualidade de solução, como demonstrado nos resultados dos experimentos e simulações.

6.1 Contribuições

Como discutido anteriormente, muitos trabalhos correlatos empregam o algoritmo MIRA para solução de problemas de ET dinâmicos. No entanto, o MIRA é empregado em problemas monobjetivo sujeito a uma única restrição. Além disso, o MIRA tem problemas de desempenho em função da complexidade do algoritmo MAXFLOW.

Neste contexto, uma importante contribuição deste trabalho foi o desenvolvimento de um procedimento heurístico simples, porém eficiente, que apresentou, conforme verificado nos experimentos, soluções de qualidade. O procedimento *PHP* é rápido em relação aos algoritmos baseados no MIRA e resolve problemas de caminhos mínimos sujeitos a múltiplas restrições, com bom grau de otimalidade.

Na formulação matemática empregada, a contribuição vem no sentido da incorporação de uma estratégia de diferenciação de serviços, pelo uso do parâmetro de prioridade. Nos experimentos adotados no trabalho, foram empregados somente duas classes de serviços, mas o modelo suporta um número indeterminado de classes, favorecendo a integração da solução proposta com a arquitetura DiffServ.

Ainda em relação à formulação matemática, a discretização das taxas de transmissão dos fluxos de dados em níveis foi uma estratégia inovadora, permitindo o uso de programação linear inteira na solução.

Finalmente, a abordagem empregada não exige grandes alterações na infra-estrutura de redes existente. Praticamente nenhuma alteração nos equipamentos de rede é exigida, nem nos protocolos de rede utilizados. No entanto, um modelo de gerenciamento baseado em políticas

deve ser empregado. Somente os dispositivos de borda (LERs) devem ter incorporadas funcionalidades similares a dos PEPs (police enforcement points), a fim de transmitir as informações lógicas (demandas de taxa de transmissão e limites de atraso máximo fim-a-fim dos fluxos) a uma entidade centralizada, similar ao PDP (police decision point) para processar o *PHP*.

6.2 Perspectivas de Trabalhos Futuros

A fim de aprimorar e estender as funcionalidades da solução proposta, podem ser citadas como perspectivas de trabalhos futuros:

- **Aprimoramento da modelagem matemática — Problema multi-objetivo:** desenvolvimento de um novo modelo de otimização multi-objetivo. Como discutido anteriormente, o problema apresentado é do tipo mono-objetivo — maximização da vazão dos fluxos de dados injetados na topologia. No entanto os experimentos demonstraram que, principalmente, os fluxos de baixa prioridade podem ser penalizados com a rejeição na rede. Sendo assim, um objetivo adicional de ET poderia ser incluído na função objetivo para minimizar as rejeições dos fluxos de dados. Além disso, um outro objetivo de ET que poderia ser incorporado ao modelo, seria a minimização das reconfigurações dos LSPs (re-roteamentos e renivelamento) de forma a reduzir a variação de atraso, sobretudo dos fluxos de alta prioridade. Para tanto, um novo modelo matemático e heurísticas de solução deverão ser desenvolvidos. A complexidade deste novo modelo exigirá o desenvolvimento de heurísticas eficientes para que as mesmas possam ser empregadas em tempo real;
- **Aprimoramento da implementação dos algoritmos de otimização:** como os algoritmos apresentados podem ser resolvidos em tempo polinomial, a fim de melhorar a eficiência os mesmos poderiam ser implementados com suporte a processamento paralelo, podendo ser executados de forma mais rápida. Esta tarefa exigirá a re-codificação de todos os algoritmos e uso de plataforma que suporte processamento simétrico e paralelo.
- **Implementação de protótipos:** através do uso da infra-estrutura de rede proposta no capítulo 5, com o emprego do modelo COPS, poderá ser desenvolvido um protótipo de ferramenta de gerenciamento de ET, baseada nos algoritmos desenvolvidos. Para tanto poderão ser utilizados roteadores com suporte a MPLS comerciais ou implementações baseadas em computadores PC, com sistema operacional GNU-Linux. Neste sentido, já

estão sendo desenvolvidos trabalhos junto ao projeto UCER: "Uso controlado e eficiente de redes IP sobre Tecnologia MPLS", com financiamento parcial do CNPq, em parceria com a empresa Wireless to Business (W2B), desenvolvido no Núcleo de Computação de Alto Desempenho e Redes de Alta Velocidade da UFSC (NURCAD).

- **Proposta de uma plataforma de engenharia de tráfego**, que possa seguir as especificações de um contrato de nível de serviço — Service Level Agreement, seguindo os procedimentos heurísticos desenvolvidos.

6.3 Comentários finais

O presente trabalho corrobora a viabilidade de utilização de ferramentas de engenharia de tráfego para melhoria geral da QoS de aplicações tempo-real em redes IP, como as multimídia distribuídas.

A maior parte dos trabalhos pesquisados aponta o uso da tecnologia MPLS como suporte necessário às implementações destas ferramentas. Sua capacidade de roteamento explícito, incremento de desempenho no encaminhamento dos pacotes e flexibilidade, tornam esta tecnologia promissora.

A fim de viabilizar a adoção das ferramentas de ET, as soluções devem ser simples, do ponto de vista de implementação e eficientes, do ponto de vista do tempo de execução. A abordagem proposta neste trabalho caminha neste sentido. Além disso, esta abordagem é bastante genérica, podendo ser empregado em outras tecnologias de rede que possibilitem o roteamento explícito.

A expectativa gerada com este trabalho é a possibilidade do seu emprego em cenários de operadoras de telecomunicações que necessitem oferecer diferentes níveis de serviço, possibilitando tarifação diferenciada.

Apêndice A

Gerenciamento baseado em políticas

Este apêndice visa introduzir conceitos básicos sobre gerenciamento baseado em políticas. Serão descritos os conceitos básicos e principais componentes de arquiteturas que propiciem a implementação e gerenciamento destas políticas. O enfoque deste apêndice é a descrição de uma infra-estrutura de gerenciamento de políticas proposto pelo IETF.

A.1 Conceitos básicos

A arquitetura Serviços Diferenciados (Blake 1998) desponta como a principal arquitetura para provimento de QoS em redes IP de grande abrangência como a Internet. A implementação desta arquitetura, entretanto, pelos provedores de serviço implica na obtenção por parte de alguns usuários ou aplicações a prestação de um serviço melhor do que a outros.

Algumas aplicações executadas em tempo real, como videoconferência, voz sobre IP, precisam de tratamento privilegiado pelos dispositivos de rede. Além disso, aplicações de gestão empresarial, com caráter transacional, poderão ter maior prioridade que aplicações assimétricas como tráfego de correio eletrônico.

Considerando que um dado domínio DiffServ abranja somente uma única corporação, a implementação de classes de serviços implicará em privilegiar uma aplicação em detrimento de outras. O controle de *quem* pode utilizar o tráfego de maior prioridade na rede exige uma infra-estrutura que permita o controle e o monitoramento. Num cenário de um provedor de serviços de Internet, o usuário que tem maior privilégio no acesso aos recursos da rede provavelmente pagará mais caro pelos serviços. Num cenário de uma empresa, as aplicações (ou usuários) que necessitem de maior prioridade no acesso aos recursos da rede corporativa devem

realmente ser privilegiados, evitando que tráfegos menos críticos sejam tratados com maior prioridade. Sendo assim, é necessário estabelecer-se regras (rules) e políticas (policies) que forcem a implementação destas regras.

Define-se como política (policy), de forma simplificada (Durham 2000), uma ou mais regras que descrevem as ações que devem ser executadas a partir de uma determinada condição. Um exemplo de regra é descrito no parágrafo a seguir.

Assumindo-se que a seguinte regra de negócio é implementada como política: prover um vídeo MPEG2 com baixa variação de atraso (jitter free), para usuários autorizados entre pontos de rede autorizados, mas somente em intervalos de tempo acordados.

Esta regra pode ser traduzida na semântica da infra-estrutura do gerenciamento de políticas da seguinte forma:

```
IF user IN ApprovedUsers, AND service IN VideoServices,  
AND Source IN Video Sources, AND destination IN VideoDestinations,  
AND time IN AprovedTimePeriods  
  
THEN provide JitterFreeMPEG2
```

Estas regras são escritas de forma declarativa, podendo ser empregada a notação ASN1 (CCITT 1988), comum na especificação de protocolos de rede. Neste exemplo as condições das políticas estão associadas a objetos pré-definidos e reconhecidos pela infra-estrutura (usuários, serviços, fonte, destino).

Uma política normalmente é composta de regras que são encadeadas de forma hierárquica a fim de facilitar o seu gerenciamento. O objetivo da implementação de um sistema de gerenciamento de políticas consiste em: políticas de alto nível (com regras especificadas numa linguagem próxima a linguagem natural) que correspondem aos termos de um Service Level Agreement (SLA); tradução destas regras de alto nível em políticas de baixo nível, que possam ser aplicadas aos dispositivos de rede (como políticas de enfileiramento, temporizações, controles de descartes de pacotes, esquemas de engenharia de tráfego, etc.).

As redes de computadores têm um comportamento não determinístico o que exige uma infra-estrutura complexa que tenha, basicamente, as seguintes funções:

- **Repositório de Informações:** representa uma estrutura de alto nível onde informações sobre os usuários e/ou requisitos das aplicações, dispositivos de rede são armazenadas. Este repositório pode estar armazenado em um servidor de banco de dados relacional ou em servidor de diretórios. Para acessá-lo devem ser utilizados mecanismos em ambiente cliente servidor como acessos SQL (Structured Query Language) para acesso a banco de dados ou o protocolo LDAP (Lightway Directory Application Protocol)(Wahl 1997) para acesso a serviços de diretórios;
- **Tomada de decisão:** implica em obter, interpretar, detectar conflitos nas políticas. A tomada de decisão pode ser executada num elemento centralizado, facilitando o gerenciamento. Este por sua vez é responsável em transmitir os resultados da tomada de decisão aos elementos que têm a função de implementá-la, bem como obter a realimentação do estado geral da rede a partir destes elementos terminais;
- **Imposição de políticas e monitoramento:** aplicação das decisões estabelecidas pela função de tomada de decisão. Normalmente é uma função exercida pelos nós de rede. Além disso este componente executa o exame ativo ou passivo da rede e seus elementos constituintes, a fim de que uma auditoria dos recursos possa realimentar as ações a serem estabelecidas pelos elementos que tomam as decisões, nesta infra-estrutura. A comunicação entre os elementos que tomam as decisões e os elementos que impõem as políticas, poderá ser efetuado a partir de mensagens trocadas entre os mesmos. Para tanto podem ser empregados protocolos do tipo "statefull" como o COPS (Common Open Police Services), ou "stateless" como o SNMP (Simple Network Management Protocol), ou até mesmo através de interfaces de linha de comando, "command line interfaces"(CLI).

A.2 Gerenciamento baseado em políticas do IETF

Inicialmente definida pelo grupo de trabalho RAP (Resource Allocation Protocol) para gerenciamento de políticas de controle para o protocolo RSVP (Resource Reservation Protocol), que é a base da arquitetura *Serviços Integrados* (Braden 1994), o IETF propôs uma infra-estrutura básica para gerenciamento de políticas de rede em (Durham 2000). Esta infra-estrutura foi imediatamente aplicada às necessidades do DiffServ e é suficientemente genérico para ser implementado em outros aspectos das tecnologias de rede como segurança (sendo

aplicável a firewall, sistemas de controle de acesso, IPsec, redes virtuais privadas "Virtual Private Networks"(VPNs).

A figura A.1 mostra o modelo de gerenciamento baseado em políticas proposto pelo IETF.

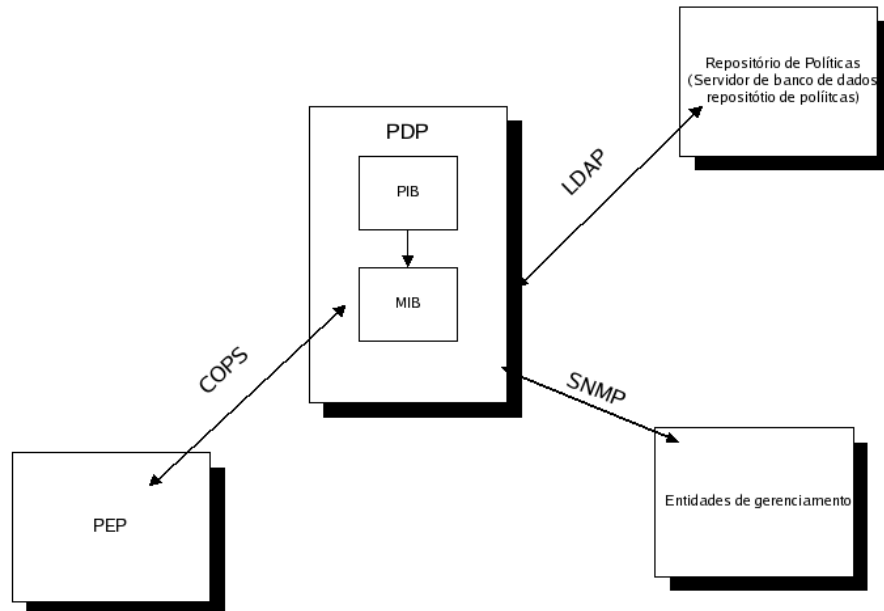


Figura A.1: Arquitetura para gerenciamento de rede baseado em políticas do IETF

Os componentes do modelo apresentado na figura A.1 são descritos a seguir:

- PEP (police enforcement point): é a componente com a função de imposição de políticas;
- PDP (police decision point) é o elemento com a função de tomada de decisão;
- Repositório de políticas: contém informações especificadas em linguagem declarativa de alto nível relativas ao conjunto de políticas definidas para um dado domínio administrativo. Estas informações podem ser acessadas a partir do protocolo LDAP (lightweight Directory Access Protocol). Este protocolo permite o acesso e manutenção do repositório de políticas pelo PDP;
- Servidor de Autenticação: elemento opcional que pode ser utilizado para autenticar e validar um determinado usuário de uma aplicação para ser admitido na rede;
- Entidade de gerenciamento: elemento que executa a função de monitoramento;
- PIB (police information base): as informações contidas no repositório de políticas, são definidas em linguagem declarativa de alto nível e contém informações sobre as regras de rede, que são recuperadas pelo PDP através do LDAP. Outras informações são ne-

cessárias para refletir o estado dinâmico da rede ou informações específicas de um fabricante de dispositivo (interface). Assim, a PIB é usada para suprir a necessidade destas informações específicas de uma interface. A PIB, portanto, contém informações de configuração de políticas que devem ser implementadas nos PEPs. As informações contidas na PIB, por outro lado, devem ser mapeadas em informações de gerência inteligíveis pelos dispositivos. Para tanto, são utilizados mapeamentos de PIBs para MIBs (management information base), de forma que se utilize o protocolo SNMP para gerenciamento de configuração dos dispositivos;

- COPS (Common Open Police Service): é um protocolo utilizado para enviar mensagens de sinalização de políticas entre os PEPs e o PDP. É um protocolo do tipo "statefull" (transmite informações de estado dos dispositivos em suas mensagens).

Assim, os PEPs registram seus estados nos PDPs e somente a cada mudança de estado o PEP envia nova mensagem. Isto evita uma consulta excessiva, verificada em protocolos do tipo "stateless" (sem informação de estado) como o SNMP, melhorando a performance.

A separação entre o PDP e o repositório de políticas é uma visão lógica do modelo. Estes elementos poderiam estar convivendo em um mesmo dispositivo físico. A definição atual do framework especificado em (Wahl 1997) sugere que os PEPs estejam contidos nos dispositivos de rede (onde são implementadas as ações correspondentes às políticas). O PDP deve estar localizado em uma entidade centralizada, que se comunica com os PEPs através do COPS para enviar as ações após a tomada de decisão e receber informações de estado dos PEPs. O mesmo documento sugere a existência de um ou no máximo dois PDPs e um ou no máximo dois repositórios de políticas. A existência de mais do que um PDP e mais do que um repositório de políticas tem a função de prover tolerância a falhas, mas a limitação nestes números tem a justificativa de simplificar a administração.

A.3 Relacionamento com a abordagem proposta

O intuito de discutir, neste anexo, o gerenciamento baseado em políticas foi de introduzir os conceitos básicos que permitam relacionar estes conceitos com a proposta de trabalho.

O procedimento de ET proposto neste trabalho é do tipo centralizado. As ações de configuração dos LSPs da tecnologia MPLS são implementadas na borda da rede pelos LERs.

Desta forma, a implementação da abordagem poderia utilizar a infra-estrutura de gerenciamento de políticas apresentada aqui.

Sendo assim, o procedimento PHP poderia ser executado numa entidade similar ao PDP, que toma a decisão de configuração dos LSPs na rede, a partir das informações lógicas e topológicas fornecidas pelos dispositivos de borda (LERs) que atuariam como PEPs. Cabe aos LERs, receber as informações relativas às demandas dos fluxos de dados (taxas de transmissão e atraso máximo fim-a-fim). Estas informações poderiam ser transmitidas ao PDP, utilizando um protocolo de sinalização como o COPS. A implementação das ações de configuração dos LSPs poderia implicar no desenvolvimento de PIBs específicas para mapear as decisões do PDP em ações de configuração executadas pelos LERs.

Referências Bibliográficas

- Ahn, G. e Chun, W. (2002). Architecture of MPLS network simulator (MNS), *Relatório técnico*, Department of Computer Engineering, Chungman University of Korea.
- Anderson, L. e. a. (2001). LDP Specification — Request for Comments.
- Aukia, P. e. a. (2000). RATES: A server for MPLS traffic engineering, *IEEE Network Magazine*, p. 34–41.
- Awduche, D. (2001). RFC 3209. RSVP-TE: Extensions to RSVP for LSP tunnels—request for comments.
- Awduche, D., Malcom, J., Agogbua, J., O’Dell, J. e McManus, J. (1999). Requirements for traffic engineering over mpls—request for comments.
- Banerjee, G. e Sidhu, D. (2002). Comparative analysis of path computation techniques for MPLS traffic engineering, *Computer Networks: The Int. Journal of Computer and Telecommunications Networking* **40**(1): 149–165.
- Banerjee, J. e. a. (2001). Generalized Multiprotocol Label Switching: An overview of routing and management enhancements, *IEEE Communication Magazine*.
- Bertsekas, D. P. (1995). *Nonlinear Programming*, Athena Scientific.
- Black, D. P. (1999). *Building Switched Networks*, Addison Wesley Longman.
- Blake, S. e. a. (1998). RFC 2475 - An Architecture for Differentiated Services.
- Braden, R. e. a. (1994). Integrated services in the internet architecture: an overview.
- Callon, R., Rosen, E. e Viswanathan, C. (2001). RFC 3031. Multiprotocol label switching architecture—request for comments.
- Cardoso, L., Patro, J., Lopes, C., de Sousa, A. e Valadas, R. (2001). A Tool for MPLS Traffic Engineering, *III Conferência de Telecomunicações.*, Figueira da Foz. Portugal.

- CCITT (1988). CCITT Recommendation X.208: Specification of Abstract Syntax Notation One (ASN.1).
- Cormen, T. H., Leiserson, C. E. e Rivest, R. L. (1990). *Introduction to Algorithms*, MIT Press.
- Dantzig, G. B. (1957). Discret variable extremum problem, *Opns. Res* **5**: 266–267.
- Das (2002). *Xpress-MP User Guide*.
- Davie, B. e Rekter, Y. (2000). *MPLS. Technology and Applications*, Morgan Kaufmann.
- Dias, R., Camponogara, E., Farines, J.-M., Willrich, R. e Campestrini, A. (2003a). Implementing traffic engineering in MPLS-based IP networks with Lagrangean relaxation, *IEEE ISCC 2003*, p. 373—378.
- Dias, R., Camponogara, E., Farines, J.-M., Willrich, R. e Campestrini, A. (2003b). Otimização Lagrangeana em Engenharia de Tráfego para Redes IP sobre MPLS, *XXI Simpósio Brasileiro de Redes de Computadores*, Natal. Brasil, p. 475–490.
- Dias, R., Camponogara, E., Farines, J.-M., Willrich, R. e Campestrini, A. (2003c). Using Lagrangean Relaxation to Improve Performance on IP Networks over MPLS, *Gestion de Reseaux et de Services — GRES'2003*, Ceará. Brasil, p. 27–37.
- Dias, R., Camponogara, E., Farines, J.-M., Willrich, R. e Campestrini, A. (2004). Engenharia de Tráfego em Redes IP sobre Tecnologia MPLS: Otimização Baseada em Heurísticas, *XXII Simpósio Brasileiro de Redes de Computadores*, Gramado. Brasil.
- Dias, R. e Willrich, R. (2001). Serviços Diferenciados Baseado na Tecnologia MPLS em Redes Heterogêneas, *Anais do VII Simpósio Brasileiro de Sistemas Multimídia e Hipermídia*, Florianópolis. Brasil.
- Durham, D. e a. (2000). RFC 2748. The COPS (common open policy service) protocol.
- Garey, M. R. e Johnson, D. S. (1979). *Computers and Intractability - A Guide to the Theory of NP-Completeness*, W. H Freeman and Company.
- Girish, M., Zhou, B. e Hu, J. (2000). Formulation of the traffic engineering problems in MPLS based IP networks, *IEEE ISCC*.
- Goldberg, M. C. e Luna, H. P. L. (2000). *Otimização Combinatória e Programação Linear*, Campus.
- Ishiguro, K. e a. (2003). Traffic engineering extensions to ospf version 3— internet draft.

- Jamoussi, B. e. a. (2002). Constraint-based lsp setup using ldp — request for comments.
- Kodialam, K. e Lakshman, T. (2000). Minimum interference routing with applications to MPLS traffic engineering, *IEEE INFOCOM*.
- Kompella, K. e. a. (2003). Traffic engineering extensions to ospf version 2— internet draft.
- Kompella, K. e Rekhter, Y. (2002). OSPF extensions in support of generalized MPLS. Internet Draft.
- Liu, G. e Ramakrishnan, R. (2001). A*prune: An algorithm for finding K shortest paths subject to multiple constraints, *IEEE INFOCOM 2001*, p. 743–749.
- Magalhães, M. e Cardoso, E. F. (2001). Engenharia de tráfego e mpls. Apostila do Departamento de Engenharia Elétrica da Universidade Estadual de Campinas.
- Medina, A., Lakhina, A., Matta, I. e Byers, J. (2001). Brite: Universal topology generation from a user’s perspective, *International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunications Systems – MASCOTS ’01*.
- Michalewicz, Z. e Fogel, D. B. (2000). *How to Solve It: Modern Heuristics*, Springer-Verlag.
- Nemhauser, G. L. e Wolsey, L. A. (1988). *Integer and Combinatorial Optimization*, John Wiley & Sons.
- Papadimitriou, C. H. (1998). NP-completeness: A retrospective, *ICALP*.
- Pepelnkak, I. e Guichard, J. (2001). *MPLS and VPN Architectures*, Cisco Press.
- Rabbat, R. (2000a). Traffic engineering algorithms using MPLS for service differentiation, *IEEE ISCC*, p. 791–795.
- Rabbat, R. e. a. (2000b). Traffic engineering algorithms using MPLS for service differentiation, *Proceedings of the International Conference on Communications*.
- Ricciato, F. S. e Angelo, S. B. (2002). Off-line configuration of a MPLS over WDM network under time-varying offered traffic, *IEEE INFOCOM*.
- Salsano, S., Ricciato, F., Listanti, M. e Belmonte, A. (2002). Off-line configuration of a MPLS over WDM network under time-varying offered traffic, *IEEE INFOCOM 2002*, p. 57–65.
- Salvadori, E. e Batiti, R. (2003). A Load Balancing Scheme for Congestion Control im MPLS Networks, *IEEE ISCC*.

- Suri, S., Waldvogel, M. e Warkhede, P. R. (2001). Profile-Based Routing: A New Framework for MPLS Traffic Engineering, *Quality of future Internet Services*, n. 2156 In: *Lecture Notes in Computer Science*, Berlin, p. 138–157.
- VINT (2003). *Network Simulator (NS)*. <http://www.isi.edu/nsnam/ns>.
- Wahl, M. e. a. (1997). RFC 2251 - Lightweight Directory Access Protocol (v3).
- Wang, Z. (1999). On the complexity of quality of service routing, *Information Processing Letters* **69**: 111–114.
- Willrich, R. e Dias, R. (2003). Estudo Experimental da Tecnologia MPLS: Avaliação de Desempenho e Qualidade de Serviço, *Workshop da Rede Nacional de Pesquisa*, Natal Brasil.
- Wolsey, L. A. (1998). *Integer Programming*, John Wiley & Sons.
- Wrocrawski, J. (2001). The use of RSVP whit IETF Integrated Services — Request for Comments.
- Xiao, X. e. a. (2000). Traffic Engineering with MPLS in the Internet, *IEEE Network*.
- Ziegelmann, M. (2001). *Constrained Shortest Paths and Related Problems*, PhD thesis.