

---

MESTRADO EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO

---

Implementação de  
**Algoritmos Geradores de Topologias Lógicas**  
Apropriadas a  
**Redes de Comunicações Ópticas**  
Baseadas em  
**Técnicas de Multiplexagem de Comprimentos de Onda**

---

**WDM**

---

Autora:

*Noélia Susana Costa Correia*



Unidade de Ciências Exactas e Humanas  
UNIVERSIDADE DO ALGARVE  
FARO - PORTUGAL

1997





*Noélia Susana Costa Correia*

Implementação de  
Algoritmos Geradores de Topologias Lógicas  
Apropriadas a  
Redes de Comunicações Ópticas  
Baseadas em  
Técnicas de Multiplexagem de Comprimentos de Onda

Dissertação realizada sob a orientação da Prof<sup>a</sup> Doutora Maria do Carmo de Medeiros,  
para a obtenção do grau de Mestre em Engenharia de Sistemas e Computação.

Unidade de Ciências Exactas e Humanas  
UNIVERSIDADE DO ALGARVE  
FARO – PORTUGAL  
1997

UNIVERSIDADE DO ALGARVE  
SERVIÇO DE DOCUMENTAÇÃO

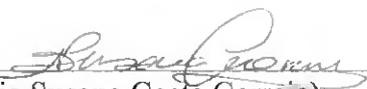
1-11	-5
621.39	
COR+JMP	

1

## DECLARAÇÃO DE ORIGINALIDADE

Em cumprimento do disposto na alínea b) do nº 2 do artigo 5º do Decreto-Lei nº216/92 de 13 de Outubro, Noélia Susana Costa Correia declara que todo o trabalho desenvolvido na presente dissertação é de sua autoria.

### Mestrando

  
\_\_\_\_\_  
(Noélia Susana Costa Correia)

### Orientador

  
\_\_\_\_\_  
(Profª Doutora Maria do Carmo Raposo de Medeiros)

*À minha família*

## **MENÇÃO DE APOIO**

Este trabalho foi apoiado financeiramente pelo sub-programa Ciência e Tecnologia do 2º Quadro Comunitário de Apoio, do Ministério da Ciência e Tecnologia.

## **AGRADECIMENTOS**

A realização desta dissertação contou com o apoio de um conjunto de pessoas às quais quero prestar os meus sinceros agradecimentos. À Prof<sup>ª</sup> Maria do Carmo de Medeiros pela orientação, estímulo e empenho pessoal que manifestou durante a realização deste trabalho. À Prof<sup>ª</sup> Marília Pires pelo apoio prestado na área da Investigação Operacional. Ao Stefano Baroni do Grupo de Redes Ópticas da UCL (University College of London) pelos comentários e sugestões. À Paula e Emanuel, colegas e amigos de todas as horas. Um obrigado muito especial ao Álvaro pelo apoio e carinho nas horas difíceis.

## RESUMO

Esta dissertação tem como objectivo o estudo e implementação de algoritmos geradores de topologias lógicas apropriadas a redes ópticas *multi-hop* baseadas em técnicas de multiplexagem de comprimento de onda (WDM). Para uma rede física e um dado cenário de tráfego, pretende-se desenhar uma topologia lógica e encaminhar o tráfego de forma a minimizar o congestionamento, o que se traduz numa maximização da quantidade de tráfego que a rede consegue transmitir. Para simular o cenário de tráfego foram usados processos *renewal* com comportamento *self-similar*.

O problema de criação de uma topologia lógica e encaminhamento do tráfego pode ser formulado sob a forma de um problema *Mixed Integer Linear Programming* (MILP). Esta abordagem torna-se, no entanto, intratável para redes de grandes dimensões. Por este motivo, o problema é dividido em duas partes: desenho da topologia lógica, em cuja resolução são usados algoritmos heurísticos, e encaminhamento do tráfego, formulado como um problema de programação linear. Esta dupla abordagem torna o problema mais tratável.

São feitos estudos comparativos do desempenho de vários algoritmos heurísticos usados para desenho da topologia lógica. Os parâmetros de rede analisados são: o nível de congestionamento, o grau de utilização de comprimentos de onda, o tempo de computação e as restrições de atraso. É também estudado o comportamento da rede perante valores dinâmicos de tráfego.

## SUMMARY

The aim of this dissertation is the study and implementation of algorithms able to generate logical topologies appropriate to multi-hop optical networks using wavelength division multiplexing (WDM). For a given physical network and traffic scenario, a logical topology must be created and the traffic directed in such a way that minimizes traffic congestion, i.e., maximizing the network throughput. Renewal processes with self-similar behaviour were used to simulate the traffic scenario.

The problem of creating a logical topology and the corresponding traffic routing can assume the shape of a Mixed Integer Linear Programming problem, an approach that is computationally heavy for large networks. The adopted method divides the problem in two parts: the logical topology design, which is implemented using heuristic algorithms, and traffic routing, formulated as a linear programming problem. This double approach became more feasible.

Comparative performance studies were made for the heuristic algorithms used for the logical topology design. The parameters under focus were the congestion level, wavelength consumption, computation time and delay restrictions. The network behaviour facing dynamic traffic values was also studied.

# ÍNDICE

<b>1. INTRODUÇÃO.....</b>	<b>1</b>
<b>2. REDES WDM E TECNOLOGIA DE SUPORTE .....</b>	<b>5</b>
2.1 INTRODUÇÃO .....	5
2.2 REDES DE COMUNICAÇÃO WDM .....	6
2.2.1 <i>Redes WDM de Comprimento de Onda Ágil</i> .....	7
2.2.2 <i>Redes WDM de Comprimento de Onda Fixo</i> .....	9
2.2.3 <i>Reutilização de Comprimentos de Onda</i> .....	10
2.2.4 <i>Routing em Redes WDM</i> .....	10
2.3 TECNOLOGIA DE SUPORTE.....	14
2.3.1 <i>Fibras Ópticas</i> .....	14
2.3.2 <i>Emissores Ópticos</i> .....	15
2.3.3 <i>Receptores Ópticos</i> .....	17
2.3.4 <i>Filtros Ópticos</i> .....	19
2.3.5 <i>Multiplexadores/Desmultiplexadores Ópticos</i> .....	22
2.3.6 <i>Amplificadores Ópticos</i> .....	24
2.4 EXEMPLOS ILUSTRATIVOS DE REDES WDM.....	25
2.4.1 <i>Rede Rainbow</i> .....	25
2.4.2 <i>Rede MONET</i> .....	27
2.5 SUMÁRIO .....	28
<b>3. OPTIMIZAÇÃO DE REDES MULTI-IOP.....</b>	<b>30</b>
3.1 INTRODUÇÃO .....	30
3.2 PARÂMETROS DE DESEMPENHO.....	31
3.3 FORMULAÇÃO DO PROBLEMA DE OPTIMIZAÇÃO .....	32
3.3.1 <i>Análise Detalhada das Restrições do Problema de Optimização</i> .....	33
3.3.2 <i>Forma Agregada do Problema</i> .....	37
3.4 HIPÓTESES DE RESOLUÇÃO .....	38
3.4.1 <i>Abordagem Global</i> .....	38
3.4.2 <i>Abordagem Parcial</i> .....	39

3.4.3 Complexidade das Possíveis Abordagens.....	41
3.5 SUMÁRIO.....	43
<b>4. ANÁLISE E IMPLEMENTAÇÃO DOS ALGORITMOS GERADORES DE TOPOLOGIAS LÓGICAS E OPTIMIZAÇÃO DE TRÁFEGO PARA REDES MULTI-HOP.....</b>	<b>44</b>
4.1 INTRODUÇÃO.....	44
4.2 ESTRUTURA DO OPTIMIZADOR.....	46
4.2.1 Desenho da Topologia Lógica.....	46
4.2.2 Algoritmo para Formulação do Problema de Encaminhamento de Tráfego.....	54
4.2.3 Algoritmo de Optimização.....	56
4.2.4 Análise Pos-Ótimo.....	62
4.3 SUMÁRIO.....	68
<b>5. SIMULAÇÃO DE TRÁFEGO EM REDES DE COMUNICAÇÃO.....</b>	<b>69</b>
5.1 INTRODUÇÃO.....	69
5.2 PROCESSOS <i>RENEWAL</i> .....	70
5.3 PROCESSOS COM COMPORTAMENTO <i>SELF-SIMILAR</i> .....	70
5.4 MODELOS ESTATÍSTICOS DE TRÁFEGO.....	71
5.4.1 Distribuição Pareto.....	71
5.5 EXEMPLO ILUSTRATIVO.....	72
5.6 SUMÁRIO.....	75
<b>6. APLICAÇÃO DOS ALGORITMOS DE OPTIMIZAÇÃO A UMA REDE WDM.....</b>	<b>76</b>
6.1 INTRODUÇÃO.....	76
6.2 REDE FÍSICA EM ESTUDO.....	76
6.3 MATRIZES DE TRÁFEGO.....	77
6.4 ANÁLISE DOS RESULTADOS.....	77
6.4.1 Níveis de Congestionamento.....	77
6.4.2 Utilização de Comprimentos de Onda.....	79
6.4.3 Tempos de Computação.....	82
6.4.4 Restrições de Atraso.....	86
6.4.5 Performance da Rede usando Redistribuição Frequente do Tráfego.....	87
6.4.6 Topologias Lógicas ShuffleNet e de Bruijn Graph.....	91

6.5 SUMÁRIO.....	94
7. CONCLUSÃO.....	96
APÊNDICE A – OS MÉTODOS SIMPLEX E SIMPLEX REVISTO .....	98
BIBLIOGRAFIA .....	109

## ÍNDICE DE FIGURAS

Fig. 2.1 - Evolução das capacidades de transmissão dos sistemas TDM e WDM. ....	5
Fig. 2.2 - Conceito básico de rede WDM. ....	7
Fig. 2.3 - Percursos ópticos numa rede WDM. ....	7
Fig. 2.4 - Conectividades lógicas para uma rede física. ....	8
Fig. 2.5 - Estrutura dos receptores de detecção directa e coerentes. ....	17
Fig. 2.6 - Selecção de um canal usando um filtro sintonizável. ....	19
Fig. 2.7 - Arquitectura da rede Rainbow. ....	26
Fig. 2.8 - Rede MONET. ....	28
Fig. 3.1 - Nó $i$ exemplo. ....	34
Fig. 3.2 - Topologia física exemplo. ....	41
Fig. 3.3 - Topologia lógica exemplo. ....	41
Fig. 4.1 - Diagrama do sistema de optimização. ....	44
Fig. 4.2 - Estrutura funcional do optimizador. ....	46
Fig. 4.3 - Topologia <i>ShuffleNet</i> ( $p=2, k=2$ ) de 8 utilizadores. ....	48
Fig. 4.4 - Topologia <i>de Bruijn Graph</i> ( $\delta=2, D=3$ ) de 8 utilizadores. ....	49
Fig. 4.5 - Rede exemplo. ....	51
Fig. 4.5 a) - Rede exemplo - passo 1. ....	52
Fig. 4.5 b) - Rede exemplo - passo 2. ....	52
Fig. 4.5 c) - Rede exemplo - passo 3. ....	52
Fig. 4.5 d) - Rede exemplo - passo 4. ....	52
Fig. 4.5 e) - Rede exemplo - passo 5. ....	52
Fig. 4.6 - Problema LP exemplo. ....	66
Fig. 4.6 a) - Problema LP exemplo: efeitos de uma alteração ao modelo inicial. ....	66
Fig. 4.6 b) - Problema LP exemplo: efeitos de uma nova alteração ao modelo inicial. ....	66
Fig. 5.1 - Distribuição <i>Pareto</i> . ....	72
Fig. 5.2 - Amostras da rede do Centro de Computação da Universidade de Auckland. ....	72
Fig. 5.3 - Histograma das amostras da rede do Centro de Computação da Universidade de Auckland. ....	72
Fig. 5.4 - Variância de decaimento lento das amostras da rede do Centro de Computação da Universidade de Auckland. ....	73
Fig. 5.5 - Simulação da distribuição de <i>Pareto</i> . ....	73
Fig. 5.6 - Histograma da simulação da distribuição de <i>Pareto</i> . ....	73
Fig. 5.7 - Variância de decaimento lento da simulação da distribuição de <i>Pareto</i> . ....	74
Fig. 5.8 - Simulação da distribuição de <i>Poisson</i> . ....	74
Fig. 5.9 - Histograma da simulação da distribuição de <i>Poisson</i> . ....	74
Fig. 5.10 - Variância de decaimento lento da simulação da distribuição de <i>Poisson</i> . ....	75
Fig. 6.1 - Rede NSFNET de 14 nós. ....	76
Fig. 6.2 - Níveis de congestionamento dos algoritmos geradores de topologias lógicas. ....	78
Fig. 6.3 - Topologias lógicas geradas pelos algoritmos para um <i>degree</i> igual a 1: a) HLDA; b) TANC. ....	79

Fig. 6.4 - Consumo de comprimentos de onda dos algoritmos geradores de topologias lógicas.....	80
Fig. 6.5 - Níveis de congestionamento obtidos pelos algoritmos geradores de topologias lógicas para diferentes números de comprimentos de onda disponíveis: a) HLDA; b) TANC; c) MLDA.....	80
Fig. 6.6 - Níveis de congestionamento obtidos para um número de comprimentos de onda disponível: a) 2; b) 4; c) 8; d) 12; e) infinito.....	81
Fig. 6.7 - Tempos de computação para um número crescente de nós numa rede física em anel usando o algoritmo TANC e formulação desagregada.....	82
Fig. 6.8 - Problema LP para um número crescente de nós numa rede física em anel usando o algoritmo TANC e tipo de formulação desagregada: a) Variáveis; b) Restrições.....	83
Fig. 6.9 - Previsão do número de variáveis e restrições do problema LP para um número crescente de nós numa rede física em anel usando o algoritmo TANC e tipo de formulação desagregada: a) Variáveis; b) Restrições.....	83
Fig. 6.10 - Tempos de computação para um número crescente de nós numa rede física em anel usando o algoritmo TANC e formulação agregada.....	84
Fig. 6.11 - Problema LP para um número crescente de nós numa rede física em anel usando o algoritmo TANC e tipo de formulação agregada: a) Variáveis; b) Restrições.....	84
Fig. 6.12 - Previsão do número de variáveis e restrições do problema LP para um número crescente de nós numa rede física em anel usando o algoritmo TANC e tipo de formulação agregada: a) Variáveis; b) Restrições.....	85
Fig. 6.13 - Tempos de computação obtidos nos estudos feitos na NSFNET usando formulação desagregada.....	85
Fig. 6.14 - Tempos de computação obtidos nos estudos feitos na NSFNET usando formulação agregada.....	85
Fig. 6.15 - Níveis de congestionamento para um <i>degree</i> de 4 e várias restrições de atraso.....	86
Fig. 6.16 - Atraso médio de um pacote para um <i>degree</i> de 4 e várias restrições de atraso.....	87
Fig. 6.17 - Comportamento das topologias lógicas geradas pelos algoritmos HLDA e TANC durante o período de 1000s para um <i>degree</i> igual a 1.....	88
Fig. 6.18 - Histogramas dos comportamento das topologias lógicas geradas pelos algoritmos HLDA e TANC durante o período de 1000s para um <i>degree</i> igual a 1.....	88
Fig. 6.19 - Comportamento das topologias lógicas geradas pelos algoritmos HLDA e TANC durante o período de 1000s para um <i>degree</i> igual a 2.....	89
Fig. 6.20 - Histogramas dos comportamento das topologias lógicas geradas pelos algoritmos HLDA e TANC durante o período de 1000s para um <i>degree</i> igual a 2.....	89
Fig. 6.21 - Comportamento da topologia lógica gerada pelo HLDA durante o período de 1000s para um <i>degree</i> igual a 6: a) Dados; b) Histograma.....	90
Fig. 6.22 - Comportamento da topologia lógica gerada pelo TANC durante o período de 1000s para um <i>degree</i> igual a 6: a) Dados; b) Histograma.....	90
Fig. 6.23 - Comportamento da topologia lógica gerada pelo MLDA durante o período de 1000s para um <i>degree</i> igual a 6: a) Dados; b) Histograma.....	90
Fig. 6.24 - Comparação dos histogramas do comportamento das topologias lógicas geradas pelos algoritmos HLDA, TANC e MLDA durante o período de 1000s para um <i>degree</i> igual a 6.....	91
Fig. 6.25 - Topologia física para estudo das topologias lógicas <i>shuffle</i> ( $p=2,k=2$ ) e de <i>Bruijn</i> ( $\delta=2,D=3$ ).....	92
Fig. 6.26 - Níveis de congestionamento para uma rede de 8 nós e dois tipos de tráfego.....	93
Fig. 6.27 - Consumo de comprimentos de onda para rede de 8 nós e dois tipos de tráfego.....	93
Fig. A.1 - Exemplo de um problema LP.....	98

## ÍNDICE DE TABELAS

<b>Tabela 2.1</b> - Evolução dos sistemas WDM. ....	6
<b>Tabela 2.2</b> - Características da fibra <i>single-mode</i> . ....	15
<b>Tabela 2.3</b> - Características de lasers sintonizáveis. ....	17
<b>Tabela 2.4</b> - Características de filtros ópticos sintonizáveis. ....	20
<b>Tabela 3.1</b> - Número de variáveis e restrições da rede exemplo usando abordagem global. ....	41
<b>Tabela 3.2</b> - Número de variáveis e restrições da rede exemplo usando abordagem parcial. ....	41
<b>Tabela 6.1</b> - Matriz de tráfego médio simulado ao longo de 1000s. ....	77
<b>Tabela 6.2</b> - Valores de congestionamento dos algoritmos geradores de topologias lógicas. ....	78
<b>Tabela 6.3</b> - Valores dos tempos de computação obtidos nos estudos feitos na NSFNET usando formulação desagregada. ....	85
<b>Tabela 6.4</b> - Valores dos tempos de computação obtidos nos estudos feitos na NSFNET usando formulação agregada. ....	85
<b>Tabela 6.5</b> - Valores dos níveis de congestionamento para um <i>degree</i> de 4 e várias restrições de atraso. ....	86
<b>Tabela 6.6</b> - Valores dos atrasos médios de um pacote para um <i>degree</i> de 4 e várias restrições de atraso. ....	87
<b>Tabela 6.7</b> - Valores dos tempo de computação da análise pos-ótimo para um período de 1000s. ....	91
<b>Tabela 6.8</b> - Matrizes de tráfego médio para as topologias lógicas: a) <i>ShuffleNet</i> ; b) <i>de Bruijn Graph</i> . ....	92
<b>Tabela 6.9</b> - Valores de congestionamento para uma rede de 8 nós e dois tipos de tráfego. ....	93

## ÍNDICE DE LISTAGENS

<b>Listagem 4.1</b> - Algoritmo <i>ShuffleNet</i> . ....	48
<b>Listagem 4.2</b> - Algoritmo <i>de Bruijn Graph</i> . ....	50
<b>Listagem 4.3</b> - Algoritmo HLDA. ....	50
<b>Listagem 4.4</b> - Algoritmo MLDA. ....	53
<b>Listagem 4.5</b> - Algoritmo TANC. ....	54
<b>Listagem 4.6</b> - Criação das variáveis do problema de encaminhamento usando a forma desagregada. ....	55
<b>Listagem 4.7</b> - Criação das variáveis do problema de encaminhamento usando a forma agregada. ....	56
<b>Listagem 4.8</b> - Algoritmo Simplex Revisto. ....	61
<b>Listagem 4.9</b> - Algoritmo da análise pos-ótimo. ....	67

## 1. INTRODUÇÃO

A área das telecomunicações está neste momento numa fase de crescimento acelerado. A procura de novos serviços como aplicações multimédia, transmissão de imagens de alta definição ou transferência de ficheiros volumosos, trará num futuro próximo níveis absolutamente inéditos de tráfego. Com as elevadas taxas de transmissão envolvidas serão necessárias redes de telecomunicações de banda larga capazes de suportar esses serviços. Neste contexto, com cerca de 30 THz de largura de banda, a fibra óptica surge como um meio de transmissão privilegiado, embora ainda altamente subaproveitado pelos sistemas actualmente em vigor. A técnica de multiplexagem de sinais ópticos no domínio da frequência WDM (*Wavelength Division Multiplexing*), onde sinais de diferente comprimento de onda são enviados em simultâneo, surge como uma forma de aproveitar a largura de banda disponibilizada pela fibra óptica. Nestes sistemas o espectro óptico é dividido em vários canais lógicos, tendo cada emissor posse exclusiva da sua faixa de comprimentos de onda. Esta técnica possui a vantagem de permitir uma velocidade máxima apenas dependente dos componentes intervenientes enquanto que, na técnica TDM (*Time Division Multiplexing*), por exemplo, a velocidade máxima permitida é agregada, sendo determinada pelo componente mais lento. Além disso, permite embutir uma conectividade lógica numa rede física através da atribuição dinâmica de comprimentos de onda a emissores e receptores, tornando possível acomodar conexões com diferentes protocolos em comprimentos de onda diferentes.

A técnica WDM quando aliada ao uso de componentes sintonizáveis torna possível alterar a conectividade lógica em resposta não só a alterações de tráfego mas também a falhas na rede, ou fazer a rede crescer por inserção de novos elementos. Esta independência entre a topologia física e a conectividade lógica traz grandes vantagens ao mundo das telecomunicações onde os serviços e as necessidades dos utilizadores evoluem de forma muito rápida.

A liberdade na construção da conectividade lógica obriga, no entanto, à tomada de decisões de desenho, isto é, a uma escolha da conectividade mais adequada. Estas decisões devem basear-se em previsões de performance conseguidas por simulação, que devem ter em consideração o compromisso entre o detalhe da simulação e o seu tempo de execução. Um elemento chave na simulação de redes de comunicações é a modelização de tráfego. Isto porque, uma boa compreensão da natureza do tráfego permite a selecção apropriada de um modelo, factor crítico para um encaminhamento eficiente da informação na rede. De salientar que decisões baseadas em previsões pobres podem prejudicar bastante o desempenho do sistema real.

Uma vez que a quantidade de tráfego transmitida (*throughput*) é uma medida de performance crítica em qualquer rede, é razoável ter como objectivo de desenho a criação de uma conectividade lógica e particionamento do tráfego com vista à minimização do congestionamento. Este é o objectivo desta dissertação.

Para além desta introdução, a dissertação possui mais seis capítulos dos quais se segue uma breve descrição:

**Capítulo 2** - São analisadas as características e vantagens de sistemas WDM *single-hop* e *multi-hop*, sendo realçado que as grandes capacidades disponibilizadas por estes sistemas exigem técnicas de *routing* dinâmico. É feita uma descrição dos componentes ópticos utilizados em redes WDM e são também introduzidos conceitos relacionados com as configurações de uma rede e o seu funcionamento. No final são analisados exemplos ilustrativos de redes WDM Rainbow e Monet actualmente em investigação.

**Capítulo 3** - O problema de desenho da topologia lógica e encaminhamento do tráfego para redes *multi-hop* é formulado matematicamente como um problema MILP (*Mixed Integer Linear Programming*) cujo objectivo é minimizar o congestionamento na rede, abordagem indicada

caso se pretenda maximizar o *throughput*. O problema é apresentado quer na forma desagregada quer na forma agregada. A abordagem adoptada divide este problema em duas partes: desenho da topologia lógica e encaminhamento do tráfego. A vantagem desta abordagem é ser de resolução mais rápida e permitir análise pos-óptimo para uma redistribuição frequente do tráfego.

**Capítulo 4** - É feita uma descrição de toda a concepção e implementação do sistema de optimização. Para a criação da topologia lógica foram implementados os algoritmos heurísticos HLDA, MLDA e TANC, o último dos quais foi desenvolvido pela autora desta dissertação, sendo proposto com o objectivo de corrigir os aspectos negativos do HLDA e MLDA. São também implementadas as topologias lógicas regulares *ShuffleNet* e *de Bruijn Graph*. Para optimização do tráfego foi implementado o algoritmo Simplex Revisto que possui a grande vantagem de permitir análise pos-óptimo eficiente, o que permitirá reajustar o tráfego de acordo com a matriz de tráfego do momento, evitando níveis de congestionamento elevados. Com o uso desta facilidade é feita uma proposta de funcionamento para redes *multi-hop*.

**Capítulo 5** - É analisada a importância da escolha adequada de um modelo que melhor descreva a chegada de pacotes nas redes de comunicações. Para esse efeito, e segundo estudos da Bellcore, são necessários processos *renewal* de comportamento *self-similar* que podem ser simulados por distribuições de variância infinita, como a *Pareto* usada neste trabalho. Os modelos matemáticos são confrontados com tráfego experimental e validados.

**Capítulo 6** - São feitos estudos comparativos de desempenho dos vários algoritmos. Para as simulações foi usada a rede física NSFNET e um cenário de tráfego gerado usando a distribuição de *Pareto*. Os parâmetros de rede utilizados no estudo serão: análise dos níveis de congestionamento obtidos, grau de utilização de comprimentos de onda, tempos de computação e restrições de atraso. Será também estudado o comportamento da rede face a valores dinâmicos

de tráfego usando análise pos-ótimo, técnica que este trabalho demonstrou ser uma alternativa bastante eficiente na obtenção de um encaminhamento ótimo quase permanente perante alterações frequentes do tráfego.

*Capítulo 7* - São tiradas conclusões sobre o trabalho realizado e são feitas propostas de trabalho futuro.

## 2. REDES WDM E TECNOLOGIA DE SUPORTE

### 2.1 INTRODUÇÃO

O conceito de multiplexagem no comprimento de onda, WDM, data do início da década de 70 e surgiu como uma solução capaz de aproveitar a vasta largura de banda disponível nas fibras ópticas ( $\cong 30$  THz). Inicialmente os sistemas WDM não tiveram grande impacto devido à complexidade e custo dos componentes envolvidos. Em termos tecnológicos e comerciais era preferível, para uma mesma capacidade, utilizar técnicas de multiplexagem por divisão no tempo (TDM).

O aparecimento do amplificador de fibra dopada de Erbium, caracterizado por uma grande largura de banda de amplificação ( $\cong 30$  nm) permitindo amplificar vários sinais com diferentes comprimentos de onda, fez ressurgir o interesse nos sistemas WDM que desde então têm vindo a progredir continuamente.

A figura seguinte compara a evolução dos sistemas baseados em técnicas TDM (*Time Division Multiplexing*) e WDM.

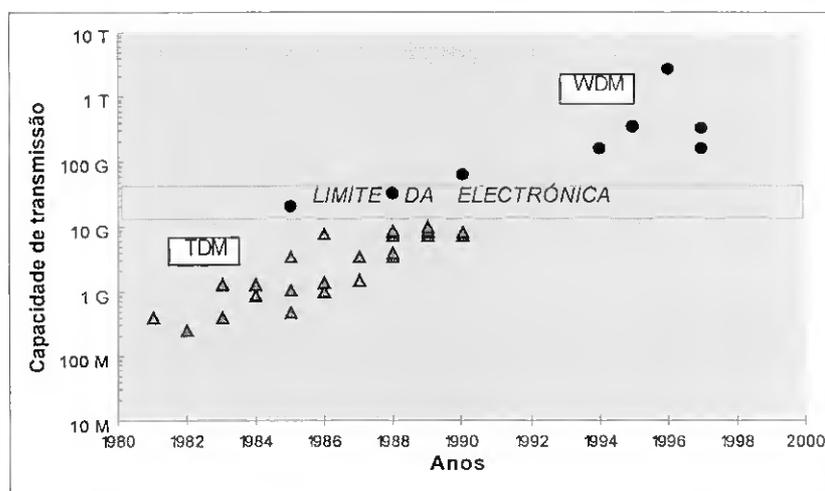


Fig. 2.1 - Evolução das capacidades de transmissão dos sistemas TDM e WDM.

Para taxas de transmissão superiores a 10 Gb/s, como mostra a figura, a tecnologia eleita é a WDM. Em 1985, 1988 e 1990 foram conseguidos sistemas WDM com 10, 16 e 100 canais,

correspondendo a taxas de transmissão agregadas de 20, 32 e 62 Gb/s respectivamente [1][2][3]. Em 1994 foi apresentado um sistema WDM com 8 canais de 20 Gb/s [4]. Em 1995 e 1996 foram feitas experiências em laboratório que permitiram transmissões de 17 canais a 20 Gb/s (340 Gb/s) e 132 canais a 20 Gb/s (2.64 Tb/s) respectivamente [5][6]. Relativamente às distâncias, as conseguidas inicialmente rondavam os 50 km, tendo as últimas 3 experiências sido feitas para 300 (50 km entre amplificadores), 150 (50 km entre amplificadores) e 120 km respectivamente. Muito recentemente foram demonstrados sistemas WDM para distâncias de 640 e 9300 km [7]. A seguinte tabela sumariza estes resultados.

Anos	Capacidade agregada (Gb/s)	Número de canais	Distância percorrida (km)
1985	20	10	68.3
1988	32	16	?
1990	62	100	50
1994	160	8	300
1995	340	17	150
1996	2640	132	120
1997	160	32	9300
1997	320	32	640

**Tabela 2.1** - Evolução dos sistemas WDM.

A aplicação primária dos sistemas WDM ponto a ponto, juntamente com os amplificadores ópticos, será no aumento da capacidade dos sistemas de telecomunicações de longa distância e na diminuição do congestionamento de troços metropolitanos.

Apesar das várias vantagens atribuídas aos sistemas WDM e do grande esforço que está a ser despendido a nível internacional, a sua implementação não está a decorrer tão rapidamente como previsto. Um dos entraves mais relevantes ao desenvolvimento destes sistemas é o seu elevado custo, determinado em grande medida pelos componentes de apoio.

## **2.2 REDES DE COMUNICAÇÃO WDM**

Do conceito de sistema WDM ponto a ponto rapidamente se passou ao conceito de rede WDM. Uma rede WDM, de um modo geral, é constituída por um conjunto de nós espalhados

geograficamente e que partilham o mesmo meio óptico, como se mostra esquematicamente na figura.

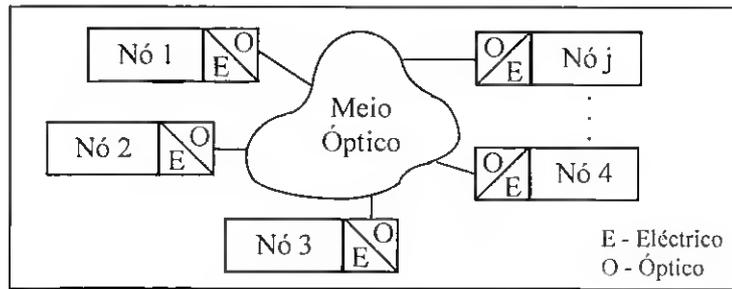


Fig. 2.2 - Conceito básico de rede WDM.

Existem basicamente duas alternativas de implementação de redes WDM. Redes em que os emissores e receptores transmitem e recebem num comprimento de onda fixo (redes WDM de comprimento de onda fixo) e redes em que tanto os receptores como os emissores são sintonizáveis (redes WDM de comprimento de onda ágil). Seguindo a ordem cronológica das propostas de rede, será feita referência primeiro às redes WDM de comprimento de onda ágil.

### 2.2.1 Redes WDM de Comprimento de Onda Ágil

Neste tipo de rede o emissor e/ou receptor podem ser sintonizados para um comprimento de onda específico. Deste modo, quando dois quaisquer nós *A* e *B* querem comunicar, o nó *A* terá que sintonizar o seu emissor para o comprimento de onda  $\lambda_1$  e o seu receptor para  $\lambda_2$ , por exemplo, enquanto o nó *B* terá que sintonizar o seu emissor para  $\lambda_2$  e o seu receptor para  $\lambda_1$ . Deste modo podem ser estabelecidos diferentes percursos ópticos (canais) numa rede, como ilustra a figura.

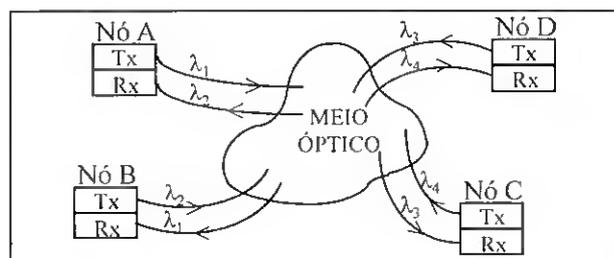


Fig. 2.3 - Percursos ópticos numa rede WDM.

Através da escolha apropriada de comprimentos de onda para os emissores e receptores, pode embutir-se uma conectividade lógica numa rede física. Dito de outra forma, através da sintonização dos transmissores e receptores para comprimentos de onda específicos, pode definir-se um modelo de ligação entre nós, modelo este que pode ser alterado dinamicamente. Esta situação é ilustrada na seguinte figura.

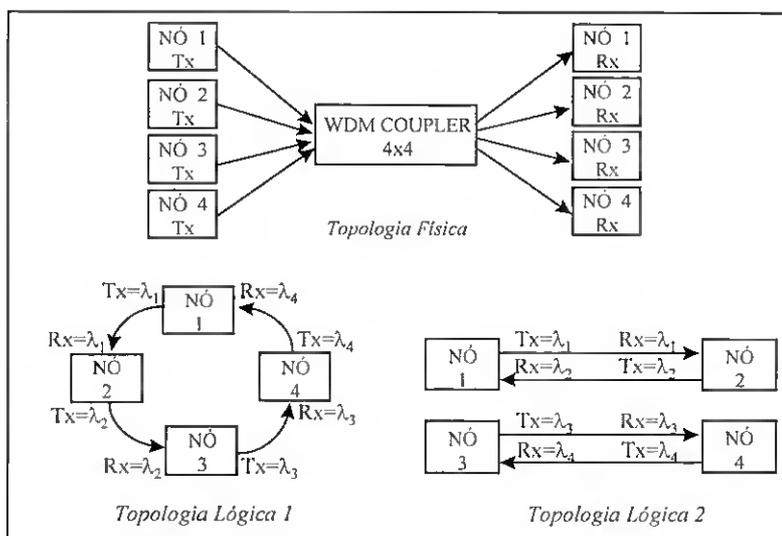


Fig. 2.4 - Conectividades lógicas para uma rede física.

Assim, numa rede WDM com emissores e receptores sintonizáveis, o algoritmo responsável por encaminhar o tráfego possui um grau de liberdade adicional: pode escolher e modificar a topologia de acordo, por exemplo, com as exigências de tráfego. Redes deste tipo irão necessitar de um protocolo de acesso ao meio capaz de controlar a alocação dos comprimentos de onda e de componentes ópticos de sintonização rápida para uma eficiente actualização das ligações da rede.

A característica fundamental destas redes é que, para  $N$  comprimentos de onda disponíveis, é possível processar em simultâneo  $N$  ligações. Caso o número de comprimentos de onda disponíveis seja maior que o número de nós na rede, então um nó pode estabelecer mais que uma ligação.

A grande vantagem das redes WDM de comprimento de onda ágil é a sua “transparência” relativamente a sinais de formato diferente. Como os sinais viajam no domínio óptico directamente entre os dois nós sem qualquer necessidade de conversão intermédia para o domínio eléctrico, o seu formato só depende dos nós envolvidos na comunicação. Deste modo, poderá encontrar-se numa mesma rede não só taxas de transmissão variadas, como também diferentes formatos de sinal.

A grande limitação destas redes torna-se visível quando se aumenta o número de nós da rede. Para se manter o funcionamento descrito terá que haver um aumento proporcional do número de comprimentos de onda disponíveis o que, por limitações físicas, não é possível fazer indiscriminadamente. Um aumento do número de comprimentos de onda traduz-se num menor espaçamento entre canais provocando um aumento do *crosstalk*<sup>1</sup> para níveis eventualmente intoleráveis. Além disso, seriam necessários filtros ópticos de alta resolução e todo um conjunto de componentes ópticos de difícil ou impossível implementação. Por estes motivos, estas redes estão limitadas a um número de nós entre 16 e 32.

### **2.2.2 Redes WDM de Comprimento de Onda Fixo**

No funcionamento atrás descrito é exigido um canal directo entre o emissor e o receptor para que haja transferência de informação. A esse funcionamento é dado o nome de *single-hop*. Assim, um nó que queira receber (enviar) informação dos restantes nós na rede terá que possuir tantos receptores (emissores) de comprimentos de onda afastados quantos os nós com que deseja comunicar, ou então, terá que possuir um receptor (emissor) sintonizável que deve sintonizar o seu comprimento de onda de acordo com a ligação que deseja estabelecer. Na primeira situação conseguem-se transmissões muito rápidas mas, em contrapartida, existe um

---

<sup>1</sup> Interferência de canais adjacentes provocada pela sobreposição do espectro.

investimento em equipamento bastante elevado. Na segunda, o nó emissor poderá ter que esperar muito tempo para enviar a sua informação se a concorrência for elevada.

Quando o número de nós numa rede começa a crescer, torna-se incomportável um funcionamento *single-hop*. Nestes casos é melhor optar-se por uma rede de comprimentos de onda fixos. Nestas redes, também chamadas de *multi-hop*, as conexões podem usar nós intermediários que recebem a informação num comprimento de onda e reenviam noutra até a informação chegar ao seu destino. Ou seja, a rede encontra-se parcialmente conectada, de forma a minimizar o equipamento e comprimentos de onda necessários, e, todos os pares de nós que desejam comunicar e não têm uma ligação directa podem usar nós intermédios. Estas redes têm necessariamente maiores atrasos de transmissão.

### **2.2.3 Reutilização de Comprimentos de Onda**

Os comprimentos de onda são um recurso escasso e limitado já que existem limitações não só ao número de comprimentos de onda a que um nó pode aceder mas também ao número de comprimentos de onda que podem ser acomodados numa rede. Por este motivo torna-se importante gerir eficientemente este recurso. Uma forma de atingir este objectivo passa pela sua reutilização, ou seja, pela possibilidade de se usar um mesmo comprimento de onda em diferentes partes da rede com vista à criação de vários canais independentes. O único requisito é a sua não sobreposição em qualquer segmento físico para evitar interferências.

### **2.2.4 Routing em Redes WDM**

O protocolo de encaminhamento (*routing*) pertence à camada de rede do modelo OSI<sup>2</sup> e tem como objectivo direccionar o tráfego de um nó fonte para um nó destino, de acordo com os objectivos de serviço da rede. Estes objectivos poderão ser, por exemplo, a minimização do

---

<sup>2</sup> Modelo usado para interconexão de sistemas abertos.

atraso dos pacotes, a minimização dos custos de equipamento ou a maximização do *throughput*.

Desta forma, o *routing* envolve duas actividades básicas:

- Determinação dos melhores caminhos entre pares de nós origem-destino.
- Transporte dos grupos de informação (pacotes).

O cumprimento destes objectivos poderá, no entanto, ser limitado por factores como a tecnologia de comutação, o volume de tráfego e serviços requeridos pela rede. É esta natureza multi-objectivos e multi-restrições do *routing* que torna a determinação de caminhos uma tarefa complexa nas redes de comunicações.

### **MÉTODOS TRADICIONAIS**

Os sistemas tradicionais utilizam técnicas de *routing* estáticas que se baseiam num estado esperado da rede. Torna-se claro que um esquema fixo de *routing*, determinado por um horário de tráfego previsto, não pode por exemplo, acomodar eficientemente situações de tráfego inesperadas. Quando um segmento falha, a robustez da rede fica comprometida já que isso representa a perda de ligação a uma parte da rede. Para que seja possível resolver situações anómalas como esta (redireccionando o tráfego por exemplo), os sistemas de comutação tradicionais possuem uma série de controlos para gestão da rede. Em geral, este controlo é feito pelo administrador do sistema só após ocorrida a falha ou congestionamento, tendo já havido perda de tráfego.

### **EVOLUÇÃO DE *ROUTING* FIXO PARA *ROUTING* DINÂMICO**

Com o aparecimento de novos serviços e tecnologias tornou-se evidente que só evoluindo de um *routing* fixo para um *routing* dinâmico seria possível melhorar a eficiência da rede. Este tornou-se, então, um tema sob forte investigação em todo o mundo.

Um sistema de *routing* dinâmico selecciona os caminhos que a informação irá percorrer baseado-se em informação do estado actual da rede (e não no seu estado previsto, como o *routing* fixo) na altura do pedido de tráfego.

#### ESTRATÉGIAS DE ROUTING DINÂMICO

Foram propostas e/ou implementadas várias estratégias de *routing* dinâmico [8][9]:

- **Routing Não Hierárquico Dinâmico (DNHR).** Usado na rede da AT&T de 1984 a 1991, trata-se de um sistema centralizado que utiliza um segmento de dados dedicado de 2400 bps para recolher e distribuir informação de sinalização e *routing* pelos nós da rede. Cada nó mantém uma tabela com caminhos alternativos, constituídos apenas por 2 segmentos, para a eventualidade do segmento directo entre a fonte e o destino estar indisponível. Existem cerca de 14 caminhos alternativos para cada segmento directo da rede. O DNHR divide o dia em 10 períodos diferentes cujo *routing* se baseia em dados históricos sobre o tráfego. Obviamente que o sucesso deste método depende de quão apuradas são as previsões do tráfego, ou seja, de qual o plano de encaminhamento de tráfego escolhido.
- **Routing Controlado Dinamicamente (DCR).** Esta abordagem dinâmica foi desenvolvida pela Bell Northern Research e Northern Telecom para funcionar com a sua família de comutadores. O DCR é um método de *routing* dependente do estado da rede que adapta os padrões de *routing* às condições da rede. É usado um processador de rede (PR) central para monitorizar as condições de tráfego e gerar recomendações de *routing* quase em tempo real. É usado um ciclo de 10 segundos onde ocorrem dois tipos de transferências de dados:

1. Cada comutador envia ao PR dados sobre o tráfego local que incluem: o número de segmentos desocupados, número de chamadas que usam segmento

directo e ocupação do CPU. Com informação sobre todos os nós da rede, o PR constrói uma visão global da rede.

2. O PR desenvolve e envia recomendações de *routing* para cada comutador.

Tal como o DNHR, o DCR assume uma arquitectura de rede não hierárquica. A diferença fundamental é que o DNHR se baseia em padrões de *routing* desenvolvidos em *off-line*, enquanto que o DCR controla o *routing* usando dados sobre o tráfego da rede recolhidos de 10 em 10 segundos.

- **Routing Alternativo Dinâmico (DAR).** Esta estratégia de *routing* adaptativa foi implementada na *British Telecom Public Switched-Trunk* e selecciona caminhos alternativos de forma estocástica. Essa selecção baseia-se em informação local sobre a carga nos segmentos de saída para determinar a viabilidade dos caminhos.
- **Routing de Rede em Tempo Real (RTNR).** Este método substituiu o DNHR na rede da AT&T em 1991 e foi desenhado para ser adaptativo em várias classes de serviços, o que o torna facilmente extensível ao transporte de tráfego B-ISDN. O RTNR possui também serviços de *routing* que facilitam a sua integração nas redes globais. Tal como os métodos anteriores, o RTNR primeiro tenta o caminho directo e a chamada só é aceite se houver um segmento disponível. Se tal não acontecer, o nó origem comunica com o nó destino através das facilidades centrais e rede CCS. O nó destino envia o status *busy-idle* de todos os segmentos que nele terminam e o nó origem compara-os com os status dos seus segmentos para encontrar o caminho com menos carga (*Least-Loaded Route*, LLR) até ao destino.

## EVOLUÇÃO DAS REDES DINÂMICAS

A introdução de novas tecnologias tem, sem dúvida, um impacto nas técnicas de *routing*. A expansão tecnológica permitirá aumentar a sofisticação dos algoritmos e parâmetros neles usados, permitindo que a capacidade do *routing* melhore.

O uso das fibras ópticas e o aumento da concentração de tráfego irá forçar os gestores de redes a examinar as técnicas de *routing* em uso. Actualmente, a ocorrência de uma falha num segmento pode ter um impacto desastroso, mesmo se for rapidamente solucionada. O *routing* dinâmico aparece como forma de minimizar os riscos associados a esta grande concentração de tráfego, permitindo ao mesmo tempo usar a capacidade das fibras disponibilizada por técnicas como a WDM.

## 2.3 TECNOLOGIA DE SUPORTE

A implementação de redes WDM está condicionada por dispositivos como filtros ópticos, lasers de comprimento de onda sintonizável estáveis, receptores sintonizáveis, conversores de frequência óptica, amplificadores ópticos, moduladores, etc. As redes WDM de comprimento de onda ágil, em particular, necessitam de emissores, receptores, filtros ópticos e conversores de frequência sintonizáveis. Embora alguns destes componentes já se encontrem no mercado, ainda possuem um custo elevado, sendo objecto de grande investigação a nível internacional.

### 2.3.1 Fibras Ópticas

As fibras ópticas são o meio de transmissão usado em comunicações ópticas. Estas, quando comparadas com outros meios de transmissão, oferecem baixa atenuação (0.2 dB/km) e forte imunidade a interferências electromagnéticas. Quando a luz se propaga no núcleo da fibra, podem existir diferentes distribuições dos campos eléctricos, responsáveis pela formação dos *modes*. Em geral, *modes* diferentes propagam-se na fibra a velocidades diferentes, resultando numa dispersão indesejável. Fibras que permitem a propagação de um único *mode* são chamadas *single-mode*. Fibras que permitem a propagação de múltiplos *modes* são chamadas *multi-mode*. No caso das fibras *single-mode* a propagação de um único *mode* é conseguida usando um diâmetro de núcleo de tal modo pequeno que os restantes *modes* são atenuados.

## FIBRAS SINGLE-MODE

A vantagem da propagação numa fibra *single-mode* é o facto de serem evitadas dispersões causadas por diferentes atrasos entre *modes*. Embora estas fibras tenham surgido recentemente (desde 1983) como meio viável para as comunicações ópticas, rapidamente se tornaram dominantes nas telecomunicações (as *multi-mode* são ainda encontradas em comunicações localizadas, LANs ), em grande parte devido ao facto de [10]:

1. Exibirem actualmente as maiores larguras de banda e menores perdas.
2. Possuírem uma qualidade de transmissão superior à de outros tipos de fibra devido à ausência de ruído modal.
3. Oferecerem capacidades de actualização para futuros serviços de grande largura de banda usando emissores e receptores mais rápidos ou técnicas de transmissão avançadas.
4. Serem compatíveis com a tecnologia óptica de integrados em desenvolvimento.

Os valores típicos de atenuação e dispersão para as janelas 1300 e 1550 nm são [11]:

Parâmetro	1300 nm	1550 nm
Atenuação	0,3 a 0,4 dB/km	0,15 a 0,25 dB/km
Dispersão	-2 a -5 ps/km.nm	15 a 18 ps/km.nm

Tabela 2.2 - Características da fibra *single-mode*.

### 2.3.2 Emissores Ópticos

Num sistema WDM, os emissores devem satisfazer os seguintes critérios:

1. **Largura de banda espectral (*linewidth*) estreita.** Para ser possível acomodar vários canais numa fibra, a separação entre canais deve ser tão pequena quanto possível. Isto só é possível se o *linewidth* do laser for pequeno, minimizando assim interferências entre canais.

2. **Gama de sintonização larga.** Este aspecto é particularmente importante para sistemas que usem emissores que possam ser sintonizados dinamicamente ao longo do tempo. Ao possuir uma larga gama de sintonização, o laser pode transmitir sinais no comprimento de onda apropriado.
3. **Sintonização rápida.** Num sistema dinâmico é importante minimizar o tempo perdido na passagem de um comprimento de onda de emissão para outro.
4. **Potência de saída elevada.** Se o sinal óptico de um emissor vai ser dividido por vários receptores, a potência óptica é um dos factores que determina o número de receptores que podem ser alcançados sem um uso excessivo de amplificadores ópticos responsáveis pela introdução de ruído de emissão espontânea.
5. **Elevada estabilidade nas frequências.** O laser deve ser capaz de manter à saída um comprimento de onda estável que não seja afectado por factores como a temperatura e envelhecimento.
6. **Chirp reduzido.** O *chirp* manifesta-se na mudança dinâmica do comprimento de onda de pico emitido pelo laser devido a variações do índice de refacção. Quando o laser é directamente modulado por um sinal de corrente, a densidade de portadoras varia provocando variações no índice de refacção. Por este motivo, quando o efeito de *chirp* for intolerável (longas distâncias em que o *chirp* associado à dispersão das fibras ópticas irá alongar os pulsos e, conseqüentemente, degradar a performance do sistema) a alternativa é utilizar modulação externa, o que exige um modulador em separado. A modulação directa, para além do efeito de *chirp*, também provoca o alargamento do *linewidth* do espectro do sinal óptico. No entanto, por motivos económicos, só é usada modulação externa quando é necessário modular uma fonte com um *linewidth* muito estreito ou quando se deseja uma performance superior à do primeiro método.

Na construção de lasers existe sempre um compromisso entre a velocidade de sintonização, e, *linewidth* e gama de frequências endereçáveis. A seguinte tabela mostra as características de lasers sintonizáveis.

Tecnologia	Gama de sintonização	Tipo de sintonização	<i>Linewidth</i>	Velocidade	Referência
Óptico-acústico	70 nm	discreta	?	3 μs	[12]
Electro-óptico	7 nm	discreta	60 kHz	100 μs	[12]
DFB 2-secções	3.3 nm	contínua	15 MHz	?	[12]
DFB 3-secções	2 nm	contínua	500 kHz	?	[12]
DFR 3-secções	4.4 nm	contínua	1.9 MHz	10 μs	[12]
VCF 3-secções	57 nm	discreta	?	?	[12]
DBR-LD	3 nm	contínua	?	60 ps	[13]
SSG-DBR	90 nm	discreta	?	5 ns	[14]

Tabela 2.3 - Características de lasers sintonizáveis.

### 2.3.3 Receptores Ópticos

Um receptor WDM é caracterizado pelos seguintes parâmetros:

- **Sensibilidade.** Potência média mínima à entrada do receptor que garante uma determinada probabilidade de erro.
- **Velocidade de sintonização.** Tempo que um receptor demora a sintonizar-se para um comprimento de onda.

Existem basicamente dois tipos de receptores ópticos: receptores de detecção directa (ou incoerente) e receptores coerentes, ilustrados na seguinte figura [12].

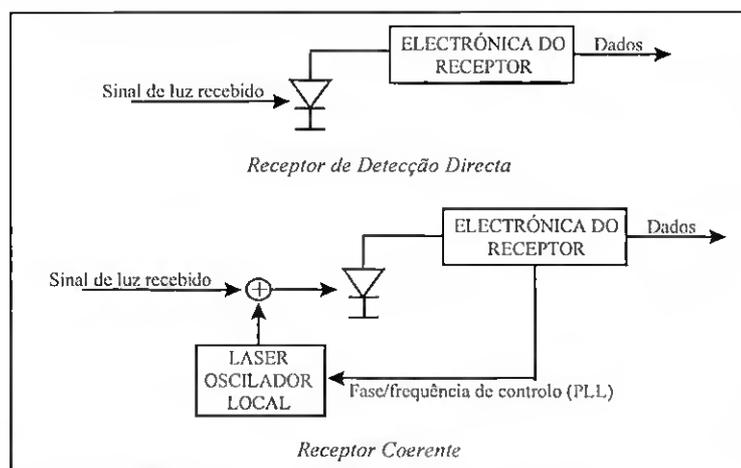


Fig. 2.5 - Estrutura dos receptores de detecção directa e coerentes.

## RECEPTORES DE DETECÇÃO DIRECTA

Nestes receptores a luz incidente é convertida em foto-corrente de forma proporcional à sua potência. Como só é detectada a potência da luz incidente, e não a sua frequência e fase, estes receptores são usados para detectar sinais modulados em amplitude [15]. Pode obter-se um receptor de detecção directa sintonizável colocando um filtro óptico sintonizável à entrada do receptor, permitindo assim que só um comprimento de onda chegue ao foto-díodo. Se o filtro óptico é sintonizável então o receptor também o é [16].

## RECEPTORES COERENTES

Estes receptores combinam o sinal de entrada com um sinal gerado localmente por um oscilador (*local oscillator laser*), antes de aplicá-lo ao foto-díodo. Assim, além da potência, estes detectam também a frequência e fase do sinal recebido. Quando a frequência do oscilador é igual à frequência do sinal de entrada, trata-se de um receptor homodino. Caso estas frequências sejam diferentes, trata-se de um receptor heteródino. Em geral, é mais fácil implementar detecção heteródina mas, em contrapartida, esta possui uma sensibilidade mais baixa. Isto porque, enquanto na detecção homodina, para se obter sincronização, as duas portadoras têm que ficar com a mesma frequência e uma pequena diferença de fase fixa, na detecção heteródina a sincronização necessita somente de assegurar que a diferença de frequências esteja dentro da banda da frequência intermédia (a relação de fase fixa não é necessária). A sincronização do receptor heteródino é por isso mais fácil [15]. Para se conseguir um receptor coerente sintonizável é usado um laser sintonizável como oscilador para seleccionar um dos comprimentos de onda recebidos, não sendo necessário qualquer processo de filtragem.

A detecção coerente tem várias vantagens sobre a detecção directa: pode ser usada para qualquer formato de modulação, é mais sensível e pode ser mais selectiva. Contudo, existem

alguns efeitos de deterioração que têm que ser combatidos para se conseguir estas vantagens, o que torna estes esquemas de detecção bastante mais complexos.

### 2.3.4 Filtros Ópticos

Estes dispositivos têm basicamente duas funções: eliminar o ruído, e seleccionar, ou deixar passar, um ou mais comprimentos de onda específicos. Numa rede WDM este é um componente chave uma vez que permite extrair um canal desejado de um grupo de canais multiplexados. A figura abaixo ilustra este princípio.

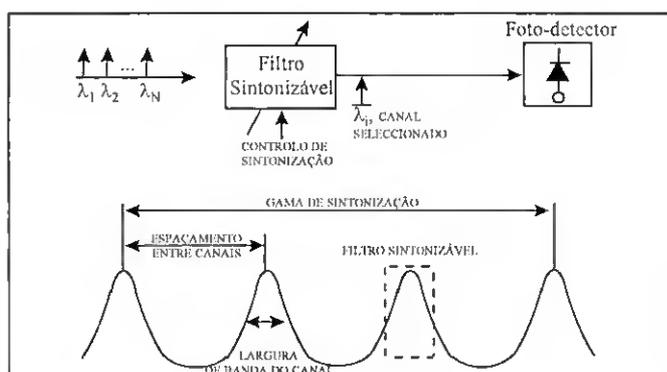


Fig. 2.6 - Selecção de um canal usando um filtro sintonizável.

Os parâmetros importantes associados aos filtros ópticos são:

- **Gama de sintonização.** Conjunto de comprimentos de onda abrangidos.
- **Número máximo de canais.** Número de comprimentos de onda seleccionáveis. Este número é determinado pela tecnologia usada no filtro sintonizável e é definido como a razão entre a gama total de sintonização e o espaçamento mínimo entre canais necessário para garantir uma degradação mínima de *crosstalk*.
- **Velocidade de sintonização.** Tempo de transição entre quaisquer dois comprimentos de onda.
- **Atenuação.** O sinal óptico seleccionado irá sofrer atenuação de potência devido a perdas nos filtros. Esta deve ser tão baixa quanto possível.

- **Não dependência da polarização.** A função de transferência deve ser independente do estado de polarização do sinal óptico.

A seguinte tabela sumariza os parâmetros importantes das várias tecnologias de filtros sintonizáveis [17].

Tecnologia	Gama de Sintonização (nm)	Largura de Banda (nm)	Número de Canais	Perdas (dB)	Velocidade de Sintonização	Aplicação
FPF-Single Stage	50	0.5	10	2	ms	Comutação de circuitos
FPF-Tandem	50	0.01	100	5	ms	Comutação de circuitos
Mach-Zehnder	5-10	0.01	100	5	ms	Comutação de circuitos
Electro-Óptico	10	1	10	≈5	ns	Comutação de pacotes
Óptico-Acústico	400	1	10	≈5	μs	Comutação de circuitos e pacotes longos
Semicondutor Activo	1-5	0.05	10	0 (ganho possível)	ns	Comutação de circuitos e pacotes
Brillouin-Fibra	10	<0.01	100	0 (ganho possível)	μs	Comutação de circuitos e pacotes longos

**Tabela 2.4** - Características de filtros ópticos sintonizáveis.

A filtragem de um comprimento de onda pode ser conseguida pelos seguintes mecanismos [18]:

1. Fenómenos interferométricos (filtros Fabry-Perot e Mach-Zehnder).
2. Acoplamento entre campos ópticos induzido por perturbações externas (filtros electro-ópticos e óptico-acústicos).
3. Amplificação ressonante nos lasers semicondutores.
4. Efeito não linear de SBS (*Stimulated Brillouin Scattering*).

**FABRY-PEROT**

Trata-se de uma cavidade ressonante selectiva que só transmite certos comprimentos de onda determinados pelo tamanho e índice de refração da cavidade. Para ajustar o tamanho da cavidade pode ser usado um cristal piezoeléctrico e uma fonte externa de tensão [15]. A alteração do comprimento de onda por variação do índice de refração é possível usando um

cristal electro-óptico [15]. Este filtro tem a vantagem de possuir baixas perdas por acoplamento mas tem o inconveniente de ser relativamente lento na sintonização.

#### MACH-ZEHNDER

Os filtros Mach-Zehnder envolvem interferência *feedforward* entre o sinal e uma versão atrasada do mesmo [15]. Têm muito boa resolução (fracção de angstrom) mas estão limitados na gama de frequências (nanómetros) e tempos de sintonização (milisegundos), requerendo tolerâncias de fabrico apertadas.

#### ELECTRO-ÓPTICO E ÓPTICO-ACÚSTICO

Estes filtros fazem uso de mecanismos de acoplamento entre campos ópticos usando efeitos electro-ópticos e óptico-acústicos respectivamente [15]. Os filtros electro-ópticos sintonizam-se rapidamente (nanosegundos) e têm uma largura de banda pequena (angstroms), mas a sua gama de sintonização é pequena. Os filtros óptico-acústicos têm uma resolução muito pobre mas uma boa gama de sintonização e tempos de sintonização razoavelmente rápidos (microsegundos).

#### AMPLIFICAÇÃO RESSONANTE NOS LASERS SEMICONDUTORES

Os lasers semicondutores fornecem filtragem espectral de forma inerente. Estes lasers, quando usados como amplificadores ópticos, amplificam só as frequências do sinal de entrada que coincidem com a estrutura amplificadora do laser. Isto faz com que, após atingido um nível de saturação específico, só alguns comprimentos de onda possam escapar da cavidade. A frequência ressonante pode variar por injeção de corrente ou variação de temperatura, resultando numa alteração do índice de refração e, logo, na alteração da frequência. Estes dispositivos fornecem ganho e selectividade.



**EFEITO NÃO LINEAR DE SBS (*STIMULATED BRILLOUIN SCATTERING*)**

Efeitos não lineares são normalmente factores que deterioram uma rede. Contudo, para fibras *single-mode*, o efeito de *Stimulated Brillouin Scattering* pode ser usado como um mecanismo de amplificação selectiva [10]. Estes têm muito boa resolução e bons tempos de sintonização mas estão limitados na gama de frequências.

**2.3.5 Multiplexadores/Desmultiplexadores Ópticos**

Os multiplexadores permitem combinar vários sinais de diferentes comprimentos de onda numa única fibra óptica. No receptor estes sinais terão que ser desmultiplexados ou filtrados antes da foto-deteccção.

As características essenciais destes dispositivos para um sistema WDM são:

1. **Baixas perdas.** Quando vários sinais são acoplados ou separados, é inevitável a existência de perda de potência. Em geral, quanto maior o número de entradas e/ou saídas maiores são as perdas.
2. **Número de *ports* elevado.** Desta forma, é permitido o acoplamento de um número elevado de sinais. Quanto maior o número de *ports*, maior o tamanho e complexidade do dispositivo.
3. **Igual desempenho para os vários comprimentos de onda.** É desejável que todos os comprimentos de onda tenham aproximadamente a mesma perda de potência. Isto permite o uso de receptores mais simples e resulta numa performance de deteção igual para os vários comprimentos de onda.
4. **Baixa dependência de polarização.** Em geral, um multiplexador dependente da frequência possui uma eficiência de combinação dependente da polarização. Isto pode dever-se à anisotropia do dispositivo. A dependência de polarização pode ser reduzida usando material isotrópico ou melhorando o seu desenho. Na separação dos

sinais a não dependência de polarização torna-se importante para manter a sintonização.

5. **FRS (*Free Spectral Range*) largo.** Duas portadoras separadas por um intervalo FSR são desmultiplexadas para a mesma saída (*grating*<sup>3</sup>) ou podem ambas passar por um filtro. Isto é, têm igual efeito de desmultiplexagem ou filtragem.
6. **Alta resolução.** É importante que o desmultiplexador ou filtro consiga diferenciar canais que estão próximos. Isto implica uma grande capacidade de difracção por parte de dispositivos que usam *grating* ou filtros com uma banda estreita. Este factor irá determinar o número de canais possíveis.

Um acoplador e um multiplexador diferem apenas no seu princípio de funcionamento. O multiplexador junta sinais de diferentes comprimentos de onda usando o princípio de *grating* ou interferometria de Mach-Zehnder<sup>4</sup> (*Mach-Zehnder interferometry*) enquanto que o acoplador se baseia no princípio de acoplamento de potência [15]. Assim, a saída de um multiplexador está fortemente dependente dos comprimentos de onda de entrada e, uma vez desenhado, não é possível alterar arbitrariamente a gama de comprimentos de onda usada. Os acopladores, ao contrário dos anteriores, possuem uma saída relativamente independente do comprimento de onda de entrada, o que os torna mais flexíveis. No entanto, estes têm a desvantagem de possuir maiores perdas que os multiplexadores. Numa rede WDM para que um nó possa sintonizar-se para diferentes canais de transmissão, o acoplamento deve ser insensível ao comprimento de onda e, por isso, é conveniente usar-se um acoplador.

A selecção de um comprimento de onda no receptor pode ser conseguida no domínio espacial, frequencial e polarizacional [15]. No primeiro caso os canais são separados para diferentes caminhos espaciais (*grating*). No domínio frequencial, é usado ou um filtro

---

<sup>3</sup> O acoplamento é feito com base na direcção da luz de entrada (domínio espacial).

<sup>4</sup> Coeficientes de acoplamento dependentes dos comprimentos de onda de entrada (domínio frequencial).

banda-passante óptico para suprimir os canais fora da banda, ou um amplificador óptico de banda estreita que amplifica somente o canal de interesse. Se todos os canais tiverem a mesma polarização, é ainda possível desmultiplexar um canal no domínio polarizacional. Isto é conseguido girando em 90° a polarização do canal desejado e usando depois um PBS (*Polarizing Beam Splitter*) para separação ou um polarizador (*polarizer*) para filtragem, isto é, só passam as ondas vibrando numa certa direcção.

### 2.3.6 Amplificadores Ópticos

Um outro componente fundamental num sistema WDM é o amplificador óptico. A amplificação é necessária para compensar perdas que ocorrem nos acopladores e separadores, e também para compensar a atenuação resultante da distância percorrida, regenerando assim o sinal. Este processo é feito directamente no domínio óptico, ao contrário dos tradicionais repetidores que exigem conversão óptico-electrónica.

As características a ter em conta no desenho de um amplificador óptico são [15]:

1. **Elevado ganho de potência.** Dependendo da potência de entrada é possível obter-se 10-30 dB.
2. **Alta eficiência de alimentação (*pumping*).** Um amplificador possui maior eficiência se, para um mesmo ganho, possuir uma menor potência externa.
3. **Largura de banda larga.** Deste modo, torna-se possível amplificar em simultâneo vários canais, facto bastante importante para redes WDM. Outra vantagem da grande largura de banda, é o facto de o ganho do amplificador ficar insensível aos comprimentos de onda do sinal incidente.
4. **Baixo efeito de saturação.** Faz com que o ganho do amplificador decresça quando a potência incidente aumenta. Assim, é desejável um baixo efeito de saturação.

5. **Baixa dependência de polarização.** Isto porque o ganho de potência está também dependente da polarização da luz incidente.
6. **Baixo ruído.** O ruído é provocado pela emissão espontânea de fótons dentro do amplificador.
7. **Baixo *crosstalk*.** Quando um amplificador é usado para amplificar vários canais em simultâneo é importante garantir que não há interferências.
8. **Baixas perdas por acoplamento.** Estas perdas ocorrem quando um amplificador é usado num troço de fibra óptica.

Existem dois tipos de amplificadores ópticos importantes: amplificadores semicondutores (SOAs) e amplificadores de fibra dopados de Erbium (EDFAs). Os EDFAs dada a sua linearidade, elevada potência de saturação, baixa figura de ruído e menor sensibilidade à polarização dos sinais, são a solução preferida quando só é necessária amplificação. Os SOAs, dada a sua natureza intrinsecamente não linear, são os escolhidos para aplicações em que é necessária alguma forma de processamento do sinal, como é o caso dos filtros, comutadores, conversores de frequência e moduladores ópticos.

## 2.4 EXEMPLOS ILUSTRATIVOS DE REDES WDM

### 2.4.1 Rede Rainbow

A rede Rainbow é uma rede metropolitana proposta pela IBM Research, desenhada para cobrir uma área de 25 km. Esta rede suporta a interconexão comutada por circuito de até 32 nós, todos a transmitirem a uma taxa de 300 Mb/s, permitindo uma capacidade total da rede de 9.6 Gb/s. As conexões são “transparentes” relativamente às outras e, por isso, podem usar taxas de dados e protocolos diferentes [19][20].

#### ARQUITECTURA DA REDE

A Rainbow usa uma arquitectura *broadcast-and-select*, como mostra a figura.

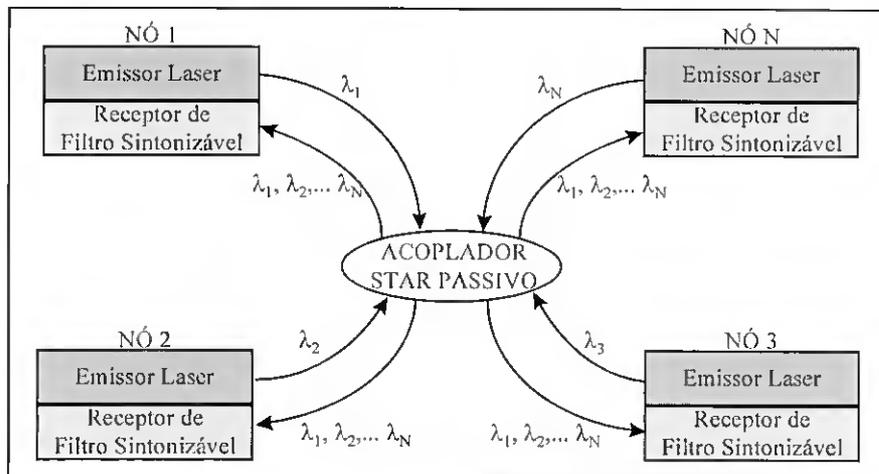


Fig. 2.7 - Arquitectura da rede Rainbow.

Cada estação está equipada com um emissor óptico que emite luz num comprimento de onda fixo diferente dos outros emissores na rede. Uma fibra liga a estação a um *hub*, que é um acoplador *star* passivo com tantos *ports* quantas as estações de trabalho. O acoplador combina as transmissões das diferentes estações e à saída é obtido aproximadamente  $1/N$  da potência óptica de cada emissor. Do acoplador sai uma fibra para cada estação havendo assim um *broadcast* da informação. Em cada estação existe um receptor sintonizável que selecciona o comprimento de onda desejado.

Os emissores são lasers *distributed-feedback* (DFB) na janela  $1.5 \mu\text{m}$  e foram escolhidos pela sua largura espectral estreita, permitindo um espaçamento entre canais estreito. Estes são modulados directamente usando modulação OOK (*On-Off Keying*). A potência média do laser para a fibra é aproximadamente  $-3 \text{ dBm}$  [19].

Em cada receptor é usado um filtro Fabry-Perot de Fibra de sintonização piezoelétrica. O tempo que o filtro leva a sintonizar-se sobre a gama desejada de comprimentos de onda e fixar-se numa delas é cerca de  $25 \text{ ms}$ . O filtro tem uma gama de sintonização de aproximadamente  $0.4 \text{ nm}$  ( $\cong 50 \text{ GHz}$ ) e um FSR de aproximadamente  $40 \text{ nm}$  ( $\cong 5000 \text{ GHz}$ ) [19].

## O PROTOCOLO

Devido ao tempo de sintonização lento, só é permitida comutação por circuitos. É então necessário um protocolo para estabelecer e desligar conexões. Basicamente, quando uma estação *A* deseja comunicar com outra estação *B*, esta tem que fazer com que a estação *B* sintonize o seu filtro para o comprimento de onda usado pela estação *A* para transmissão. Para este efeito foi implementado um protocolo simples de busca circular [19].

### 2.4.2 Rede MONET

O programa MONET (*Multiwavelength Optical NETworking*) foi estabelecido em Dezembro de 1994 e surgiu com o objectivo de definir e demonstrar a melhor forma de se atingir uma rede óptica multicanal de grande escala que seja “transparente”, de confiança, de custo viável e de performance e capacidade elevada, considerando os aspectos de arquitecturas de rede, tecnologias avançadas, gestão de redes, e negócios. Os parceiros neste projecto são a AT&T, Bellcore, Lucent Technologies, Bell Atlantic, BellSouth, Pacific Telesis, and SBC, em colaboração com a National Security Agency (NSA) e Naval Research Laboratory (NRL), e parcialmente suportado pela US DARPA [21]. O projecto iniciou-se em 1996 e tem final previsto para 1999. Neste momento está estipulado um conjunto de definições necessárias, havendo ainda várias tecnologias em estudo para escolha de um rumo a seguir.

### CENTROS DE TESTE

A tecnologia e conceitos da rede MONET estão a ser demonstrados num conjunto de redes experimentais que inclui a *New Jersey Network*, um segmento de fibra de longa distância, e a *Washington Area Network*.

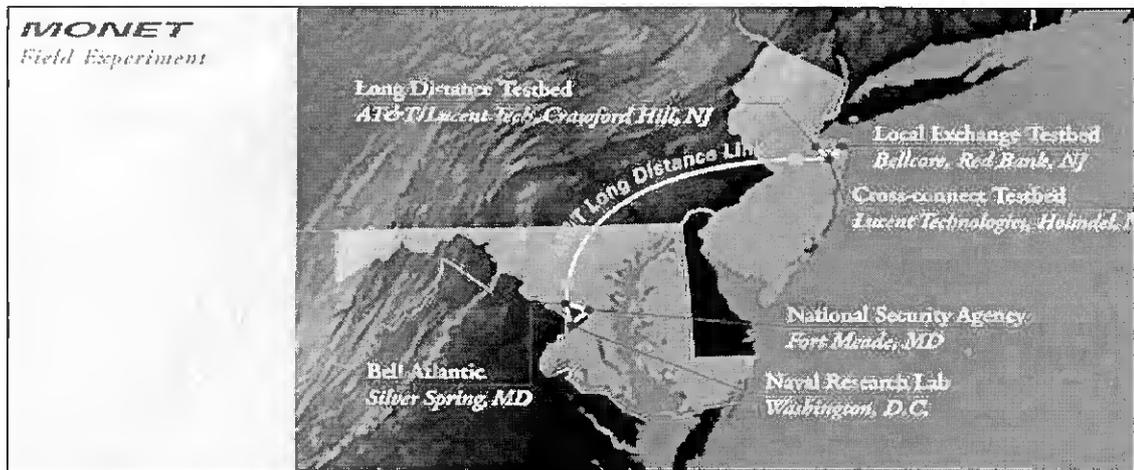


Fig. 2.8 - Rede MONET.

A *New Jersey Network* é constituída por três centros de teste interligados (AT&T - Crawford Hill Long Distance Testbed, Lucent Technologies - Holmdel Crossconnect Testbed e Bellcore - Navesink Local Exchange Testbed) e a *Washington Area Network* interliga o laboratório da Bell Atlantic (Silver Spring), o Naval Research Lab (Washington, DC) e a National Security Agency (Fort Meade, MD) [21]. Os centros de teste de interligação local serão usados para demonstrar arquiteturas WDM em anel, estrela e malha (*mesh*) e um sistema de gestão de redes distribuído. O trecho de longa distância permitirá demonstrar a performance de conexões de longa distância.

## 2.5 SUMÁRIO

Os sistemas WDM permitem um aumento drástico das capacidades de transmissão. Esta técnica permite embutir uma conectividade lógica numa rede física que pode ser alterada dinamicamente em resposta, por exemplo, a variações de tráfego. As redes WDM *single-hop* tem a vantagem de possuir “transparência” relativamente a sinais de formato diferente mas, por limitações físicas, estão condicionadas no número de nós. A alternativa são redes WDM *multi-hop* onde pode usar-se nós intermediários. Estas possuem, no entanto, maiores atrasos. As grandes capacidades actualmente disponibilizadas por sistemas como estes exigem técnicas de *routing* dinâmico. Foram discutidos os componentes fundamentais presentes em redes WDM e

mencionadas as suas características essenciais. No final do capítulo foram analisados exemplos ilustrativos de redes WDM Rainbow e MONET actualmente em implementação e estudo.

### 3. OPTIMIZAÇÃO DE REDES *MULTI-HOP*

#### 3.1 INTRODUÇÃO

Numa rede WDM, um modelo de ligação entre nós é definido pela sintonização de emissores e receptores para comprimentos de onda específicos que podem ser alterados dinamicamente. Desta forma, e ao contrário das redes tradicionais, o algoritmo de encaminhamento para redes WDM possui um grau de liberdade adicional: pode escolher, e até modificar a topologia. É possível, portanto, adaptar a conectividade lógica ao tráfego (que pode não ser uniforme) e/ou contornar falhas na rede.

Um aspecto a ter em conta nas redes WDM é o facto de os canais disponíveis (comprimentos de onda) serem partilhados pelos vários nós. Para aceder à rede um nó necessita que lhe seja alocado um comprimento de onda. Para que a transmissão esteja assegurada, o nó destino terá que sintonizar-se para o mesmo comprimento de onda do nó origem, ou então terá que existir um algoritmo de *routing* eficiente capaz de encaminhar os dados através de nós intermédios até ao seu destino. O último modelo de funcionamento é chamado de *multi-hop*. Nestas redes, a atribuição de comprimentos de onda aos nós interlocutores, permite identificar uma topologia lógica sobre a rede física e pode ser usada para providenciar tanto comutação de pacotes como comutação de circuitos. Neste estudo serão consideradas redes *multi-hop* com comutação de pacotes.

As redes *multi-hop* são especialmente adequadas a situações de tráfego estacionário durante períodos relativamente longos, não sendo exigida uma sintonização rápida dos emissores e/ou receptores. No entanto, caso estes sejam sintonizáveis, é possível modificar a topologia lógica da rede de acordo com as variações de tráfego, mas numa escala de tempo que não ultrapasse os limites físicos de sintonização dos dispositivos.

Neste capítulo começa-se por formular, primeiro de modo informal, o problema de optimização para redes *multi-hop*, passando-se depois a uma análise mais detalhada. É apresentada a forma desagregada e agregada de abordar o problema. No final são analisadas duas alternativas de resolução.

### 3.2 PARÂMETROS DE DESEMPENHO

O desempenho de uma rede WDM *multi-hop* com comutação por pacotes é caracterizada essencialmente pelo *throughput*, quantidade de tráfego que a rede consegue transmitir, e atraso sofrido pelos pacotes desde a origem até ao destino. Uma rede WDM terá melhor desempenho quanto maior for o *throughput* e quanto menor for o atraso sofrido pelos pacotes. Do ponto de vista da implementação física da rede, e considerando que cada nó é constituído por um emissor e receptor sintonizados para comprimentos de onda fixos, o desempenho da rede está limitado pelo número de comprimentos de onda disponíveis. Deste modo podemos definir o problema da seguinte forma:

*Seleccionar um modelo de ligação dos nós (topologia lógica) e fazer a distribuição do tráfego pelos segmentos do modelo adoptado (encaminhamento), de forma a optimizar um determinado objectivo.*

No que diz respeito ao objectivo a optimizar, é razoável pensar em minimizar:

- Congestionamento na rede.
- Atraso médio dos pacotes.
- Número de comprimentos de onda necessários.

Tendo como objectivo maximizar o *throughput*, optou-se neste trabalho por adoptar o primeiro ponto (congestionamento na rede) como característica a minimizar. Caso se optasse pela segunda hipótese, a melhor solução seria tornar a topologia lógica igual à topologia física e depois encaminhar o tráfego pelo caminho mais curto [22]. Quanto ao número de comprimentos de onda, as suas implicações serão consideradas nos algoritmos heurísticos apresentados.

### 3.3 FORMULAÇÃO DO PROBLEMA DE OPTIMIZAÇÃO

A abordagem escolhida assume que tanto a topologia física como a distribuição relativa do tráfego, são conhecidas *à priori*. O objectivo é desenhar a topologia lógica e o algoritmo de encaminhamento de forma a minimizar o congestionamento na rede, considerando os seguintes aspectos:

1. **Degree lógico máximo.** Cada nó só pode ser fonte e destino de um número limitado de segmentos lógicos. Isto é determinado pelo número de emissores e receptores existentes em cada nó.
2. **Número de comprimentos de onda disponíveis.** Impõe um limite ao número de segmentos lógicos que podem ser estabelecidos.
3. **Atraso máximo.** O atraso associado a um caminho possui duas componentes: atrasos de *queue* e atrasos de propagação nos segmentos. Para redes WAN, o atraso de propagação total é a componente dominante e, normalmente, os atrasos de *queue* são ignorados. Nas redes LAN acontece o inverso, isto é, a componente mais importante é o total de atrasos registados nas *queues* e o atraso de propagação é normalmente ignorado. A distribuição de tráfego poderá considerar um atraso máximo permitido.

Para uma rede com  $N$  nós, consideremos a matriz de tráfego  $A$  de dimensão  $N \times N$ :

$$A = (\lambda^{sd}) \quad s, d=1, \dots, N \quad [3.1]$$

onde  $\lambda^{sd}$  é a taxa de pacotes que parte do nó  $s$  e que se destina ao nó  $d$ . Consideremos ainda que estão disponíveis  $W$  comprimentos de onda e que  $\Delta_i$  representa o *degree* da topologia lógica a ser gerada. O objectivo será criar uma topologia lógica e um encaminhamento do tráfego sobre essa topologia de forma a minimizar o congestionamento, estando sujeito a algumas restrições.

**Objectivo:** min *congestionamento*

**Sujeito a:**

- Número de comprimentos de onda usados é no máximo  $W$ .
- Dois segmentos lógicos que partilham o mesmo segmento físico têm que ter comprimentos de onda diferentes.
- Todos os nós têm  $\Delta_r$  segmentos lógicos de entrada e  $\Delta_s$  segmentos lógicos de saída.
- Para cada par fonte-destino, o atraso na transmissão não deverá exceder um determinado limite.
- Os nós devem reenviar todo o tráfego que não lhes é dirigido.
- O fluxo num segmento lógico não poderá exceder a capacidade desse segmento.

Depois desta análise informal iremos analisar as restrições detalhadamente.

### 3.3.1 Análise Detalhada das Restrições do Problema de Optimização

#### TRÁFEGO

Como já foi referido, a matriz de tráfego  $N \times N$  será representada por  $A = (\lambda^{sd})$  onde  $\lambda^{sd}$  é a taxa de pacotes que parte do nó  $s$  e que se destina ao nó  $d$ . Se designarmos um segmento lógico por  $(i,j)$ , cada elemento da matriz de tráfego pode ser visto como:

$$\lambda^{sd} = \sum_{i,j} \lambda_{ij}^{sd} \quad \forall s, d \quad [3.2]$$

onde  $\lambda_{ij}^{sd}$  é o tráfego pertencente ao par origem-destino  $(s,d)$  que flui no segmento lógico  $(i,j)$ . Ou seja, o tráfego total de um par  $(s,d)$  é igual à soma de todas as parcelas de tráfego, pertencentes a esse par, que estão distribuídas pelos vários segmentos lógicos da rede. Da mesma forma, o fluxo total em qualquer segmento lógico  $(i,j)$  é obtido pela expressão:

$$\lambda_{ij} = \sum_{s,d} \lambda_{ij}^{sd} \quad \forall i, j \quad [3.3]$$

isto é,  $\lambda_{ij}$  constitui a soma dos fluxos de tráfego de todos os pares origem-destino  $(s,d)$  que fluem no segmento lógico  $(i,j)$ .

**CONGESTIONAMENTO**

Uma forma de abordar o problema da minimização do congestionamento numa rede é encaminhar o tráfego de maneira a minimizar o fluxo máximo em qualquer segmento lógico [22][23][24]. Isto pode ser conseguido minimizando o fluxo total no segmento mais sobrecarregado:

$$\min \lambda_{max}$$

$$\text{onde } \lambda_{max} = \max \lambda_{ij} = \max \sum_{s,d} \lambda_{ij}^{s,d} \quad [3.4]$$

e obrigando a que o fluxo nos outros segmentos não ultrapasse esse valor, ou seja:

$$\lambda_{ij} \leq \lambda_{max} \quad \forall i, j \quad [3.5]$$

Desta forma, uma vez otimizado o encaminhamento, o tráfego ficará distribuído o mais uniformemente possível.

**REENCAMINHAMENTO DO TRÁFEGO**

Uma vez que estão a ser consideradas redes *multi-hop*, os nós devem reenviar todo o tráfego que não lhes é dirigido, ou seja:

$$\sum_j \lambda_{ij}^{s,d} - \sum_j \lambda_{ji}^{s,d} = \begin{cases} \lambda^{s,d} & , \text{ se } s=i \\ -\lambda^{s,d} & , \text{ se } d=i \\ 0 & , \text{ caso contrário} \end{cases} \quad \forall s, d, i \quad [3.6]$$

A figura seguinte poderá ajudar a compreender melhor a expressão anterior.

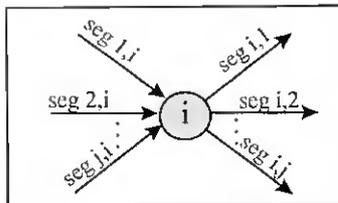


Fig. 3.1 - Nó *i* exemplo.

Se considerarmos, para um nó *i* qualquer, somente o tráfego pertencente ao par origem-destino (*s,d*), a diferença entre o tráfego que sai (nos segmentos: (*i,1*), (*i,2*),..., (*i,j*)) e o tráfego que entra (nos segmentos: (*1,i*), (*2,i*),..., (*j,i*)) é igual a:

- $\lambda^{s,d}$ , caso o nó *i* seja a fonte do tráfego (*i=s*).
- $-\lambda^{s,d}$ , caso o nó *i* seja o destino do tráfego (*i=d*).
- 0, c. contrário. Isto é, o tráfego que entra no nó *i* tem que ser igual ao tráfego que sai.

### LIMITE NO ATRASO DE PACOTES

Como já foi referido, é razoável ter como objectivo quer a minimização do congestionamento na rede quer a minimização do atraso médio dos pacotes. Algumas propostas optam por minimizar o congestionamento e ignoram o atraso, como em [23] e [24], outras preferem minimizar o atraso e ignoram o congestionamento. Em [22] é proposto minimizar o congestionamento na rede, restringindo ao mesmo tempo o atraso associado a cada par fonte-destino. Ignorar as restrições de atraso pode resultar em topologias com longos atrasos mas, por outro lado, impôr essas restrições resulta num pequeno aumento do congestionamento. A restrição de atraso proposta em [22] é a seguinte:

$$\sum_{i,j} \lambda_{ij}^{sd} d_{ij} \leq \lambda^{sd} \alpha d^{\max} \quad \forall s, d \quad [3.7]$$

onde  $d_{ij}$  é o atraso associado ao segmento lógico  $(i,j)$  (soma dos atrasos dos segmentos físicos usados pelo segmento lógico),  $d^{\max}$  é o maior atraso existente entre qualquer par  $(s,d)$  da rede, e  $\alpha$  é um número real. A expressão diz que o atraso associado a qualquer par  $(s,d)$  está limitado a  $\alpha$  vezes o pior atraso. Assim,  $\alpha d^{\max}$  constitui o atraso máximo permitido.

### CAPACIDADE DO SEGMENTO

É ainda necessário garantir que o fluxo de tráfego que percorre um segmento lógico não excede a sua capacidade. Uma vez que o objectivo é minimizar o fluxo total no segmento mais sobrecarregado,  $\lambda_{\max}$ , e está garantido que o fluxo total nos outros segmentos não excede  $\lambda_{\max}$ , só será necessário assegurar que:

$$C \geq \lambda_{\max} \quad [3.8]$$

isto é, que a capacidade dos segmentos lógicos é superior ao fluxo do segmento mais sobrecarregado,  $\lambda_{\max}$ . Esta restrição não será incluída na formulação do problema uma vez que a optimização garante que o fluxo nos segmentos será o menor possível.

**DEGREE DA TOPOLOGIA LÓGICA**

Seja  $b_{ij}$  uma variável binária que, para cada segmento lógico possível, toma o valor:

- 1, caso esse segmento lógico  $(i,j)$  exista.
- 0, caso contrário.

Então, o *degree* da topologia lógica pode ser expressado por:

$$\sum_i b_{ij} = \Delta_i \quad \forall j \quad [3.9]$$

$$\sum_j b_{ij} = \Delta_i \quad \forall i \quad [3.10]$$

Estas duas fórmulas indicam que qualquer nó da rede deverá possuir  $\Delta_i$  segmentos lógicos de entrada e  $\Delta_i$  segmentos lógicos de saída respectivamente. Para que circule tráfego somente nos segmentos lógicos criados deve ser garantido que:

$$\lambda_{ij}^{sd} \leq b_{ij} \lambda^{sd} \quad \forall i, j, s, d \quad [3.11]$$

Ou seja, o tráfego pertencente a um par  $(s,d)$  que flui num possível segmento lógico  $(i,j)$  deverá ser igual a 0, caso esse segmento não exista ( $b_{ij}=0$ ); ou deverá ser inferior ao tráfego total pertencente a esse par, caso contrário ( $b_{ij}=1$ ).

**FORMULAÇÃO GLOBAL**

Uma vez analisados os aspectos a considerar na formulação do problema e ignorando o número de comprimentos de onda disponíveis, este é apresentado na globalidade sob a forma de um problema MILP<sup>5</sup>.

<sup>5</sup> Um problema é considerado MILP (*Mixed Integer Linear Programming*) quando se trata de um problema linear onde estão envolvidas variáveis contínuas e inteiras.

<b>Objectivo:</b>	$\min \lambda_{max}$	
<b>Sujeito às restrições de:</b>		
Congestionamento:	$\lambda_{ij} \leq \lambda_{max}$	$\forall i, j$
Reencaminhamento:	$\sum_j \lambda_{ij}^{sd} - \sum_j \lambda_{ji}^{sd} = \begin{cases} \lambda^{sd} & , \text{ se } s=i \\ -\lambda^{sd} & , \text{ se } d=i \quad \forall s, d, i \\ 0 & , \text{ c. contrário} \end{cases}$	
Limite de atraso:	$\sum_{i,j} \lambda_{ij}^{sd} d_{ij} \leq \lambda^{sd} \alpha d^{max}$	$\forall s, d$
Degree da topologia lógica:	$\sum_i b_{ij} = \Delta_\ell$	$\forall j$
	$\sum_j b_{ij} = \Delta_\ell$	$\forall i$
	$\lambda_{ij}^{sd} \leq b_{ij} \lambda^{sd}$	$\forall i, j, s, d$
	$\lambda_{ij}^{sd}, \lambda_{ij}, \lambda_{max} \geq 0 \text{ e } b_{ij} \in \{0,1\}$	$\forall i, j, s, d$

[3.12]

A forma como o problema foi formulado permite que o tráfego pertencente ao par  $(s,d)$  possa ser encaminhado por vários caminhos.

Caso seja permitido atribuir vários segmentos lógicos a um par de nós, a única alteração necessária seria passar a designar um segmento lógico por  $(i,j,k)$ ,  $k=1,2,\dots, \Delta_\ell$ , e alterar as variáveis de forma a considerar mais esta dimensão.

### 3.3.2 Forma Agregada do Problema

A formulação anterior do problema é normalmente referida como “forma desagregada” porque cada par  $(s,d)$  é considerado um objecto (*commodity*). O problema pode ser formulado também na chamada “forma agregada” onde um objecto inclui todo o tráfego que sai de uma fonte. O problema formulado deste modo é bem mais tratável que o anterior visto possuir menos variáveis e restrições. Considerando  $\lambda^s = \sum_d \lambda^{sd}$  o tráfego total com origem em  $s$ , e  $\lambda_j^s$  o tráfego total com origem em  $s$  que flui no segmento lógico  $(i,j)$ , o problema agregado toma o seguinte aspecto:

<b>Objectivo:</b>	$\min \lambda_{max}$	
<b>Sujeito às restrições de:</b>		
Congestionamento:	$\lambda_{ij} \leq \lambda_{max}$	$\forall i, j$
Reencaminhamento:	$\sum_j \lambda_{ij}^s - \sum_j \lambda_{ji}^s = \begin{cases} \lambda^s & , \text{ se } s=i \\ -\lambda^s & , \text{ se } s \neq i \end{cases}$	$\forall s, i$
Limite de atraso:	$\sum_{i,j} \lambda_{ij}^s d_{ij} \leq \lambda^s \alpha d^{max}$	$\forall s$
Degree da topologia lógica:	$\sum_i b_{ij} = \Delta_\ell$	$\forall j$
	$\sum_j b_{ij} = \Delta_\ell$	$\forall i$
	$\lambda_{ij}^s \leq b_{ij} \lambda^s$	$\forall i, j, s$
	$\lambda_{ij}^s, \lambda_{ij}, \lambda_{max} \geq 0 \text{ e } b_{ij} \in \{0,1\}$	$\forall i, j, s$

[3.13]

Em [24] mostrou-se que a forma agregada simplifica o problema tornando a sua resolução mais rápida mas, em contrapartida, os valores de congestionamento obtidos não coincidem com os da forma desagregada, sendo sistematicamente maiores.

### 3.4 HIPÓTESES DE RESOLUÇÃO

#### 3.4.1 Abordagem Global

A dificuldade de um problema de optimização é maior quando algumas ou todas as variáveis envolvidas têm que tomar valores inteiros (ou somente 0 e 1, representando decisões Sim ou Não). Por este motivo, os algoritmos usados para resolver estes problemas são geralmente muito menos eficientes que os usados para problemas lineares onde essa restrição não é imposta (ex: Simplex). Os factores mais importantes que determinam o tempo de computação são o número de variáveis inteiras e a estrutura do problema. Por exemplo, para um número fixo de variáveis inteiras, a introdução de uma variável contínua pode não aumentar substancialmente o tempo de computação [25]. Por este motivo, e pelo grande número de variáveis envolvidas, o problema MILP anterior torna-se intratável, mesmo para redes

pequenas. Em [22] é usado o problema na forma *LP-relaxation*<sup>6</sup>, com uma restrição adicional mas, embora os valores de congestionamento apresentados sejam bons, o tempo de computação continua a ser muito elevado.

Os algoritmos disponíveis para resolução de problemas deste tipo baseiam-se normalmente na técnica *Branch-and-Bound*.

### 3.4.2 Abordagem Parcial

Outra forma de tornar o problema exequível é dividi-lo em dois subproblemas: desenho da topologia lógica e encaminhamento do tráfego.

#### DESENHO DA TOPOLOGIA LÓGICA

Este trata-se de um problema IP (*Integer Programming*). Os algoritmos para resolução destes problemas baseiam-se normalmente na técnica *Branch-and-Bound*. Existem também outras abordagens propostas recentemente, existindo ainda muita investigação nesta área.

Os problemas IP reais são por vezes tão grandes que nem mesmo os algoritmos mais recentes os podem resolver. Nestes casos, é comum usar o correspondente problema na forma *LP-relaxation* e depois arredondar a solução ótima para uma solução inteira. Esta abordagem é muitas vezes indesejável porque pode ser difícil (ou impossível) encontrar uma solução inteira, e a solução encontrada pode estar longe da ótima, o que é particularmente verdade para variáveis binárias [25].

Uma alternativa possível, onde não se colocam estas dificuldades, é usar algoritmos heurísticos<sup>7</sup>. Estes são relativamente rápidos e, mesmo para grandes problemas IP, geralmente encontram boas soluções que, embora não sendo ótimas, são geralmente melhores que as encontradas por arredondamento [25]. Esta foi a abordagem adoptada neste trabalho.

<sup>6</sup> O problema é idêntico mas a imposição de números inteiros é retirada.

<sup>7</sup> São desenhados intuitivamente e não garantem uma solução ótima. São usados quando as abordagens que encontram uma solução ótima são demasiado morosas.

ENCAMINHAMENTO DO TRÁFEGO

Uma vez determinada a topologia lógica, é possível fixar os valores de  $b_{ij}$  no problema MILP, tornando-o um problema de programação linear (LP) com a finalidade de encaminhar o tráfego na rede e, desta feita mais tratável. O problema de encaminhamento resume-se então ao seguinte problema LP:

<b>Objectivo:</b>	$\min \lambda_{max}$	
<b>Sujeito às restrições de:</b>		
Congestionamento:	$\lambda_{ij} \leq \lambda_{max}$	$\forall i, j$
Reencaminhamento:	$\sum_j \lambda_{ij}^{sd} - \sum_j \lambda_{ji}^{sd} = \begin{cases} \lambda^{sd} & , \text{ se } s=i \\ -\lambda^{sd} & , \text{ se } d=i \quad \forall s, d, i \\ 0 & , \text{ c. contrário} \end{cases}$	[3.14]
Limite de atraso:	$\sum_{i,j} \lambda_{ij}^{sd} d_{ij} \leq \lambda^{sd} \alpha d_{min}^{sd}$	$\forall s, d$
	$\lambda_{ij}^{sd}, \lambda_{ij}, \lambda_{max} \geq 0$	$\forall i, j, s, d$

Repare-se que a restrição de atraso utiliza  $d_{min}^{sd}$ , menor atraso entre  $s$  e  $d$ , em vez de  $d^{max}$ . O atraso permitido a um par  $(s, d)$  está limitado a  $\alpha$  vezes o menor atraso possível,  $\alpha \geq 1$ . Ou seja, quanto mais  $\alpha$  se aproxima de 1 mais tráfego terá que fluir no caminho com menor atraso. Esta nova expressão permite impor um atraso máximo diferente para cada par  $s-d$ .

A sua forma agregada fica com o seguinte aspecto:

<b>Objectivo:</b>	$\min \lambda_{max}$	
<b>Sujeito às restrições de:</b>		
Congestionamento:	$\lambda_{ij} \leq \lambda_{max}$	$\forall i, j$
Reencaminhamento:	$\sum_j \lambda_{ij}^s - \sum_j \lambda_{ji}^s = \begin{cases} \lambda^s & , \text{ se } s=i \\ -\lambda^s & , \text{ se } s \neq i \quad \forall s, i \end{cases}$	[3.15]
Limite de atraso:	$\sum_{i,j} \lambda_{ij}^s d_{ij} \leq \lambda^s \alpha d_{min}^s$	$\forall s$
	$\lambda_{ij}^s, \lambda_{ij}, \lambda_{max} \geq 0$	$\forall i, j, s$

### 3.4.3 Complexidade das Possíveis Abordagens

Para uma análise da complexidade de ambas as abordagens (global e parcial), consideremos a seguinte rede física usada em [22]. Os números associados aos segmentos físicos são valores relativos que representam os respectivos atrasos de propagação.

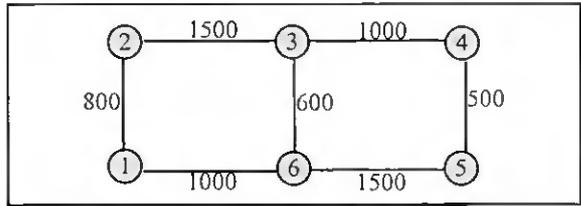


Fig. 3.2 - Topologia física exemplo.

Usando a abordagem global apresentada sob a forma de um problema MILP, que tem como objectivo desenhar a topologia lógica e encaminhar o tráfego, o número de variáveis e restrições envolvido seria:

Formulação	Nº Variáveis	Nº Restrições
Desagregada	900	1151
Agregada	180	263

Tabela 3.1 - Número de variáveis e restrições da rede exemplo usando abordagem global.

Para a abordagem parcial, consideremos que foi usado um algoritmo heurístico que, de acordo com os seus critérios, gerou a seguinte topologia lógica.

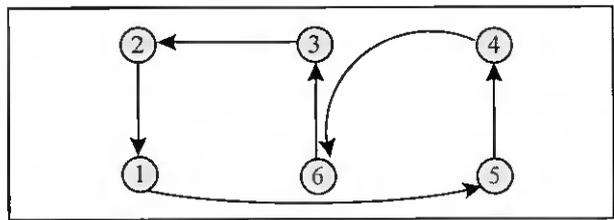


Fig. 3.3 - Topologia lógica exemplo.

O número de variáveis e restrições do problema LP, encarregue de encaminhar o tráfego, seria:

Formulação	Nº Variáveis	Nº Restrições
Desagregada	90	155
Agregada	30	47

Tabela 3.2 - Número de variáveis e restrições da rede exemplo usando abordagem parcial.

O facto de existir uma topologia lógica estabelecida antes da formulação do problema de optimização de tráfego, permite reduzir bastante o número de variáveis necessárias. Para a topologia acima e considerando o par de nós (1,5), por exemplo, não seria necessário incluir as variáveis  $\lambda_{21}^{15}$ ,  $\lambda_{32}^{15}$ ,  $\lambda_{63}^{15}$ ,  $\lambda_{46}^{15}$  e  $\lambda_{54}^{15}$  uma vez que os segmentos lógicos (2,1), (3,2), (6,3), (4,6) e (5,4) nunca seriam usados por tráfego com origem no nó 1 e destinado ao nó 5. Usando assim somente as variáveis estritamente necessárias, é possível reduzir bastante a dimensão do problema. Isto já não é possível para a abordagem global uma vez que, como não está estabelecida uma topologia lógica, todas as hipóteses têm que ficar em aberto, ou seja, têm que ser consideradas todas as combinações possíveis.

A abordagem global é, sem dúvida, muito mais precisa e complexa. É que não só possui muito mais variáveis como também possui variáveis inteiras, dificultando ainda mais a sua resolução. Desta forma, achar uma solução óptima pode levar dias de computação. Há também o risco de, uma vez encontrada a solução óptima para uma determinada topologia física e situação de tráfego, esta resultar desajustada da realidade por perda de actualidade. Isto é particularmente verdade para tráfego que varia bastante ao longo do tempo.

O problema de encaminhamento para além de ser muito mais rápido de resolver permite análise pos-óptimo, isto é, uma vez encontrada a solução óptima para o problema, torna possível saber qual seria a solução óptima caso fossem feitas algumas alterações ao problema inicial, sem ser necessário resolver o problema de optimização desde o início. Esta análise pode não exigir quaisquer iterações ou, quando muito, serão necessárias algumas iterações. Desta forma, a análise pos-óptimo permite redistribuir o tráfego frequentemente, resultando sempre numa situação óptima para o encaminhamento do tráfego.

Ao usar-se algoritmos heurísticos, é possível criar rapidamente uma topologia lógica quase óptima. Encontrar uma solução óptima para a topologia lógica é, como já foi dito,

bastante complicado e, como esta pode estar desajustada momentos depois, é preferível ter a possibilidade de alterar a topologia lógica mais frequentemente embora com uma situação perto do óptimo.

Resumindo, é aconselhável optar-se pela alteração da topologia lógica a intervalos largos mantendo o encaminhado do tráfego otimizado durante a permanência de cada topologia.

### 3.5 SUMÁRIO

O problema de desenho da topologia lógica e encaminhamento do tráfego foi formulado matematicamente como um problema MILP cujo objectivo é minimizar o congestionamento na rede, abordagem indicada caso se pretenda maximizar o *throughput*. A minimização do congestionamento é conseguida através da minimização do fluxo total no segmento mais sobrecarregado e obrigando a que o fluxo nos outros segmentos não ultrapasse esse valor. O problema é apresentado quer na forma desagregada quer na forma agregada. Esta última é de resolução mais rápida mas, em contrapartida, os valores de congestionamento obtidos são maiores.

A abordagem seguida neste trabalho, e apresentada neste capítulo, em vez de tratar o problema na sua globalidade, divide-o em dois subproblemas: desenho da topologia lógica, resolvido com algoritmos heurísticos, e encaminhamento do tráfego, resolvido por um problema de programação linear. A vantagem desta abordagem é ser de resolução mais rápida e permitir análise pos-óptimo para uma redistribuição frequente do tráfego.

## 4. ANÁLISE E IMPLEMENTAÇÃO DOS ALGORITMOS GERADORES DE TOPOLOGIAS LÓGICAS E OPTIMIZAÇÃO DE TRÁFEGO PARA REDES *MULTI-HOP*

### 4.1 INTRODUÇÃO

O diagrama apresentado na figura 4.1 esquematiza a concepção do sistema de optimização.

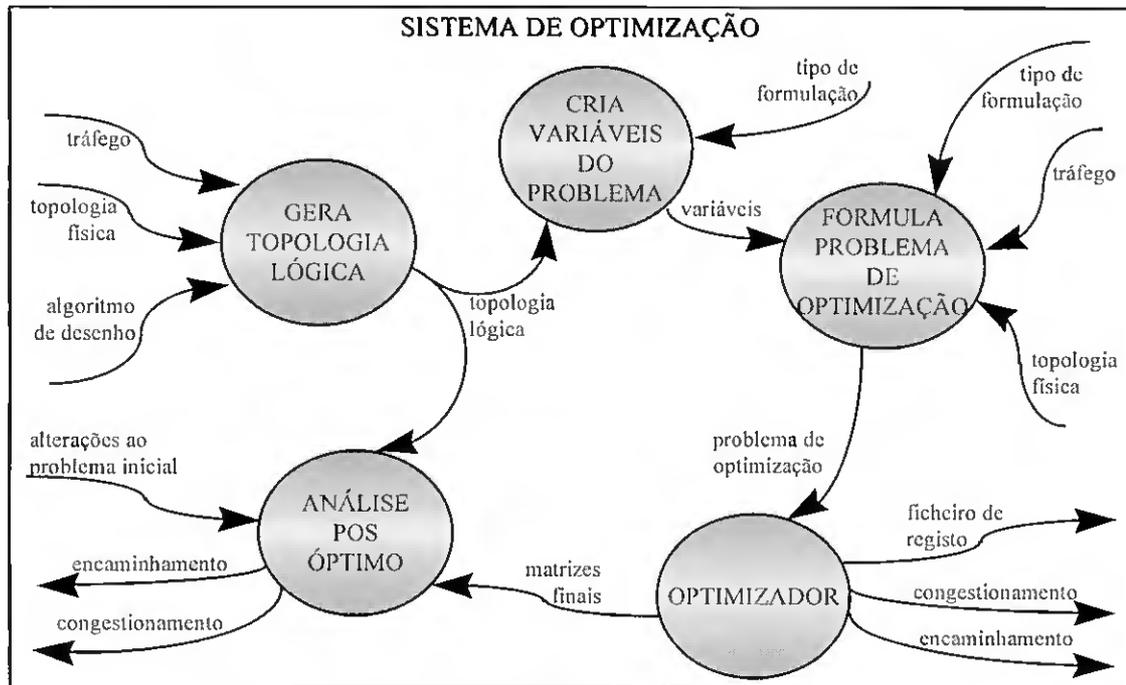


Fig. 4.1 - Diagrama do sistema de optimização.

O sistema final deverá satisfazer os seguintes requisitos:

- Perante uma topologia física e uma situação de tráfego, gerar a topologia lógica para um determinado algoritmo.
- Permitir a formulação do problema de optimização de forma desagregada ou agregada.
- Achar a solução óptima do problema de optimização tendo em conta que o objectivo é minimizar o congestionamento.

- Permitir análise pos-óptimo. Isto é, uma vez encontrada a solução óptima, permitir saber quais as consequências de uma alteração do problema inicial.

O otimizador foi desenvolvido em Watcom<sup>8</sup> C++, plataforma 32 bits. A opção pela linguagem C++ ficou a dever-se a três ordens de razões: facilidades oferecidas, perspectivas para desenvolvimento futuro e predilecção pessoal. De realçar que todos os módulos foram desenvolvidos de raiz para a realização deste trabalho.

O C++ é uma linguagem de elevada portabilidade que gera programas eficientes e rápidos. A sua sintaxe permite a elaboração de programas bem estruturados que aliam a modularidade das estruturas de controlo modernas à expressão de instruções de baixo nível, aumentando a velocidade e eficiência do programa. O uso de 32 bits permitiu não só um aumento da rapidez de execução como também a possibilidade de acesso a blocos de memória superiores a 64K<sup>9</sup>, essencial para problemas de grande dimensão. O trabalho com objectos permitiu também tratar os problemas de optimização de forma dinâmica, ajustando a quantidade de memória consumida às necessidades de cada caso.

Sendo uma linguagem de programação sofisticada e largamente utilizada no desenvolvimento profissional de sistemas, têm sido desenvolvidos compiladores de C++ para as mais diversas plataformas. Além disso, tem crescido o número de aplicações a adoptar a capacidade de integração com esta linguagem. Tais factos oferecem boas hipóteses de enriquecimento futuro deste *software*.

Por tudo o que foi exposto e por uma não menos importante predilecção pessoal, a linguagem C++ foi a adoptada para o desenvolvimento.

---

<sup>8</sup> Watcom é uma marca registada da Sybase, Inc.

<sup>9</sup> Usando apenas ponteiros *near*, isto é, manipulação apenas do *offset*.

## 4.2 ESTRUTURA DO OPTIMIZADOR

O otimizador está funcionalmente estruturado em 4 módulos principais:

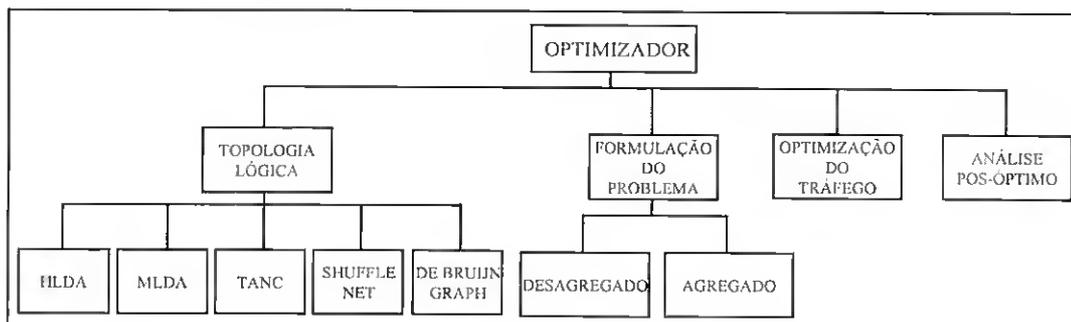


Fig. 4.2 - Estrutura funcional do otimizador.

De seguida será feita uma análise pormenorizada de cada um destes módulos.

### 4.2.1 Desenho da Topologia Lógica

Perante uma rede física e um cenário de tráfego, este módulo pretende criar uma topologia lógica que, segundo os critérios de cada algoritmo, seja considerada a mais adequada. Para este efeito foram implementados vários algoritmos heurísticos. Foram também implementadas duas topologias lógicas regulares específicas que têm sido propostas para redes *multi-hop*: *ShuffleNet* e *de Bruijn Graph* [26][27][28].

Foi então criada a classe “Cenário” que contém as variáveis e funções necessárias à implementação dos algoritmos. De realçar que o espaço de memória reservado para cada objecto varia dinamicamente conforme a dimensão do problema. As variáveis usadas por esta classe são:

*Nodes* - número de nós da rede.

*Wavelengths* - número de comprimentos de onda disponíveis. Este valor pode ser uma imposição do utilizador ou uma estimativa. Esta variável pode também não constituir uma restrição caso se pretenda avaliar o grau de consumismo dos vários algoritmos.

*DeltaL* - *degree* lógico máximo. Ou seja, o número máximo de segmentos lógicos de entrada e saída permitidos por um nó.

*Traffic* - matriz com a distribuição relativa do tráfego na rede.

*PhysTopology* - matriz com os segmentos não direccionados existentes na rede física (da diagonal para cima) e respectivos atrasos (da diagonal para baixo).

As matrizes são objectos que pertencem a uma classe chamada ‘Matr’. Esta classe possui todas as funções necessárias à manipulação de matrizes e memória por elas usada. É esta classe que permite tratar os problemas de forma dinâmica, ajustando durante a execução o espaço de memória necessário a cada caso.

#### SHUFFLENET

A *ShuffleNet* foi a primeira topologia lógica a ser proposta para redes *multi-hop* [26] existindo, por isso, muita bibliografia. Uma *ShuffleNet* ( $p, k$ ) é construída a partir de  $N$  nós onde:

$$N = k \cdot p^k \quad [4.1]$$

Os nós estão organizados em  $k$  colunas com  $p^k$  nós cada. Cada um dos  $p^k$  nós numa coluna tem  $p$  segmentos lógicos direccionados para  $p$  nós diferentes na coluna adjacente, o que significa que  $p$  é também o número de emissores e receptores fixos por estação. Deste modo, existem  $p^{k+1}$  segmentos entre colunas adjacentes. Como é associado um canal WDM a cada segmento lógico, no total a rede necessita de [27]:

$$L = k p^{k+1} = PN \quad [4.2]$$

comprimentos de onda. Se os nós de uma coluna forem numerados de 0 a  $p^k-1$ , a estação  $j$  tem segmentos para as estações  $i, i+1, \dots, i+p-1$  na coluna adjacente onde [27]:

$$i = (j \bmod p^{k-1})p \quad [4.3]$$

sendo “mod” o resto da divisão do primeiro operador pelo segundo. A última coluna está ligada à primeira de forma cilíndrica. A figura seguinte mostra uma *ShuffleNet* com  $p=2$  e  $k=2$ .

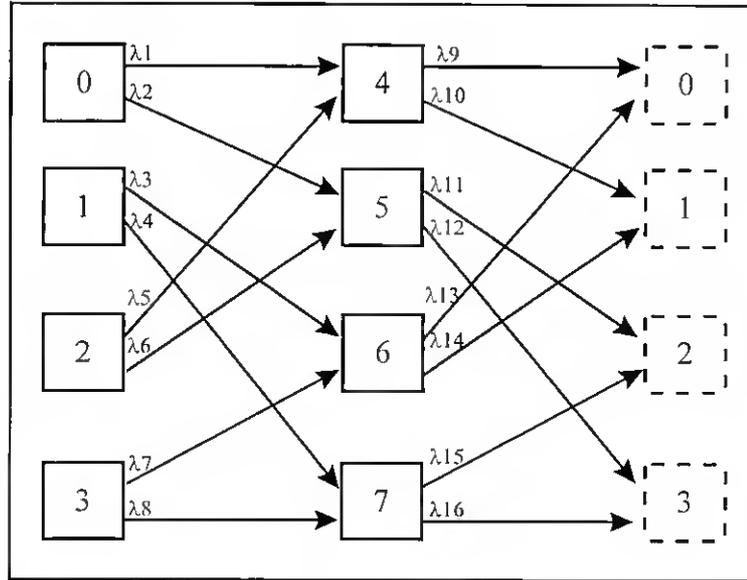


Fig. 4.3 - Topologia *ShuffleNet* ( $p=2, k=2$ ) de 8 utilizadores.

Seleccionando um nó origem e um nó destino aleatoriamente, em média o número de saltos (nós intermédios por onde passam os pacotes) é dado por [27]:

$$\bar{H} = \frac{kp^k(p-1)(3k-1) - 2k(p^k-1)}{2(p-1)(kp^k-1)} \quad [4.4]$$

```

Função: ShuffleNet
INICIO
    k=número de colunas;
    p=número de transmissores (e receptores) por estação;
    PARA CADA coluna=1,...,k FAZ
        PARA CADA nó=0,...,pk-1 FAZ
            i=(nó mod pk-1)*p;
            PARA CADA segLogico=0,...,p-1 FAZ
                s=nó+(coluna-1)*pk;
                SE coluna=k ENTÃO
                    d=i+segLogico;
                SENAO
                    d=i+segLogico+coluna*pk;
            FIMSE
            CriaSegmentoLogico(s,d);
        FIMFAZ;
    FIMFAZ;
FIM;
    
```

Listagem 4.1 - Algoritmo *ShuffleNet*.

### DE BRUIJN GRAPH

Esta é outra configuração lógica importante e bem conhecida. Uma topologia *de Bruijn Graph* ( $\delta, D$ ) é construída a partir de  $N$  nós onde [26][28]:

$$N = \delta^D, \quad \delta \geq 2 \text{ e } D \geq 2 \quad [4.5]$$

Se representarmos o índice de cada nó na base  $\delta$  usando  $D$  posições (isto é, cada posição pode assumir os números  $0, 1, 2, \dots, \delta-1$ ), só é criado um segmento lógico entre o nó  $A = a_1 a_2 \dots a_D$  e o nó  $B = b_1 b_2 \dots b_D$  se [28]:

$$b_i = a_{i+1} \text{ onde } a_i, b_i \in \{0, 1, \dots, \delta-1\} \text{ e } 1 \leq i \leq D-1 \quad [4.6]$$

Deste modo, cada nó irá possuir um *degree* igual a  $\delta$ . Note-se que nesta topologia existem segmentos onde a origem é igual ao destino (chamados *self-loops*). Estes segmentos podem ser eliminados. A figura seguinte mostra uma topologia de *Bruijn Graph* com  $\delta=2$  e  $D=3$ .

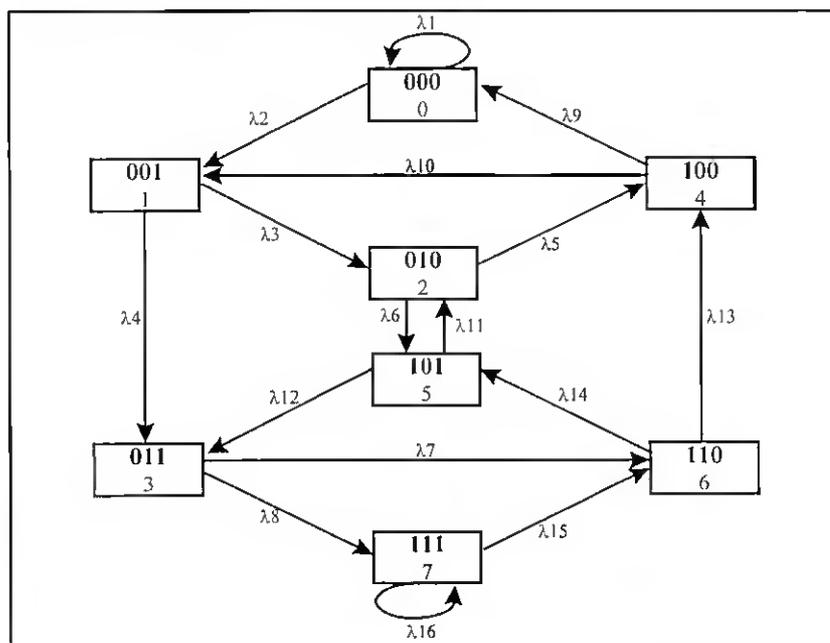


Fig. 4.4 - Topologia de *Bruijn Graph* ( $\delta=2, D=3$ ) de 8 utilizadores.

Relativamente ao número médio de saltos sofrido pelos pacotes, sabe-se que [26]:

$$\bar{H} \leq D \frac{N}{N-1} - \frac{1}{\delta-1} \quad [4.7]$$

```

Função: deBruijnGraph
INICIO
   $\delta$ =base de numeração;
  D=número de dígitos;
  NóBaseDelta={};
  PARA CADA nó=0,..., $\delta^D-1$  FAZ
    NóBaseDelta[nó]=nó representado no sistema de numeração de base  $\delta$  com D dígitos;
  FIMFAZ;
  PARA CADA nó=0,..., $\delta^D-1$  FAZ
    destino=0;
    PARA CADA dígito=1,...,D-1 FAZ
      destino=destino+NóBaseDelta[nó][dígito]* $\delta^{D-dígito}$ ;
    FIMFAZ;
    PARA CADA últimoDígito=0,..., $\delta-1$  FAZ
      SE (nó  $\neq$  destino+últimoDígito) ENTAO
        s=nó;
        d=destino+últimoDígito;
        CriaSegmentoLogico(s,d);
      FIMSE
    FIMFAZ;
  FIMFAZ;
FIM;

```

Listagem 4.2 - Algoritmo de Bruijn Graph.

#### HLDA (*Heuristic Logical topology Design Algorithm*)

Este algoritmo heurístico proposto em [22] atribui segmentos lógicos a pares de nós por ordem descendente de tráfego. Ao encaminhar o maior fluxo de tráfego num só salto o congestionamento é minimizado. Por este motivo, este algoritmo possui um bom desempenho quando o tráfego está concentrado numa pequena fracção do número total de pares fonte-destino na rede, mas não funciona bem quando o tráfego está distribuído uniformemente.

```

Função: HLDA
INICIO
  copia_trafego=matriz_de_trafego;
  nSegmentosLogicos=0;
  FAZ
    ( $s_{max}, d_{max}$ )=par com maior tráfego diferente de 0;
    SE (par encontrado) ENTAO
      SE (SegmentosDeEntrada( $s_{max}, d_{max}$ ) $<\Delta_i$  e SegmentosDeSaida( $s_{max}, d_{max}$ ) $<\Delta_i$ ) ENTAO
         $\lambda_{min}$ =primeiro comp. onda disponível no caminho com menor atraso de propagação;
        SE (existe  $\lambda_{min}$ ) ENTAO
          CriaSegmentoLogico( $s_{max}, d_{max}$ );
          Reservar o comprimento de onda nos segmentos físicos usados pelo segmento lógico;
          nSegmentosLogicos=nSegmentosLogicos+1;
          ( $s'_{max}, d'_{max}$ )=próximo par com maior tráfego, ie, que não  $s_{max}, d_{max}$ ;
          copia_trafego( $s_{max}, d_{max}$ )=copia_trafego( $s_{max}, d_{max}$ )-copia_trafego( $s'_{max}, d'_{max}$ );
        SENAO
          copia_trafego( $s_{max}, d_{max}$ )=0;
        FIMSE;
      SENAO
        copia_trafego( $s_{max}, d_{max}$ )=0;
      FIMSE;
    FIMSE;
  ENQUANTO (par encontrado);
  SE (nSegmentosLogicos $<n$ Nós* $\Delta_i$ ) ENTAO
    Sem exceder os comprimentos de onda disponíveis e  $\Delta_i$ , criar tantos segmentos lógicos quanto possível de forma aleatória;
  FIMSE;
FIM;

```

Listagem 4.3 - Algoritmo HLDA.

Note-se que este algoritmo pode atribuir múltiplos segmentos lógicos a um par de nós com elevado tráfego. Para um *degree* lógico pequeno este algoritmo poderá originar situações de *loop* que impedem o envio de tráfego de todos os nós para todos os destinos (para um *degree* lógico de 1, por exemplo, poderá acontecer o algoritmo optar por atribuir os segmentos lógicos (1,2) e (2,1), isolando estes nós da restante rede).

Para o cálculo de  $\lambda_{\min}$ , primeiro comprimento de onda disponível no caminho com menor atraso, foi usado o algoritmo de *Dijkstra* com algumas modificações. A filosofia base deste algoritmo é a seguinte:

*Cada nó tem uma indicação entre parêntesis (etiqueta) com a sua distância ao nó origem usando o melhor caminho conhecido. Inicialmente todos os nós têm o valor das suas etiquetas a infinito e, à medida que os caminhos são encontrados, o seu valor muda, reflectindo os melhores caminhos. Uma etiqueta pode ser considerada como temporária ou permanente. Inicialmente todas são temporárias. Quando se descobre que uma etiqueta representa o menor caminho da fonte até esse nó, ela fica permanente e não muda mais.*

As modificações foram feitas ao nível das etiquetas para que estas, durante a busca do melhor caminho (caminho com menor atraso), considerem também os comprimentos de onda disponíveis nos segmentos. Como ilustração, considere-se a seguinte rede onde estão representados os atrasos de propagação e os comprimentos de onda livres de cada segmento.

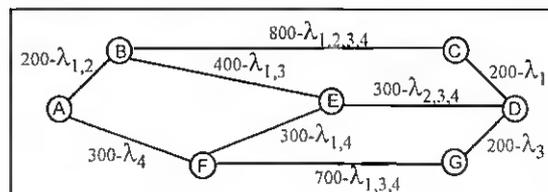


Fig. 4.5 - Rede exemplo.

Pretende-se, neste caso, encontrar o menor comprimento de onda disponível no caminho com menor atraso entre os nós *A* e *D*.

**Passo 1** - O nó origem é marcado como permanente e as etiquetas são inicializadas. Cada etiqueta contém a distância ao nó origem, último nó que actualizou a etiqueta e estado de cada comprimento de onda. Para o último elemento foram usados tantos bits quantos os comprimentos de onda existentes (bit=0, comprimento de onda livre; bit=1, caso contrário) onde o primeiro bit a contar da direita pertence a  $\lambda_1$  e o último a  $\lambda_4$ .

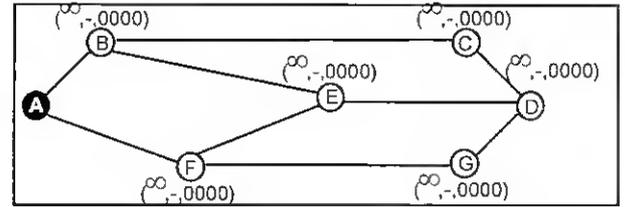


Fig. 4.5 a) - Rede exemplo - passo 1.

**Passo 2** - Os nós adjacentes ao nó A são analisados (B e F) e as suas etiquetas actualizadas. O estado dos comprimentos de onda é o resultado da operação lógica OR entre o estado dos comprimentos de onda na etiqueta anterior e no segmento (por exemplo, para B:  $0000 \oplus 1100 = 1100$ ). De seguida, todos os nós com etiqueta temporária são comparados e o menor é tornado permanente.

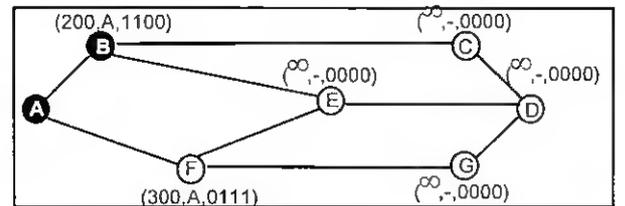


Fig. 4.5 b) - Rede exemplo - passo 2.

**Passo 3** - Os nós (não permanentes) adjacentes a B (C e E) são analisados. As suas etiquetas só são actualizadas se a distância em B somada com a distância que vai de B ao nó que está a ser considerado for menor que a etiqueta desse nó. De todos os nós com etiqueta temporária o menor é tornado permanente.

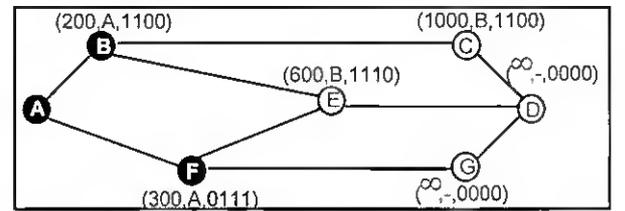


Fig. 4.5 c) - Rede exemplo - passo 3.

**Passo 4** - Os nós (não permanentes) adjacentes a F (E e G) são analisados (E e G) e as suas etiquetas actualizadas usando os mesmos critérios dos passos anteriores. Para os comprimentos de onda, como a distância em F somada com a distância F-E é igual à distância na etiqueta do nó E, para além da operação OR já referida ( $0111 \oplus 0110$ , neste caso) deve ser feito o AND entre o resultado da operação anterior e o valor na etiqueta do nó E. Em suma:  $(0111 \oplus 0110) \otimes 1110 = 0110$ . Isto porque na existência de dois caminhos mais curtos, ambos devem ser considerados.

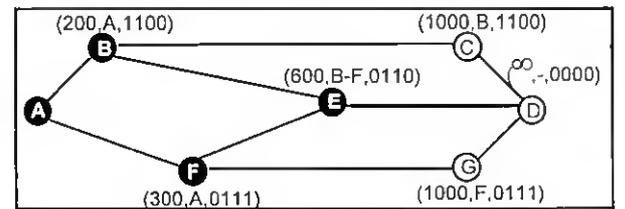


Fig. 4.5 d) - Rede exemplo - passo 4.

**Passo 5** - O nó adjacente a E é actualizado (D) e o nó com menor etiqueta é tornado permanente (D). Como se trata do nó destino, o processo terminou.

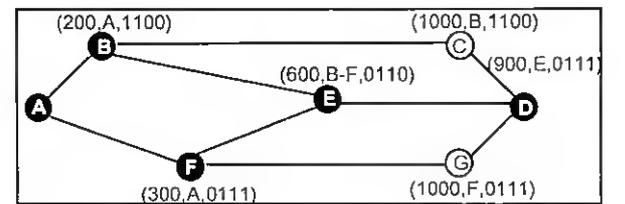


Fig. 4.5 e) - Rede exemplo - passo 5.

O menor comprimento de onda livre encontrado foi  $\lambda_4$  (0111). O percurso é conseguido com uma leitura das etiquetas em sentido inverso (*D-E-F-A*).

#### MLDA (*Minimum-delay Logical topology Design Algorithm*)

Este algoritmo heurístico também proposto em [22] está somente definido para um *degree* lógico superior ou igual ao *degree* físico<sup>10</sup>. Se este for o caso, o MLDA cria um par de segmentos lógicos (com sentidos contrários) para cada segmento físico, e os restantes segmentos são criados de acordo com o algoritmo HLDA. Deste modo, as topologias criadas pelo MLDA permitem encaminhar os pacotes pelo caminho físico mais curto e, por isso, são capazes de satisfazer as restrições de atraso por mais apertadas que sejam[22]. O problema deste algoritmo é não estar definido para um *degree* lógico inferior ao físico, o que pode acontecer.

```

Função: MLDA
INICIO
  nSegmentosLogicos=0;
  PARA CADA segFisico=1 ... totalSegFisicos FAZ
    s=nó num dos extremos do segmento físico;
    d=nó no outro extremo do segmento físico;
    CriaSegmentoLogico(s,d);
    Reservar  $\lambda_1$  no segmento físico;
    Trocar(s,d);
    CriaSegmentoLogico(s,d);
    Reservar  $\lambda_2$  no segmento físico;
  nSegmentosLogicos=nSegmentosLogicos+2;
  FIM FAZ;
  Cria restantes segmentos de acordo com HLDA;
FIM;

```

Listagem 4.4 - Algoritmo MLDA.

#### TANC (*Topology with All Nodes Connected*)

Os algoritmos heurísticos apresentados anteriormente possuem algumas desvantagens. O HLDA, para um *degree* pequeno, pode criar situações de *loop* que impedem a comunicação entre todos os nós na rede. O algoritmo MLDA tem a desvantagem de só estar definido para um *degree* lógico superior ou igual ao *degree* físico, requisito que nem sempre é possível satisfazer. O algoritmo heurístico TANC proposto neste trabalho surgiu para ultrapassar estes dois problemas. O TANC começa por encontrar o conjunto de segmentos físicos com menor atraso

<sup>10</sup> Maior número de segmentos não direccionados ligados a um nó.

global que liga todos os nós da rede. Os segmentos lógicos (direccionados) são criados sobre os segmentos físicos encontrados começando pelo par de nós adjacentes (considerando apenas o conjunto de segmentos físicos encontrado) com maior tráfego. Desta forma todos os nós estão ligados usando o caminho mais curto e o menor número de comprimento de onda possível. Se o *degree* lógico for superior a 1, para cada segmento lógico existente é criado um outro em sentido inverso. Para um *degree* lógico superior a 2 todos restantes segmentos lógicos são criados usando o algoritmo HLDA.

```

Função: TANC
INICIO
  nSegmentosLogicos=0;
  copia_trafego=matriz_de_trafego;
  Encontrar conexao com menor atraso;
  (s,d)=par de nós adjacentes com maior tráfego. Devem ser considerados apenas os segmentos
    físicos da conexao encontrada;
  CriaSegmentoLogico(s,d);
  Reservar o comprimento de onda nos segmentos físicos usados pelo segmento lógico (s,d);
  nSegmentosLogicos=nSegmentosLogicos+1;
  copia_trafego(s,d)=0;
  ENQUANTO (d ≠ 0) FAZ
    s=d;
    d=proximo segmento ligado a s (que nao o nó onde foi gerado o primeiro segmento);
    IF (d ≠ 0) ENTÃO
      CriaSegmentoLogico(s,d);
      Reservar o comprimento de onda nos segmentos físicos usados pelo segmento lógico (s,d);
      nSegmentosLogicos=nSegmentosLogicos+1;
      copia_trafego(s,d)=0;
    FIMSE;
  FIMFAZ;
  SE ( $\Delta > 1$ ) ENTÃO
    Para cada segmento lógico existente criar um outro em sentido inverso;
  FIMSE;
  SE ( $\Delta > 2$ ) ENTÃO
    Cria restantes segmentos de acordo com HLDA;
  FIMSE;
FIM;

```

Listagem 4.5 - Algoritmo TANC.

#### 4.2.2 Algoritmo para Formulação do Problema de Encaminhamento de Tráfego

Uma vez criada a topologia lógica mais adequada à topologia física e cenário de tráfego, torna-se necessário formular o problema de encaminhamento de tráfego cuja solução será encontrada pelo algoritmo de resolução de problemas LP (*Linear Programming*) analisado posteriormente. Nesta fase é possível fazer uma formulação desagregada ou agregada do problema. Independentemente do modo de formulação escolhido esta fase incluirá duas etapas:

1. Criação das variáveis envolvidas no problema de optimização.
2. Formulação do problema LP envolvendo as variáveis criadas. Esta formulação inclui a função objectivo e restrições a que estão sujeitas as variáveis.

### CRIAÇÃO DAS VARIÁVEIS

Esta fase encarrega-se de, perante a topologia lógica criada, gerar as variáveis envolvidas no problema de encaminhamento. Como já foi mencionado anteriormente na secção 3.4.3, só devem ser usadas as variáveis estritamente necessárias, isto é, para todos os pares fonte-destino deverá ser feita uma análise de quais os segmentos lógicos usados e só estes irão entrar no problema de optimização. O pseudo-código seguinte mostra em traços gerais como foram geradas as variáveis quer para a formulação desagregada quer para a agregada.

```

Função: CriaVariaveisDesag
INICIO
  segOcupados={};
  PARA CADA fonte=1 .. nNós FAZ
    PARA CADA destino=1 ... nNós FAZ
      SE (fonte ≠ destino) ENTAO
        segOcupados=MarcarSegmentosOcupados(fonte,destino);
        PARA CADA linha=1 .. nNós FAZ
          PARA CADA coluna=1 ... nNós FAZ
            nSegmentos=segOcupados(linha,coluna);
            PARA CADA segmento=1...nSegmentos FAZ
              s=fonte;
              d=destino;
              i=linha;
              j=coluna;
              k=segmento;
              CriaVariavel(s,d,i,j,k);
            FIMFAZ;
          FIMFAZ;
        FIMFAZ;
      FIMSE;
    FIMFAZ;
  FIMFAZ;
FIM;

```

Listagem 4.6 - Criação das variáveis do problema de encaminhamento usando a forma desagregada.

```

Função: CriaVariaveisAgreg
INICIO
  segOcupados={};
  PARA CADA fonte=1...nNós FAZ
    PARA CADA destino=1...nNós FAZ
      SE (fonte ≠ destino) ENTÃO
        segOcupados=segOcupados+MarcarSegmentosOcupados(fonte,destino);
      FIMSE;
    FIMFAZ;
  PARA CADA linha=1...nNós FAZ
    PARA CADA coluna=1...nNós FAZ
      nSegmentos=segOcupados(linha,coluna);
      PARA CADA segmento=1...nSegmentos FAZ
        s=fonte;
        i=linha;
        j=coluna;
        k=segmento;
        CriaVariavel(s,i,j,k);
      FIMFAZ;
    FIMFAZ;
  FIMFAZ;
  LimparMatriz(segOcupados);
FIMFAZ;
FIM;

```

Listagem 4.7 - Criação das variáveis do problema de encaminhamento usando a forma agregada.

Note-se que um segmento lógico é designado por  $(i,j,k)$  porque é permitida a atribuição de mais que um segmento lógico a um par de nós.

#### FORMULAÇÃO DO PROBLEMA LP

Uma vez definidas as variáveis envolvidas, nesta fase é gerada a função objectivo e as restrições do problema LP com as quais o programa de optimização irá trabalhar.

#### 4.2.3 Algoritmo de Optimização

O algoritmo de optimização usado para o encaminhamento do tráfego foi o Simplex Revisto. Apesar de existirem *packages* que implementam este algoritmo, optou-se pela sua implementação de raiz. Os motivos desta escolha são referidos após descrição do algoritmo uma vez que se torna necessário conhecer o seu funcionamento.

#### A ESCOLHA DO SIMPLEX

O algoritmo Simplex foi desenvolvido por George Dantzig em 1947 e provou ser bastante eficiente na resolução de problemas lineares [25]. Em 1984, Narendra Karmarkar apresentou um algoritmo (*interior-point algorithm*) que, na resolução de problemas lineares

muito grandes, apresentava tempos de computação bastante mais baixos que o Simplex. Como nenhum código foi disponibilizado, este algoritmo só pode ser usado como uma caixa negra. No que diz respeito à análise pos-óptimo, este algoritmo não consegue fazer uma análise eficiente, sendo o Simplex o mais indicado [25]. Em suma, apesar de menos rápido, o Simplex possui duas vantagens bastante importantes para este trabalho: o facto de o seu código ser conhecido e de possibilitar uma análise pos-óptimo eficiente.

### O MODELO MATEMÁTICO

Um modelo matemático envolve um conjunto de  $n$  decisões relacionadas de alguma forma e que são quantificáveis. Estas decisões são representadas pelas **variáveis de decisão**  $(x_1, x_2, \dots, x_n)$  cujos valores se pretende encontrar. A medida de performance é representada por uma função matemática destas variáveis de decisão  $(P = a_1x_1 + a_2x_2 + \dots + a_nx_n$ , onde  $a_1, a_2, \dots, a_n$  são constantes). A esta função é dado o nome de **função objectivo**. Caso existam limitações quanto aos valores que as variáveis de decisão podem tomar, elas devem ser representadas matematicamente através de igualdades ou desigualdades  $(a_1x_1 + a_2x_2 + \dots + a_nx_n \leq b$ , onde  $a_1, a_2, \dots, a_n$  e  $b$  são constantes). A estas limitações é dado o nome de **restrições**. No caso particular de um modelo que representa um problema de programação linear, a função objectivo e as restrições são funções lineares. De forma geral, estes modelos tomam o seguinte aspecto [25]:

Maximizar: $Z = c_1x_1 + c_2x_2 + \dots + c_nx_n$ Sujeito a: $a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1$ $a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2$ $\vdots$ $a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m$ $x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0$	[4.8]
--	-------

Qualquer situação que se enquadre neste modelo é um problema de programação linear. As seguintes formas são também legítimas:

1. Minimizar a função objectivo:

$$\text{Min } Z = c_1x_1 + c_2x_2 + \dots + c_nx_n \quad [4.9]$$

2. Restrições sob a forma de:

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \geq b_i \quad [4.10]$$

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n = b_i \quad [4.11]$$

3. Retirar as restrições de não negatividade das variáveis de decisão:

$$x_j \text{ sem restrição de sinal} \quad [4.12]$$

Qualquer problema que possua uma mistura destas formas continua a ser um problema de programação linear, uma vez que todas elas podem ser escritas de modo a respeitar o modelo geral.

Relativamente às soluções do modelo, uma **solução admissível** constitui uma solução (qualquer especificação de valores para  $x_1, x_2, \dots, x_n$ ) em que todas as restrições são satisfeitas. A **solução óptima** é a solução admissível com valor mais favorável na função objectivo ( $Z$ ).

Para que o nosso problema de encaminhamento do tráfego respeite o modelo geral, é necessário escrever a primeira restrição de forma a não existirem variáveis no lado direito das restrições. O modelo fica então com o seguinte aspecto.

<b>Objectivo:</b>	$\min \lambda_{max}$	
<b>Sujeito às restrições de:</b>		
Congestionamento:	$\lambda_{ij} - \lambda_{max} \leq 0$	$\forall i, j$
Reencaminhamento:	$\sum_j \lambda_{ij}^{sd} - \sum_j \lambda_{ji}^{sd} = \begin{cases} \lambda^{sd} & , \text{ se } s=i \\ -\lambda^{sd} & , \text{ se } d=i \\ 0 & \forall s, d, i \end{cases}$	$, \text{ c. contrário}$
Limite de atraso:	$\sum_{i,j} \lambda_{ij}^{sd} d_{ij} \leq \lambda^{sd} \alpha_{min}^{sd}$	$\forall s, d$
	$\lambda_{ij}^{sd}, \lambda_j, \lambda_{max} \geq 0$	$\forall i, j, s, d$

[4.13]

O método Simplex e o método Simplex Revisto encontram-se descritos e exemplificados no apêndice A.

## A IMPLEMENTAÇÃO

Apesar de existirem *packages* de optimização com o algoritmo Simplex, houve um conjunto de razões que motivaram a sua implementação de raiz:

1. A *package* MatLab<sup>11</sup> disponível obrigava à introdução do problema na forma *standard* (restrições do tipo  $a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1$ ). Como no nosso problema de encaminhamento de tráfego muitas restrições são igualdades, isto obrigava à introdução de variáveis extra para as transformar em desigualdades e depois, internamente, o MatLab voltar a colocar o problema só com igualdades.
2. O MatLab (na versão disponível – 4.2) não permitia a introdução de problemas de grande dimensão como o problema de optimização de tráfego em estudo. Esta limitação foi um factor determinante na sua não utilização.
3. Havia necessidade de manipular directamente as variáveis de trabalho usadas pelo algoritmo, nomeadamente na análise pos-óptimo.

Com todo o código desenvolvido em C++, plataforma 32 bits, foi possível a minimização do número de variáveis usadas, a introdução de problemas de grande dimensão, a manipulação directa de variáveis internas ao algoritmo e a obtenção de tempos de execução bastante mais rápidos que os obtidos pelo MatLab.

Deste modo, foi criada a classe<sup>12</sup> “Simplex” que contém todas as variáveis e funções necessárias à implementação do algoritmo. De realçar que o espaço de memória reservado para cada objecto varia dinamicamente conforme a dimensão do problema. As variáveis usadas por esta classe são:

<sup>11</sup> MatLab é uma marca registada da *MathWorks, Inc.*.

<sup>12</sup> O trabalho com objectos foi mais uma razão a determinar a escolha do C++ como linguagem.

*nDecisionVariables* - Número de variáveis de decisão.

*nArtificialVariables* - Número de variáveis artificiais.

*nTotalVariables* - Número total de variáveis (decisão+afastamento+artificiais).

*nRestrictions* - Número de restrições.

*Objective* - Objectivo do problema: minimizar ou maximizar.

*ObjectiveFunctionCoefficients* - Vector com os coeficientes da função objectivo.

*RestrictionCoefficients* - Matriz com coeficientes das restrições.

*RestrictionSignals* - Vector com sinais das restrições: menor, maior ou igual.

*IndependentTerms* - Vector com os termos independentes das restrições.

*Work* - Estrutura com todas as variáveis de trabalho usadas pelo algoritmo. Estas variáveis mudam em todas as iterações até ser encontrada a solução óptima.

*XB* - Vector das variáveis básicas.

*XN* - Vector das variáveis não básicas.

*CN* - Vector com os custos das variáveis não básicas.

*N* - Vector com os coeficientes da variável de entrada em todas as restrições.

*TI* - Vector dos termos independentes

*VAVector* - Vector das variáveis artificiais

*Binverted* - Inversa da matriz com os coeficientes das variáveis básicas.

*Z* - Actual valor da função objectivo.

Todas as variáveis tipo vector ou matriz são objectos que pertencem à classe já apresentada 'Matr'.

A listagem seguinte mostra, em traços gerais, a implementação do algoritmo Simplex Revisto.

```

Função: SimplexRevisto
INICIO
    tipoSolucao=0;
    Inicializar objecto e variáveis;
    Colocar problema na forma aumentada;
    SE (nVariaveisArtificiais ≠ 0) ENTÃO
        fase=1;
        Criar função objectivo para a fase 1 do método das duas fases;
        TrocarFunçõesObjectivos();
        Colocar problema na forma correcta;
    SENÃO
        fase=2;
    FIMSE;
    ENQUANTO (tipoSolução=0) FAZ
        varEntrada=EntradaNaBase(fase);
        SE (varEntrada ≠ 0) ENTÃO
            varSaida=SaidaDaBase();
            SE (varSaida ≠ 0) ENTÃO
                TrocaVariaveis(varSaida,varEntrada);
                ConstroiBInvertido(varSaida);
                CalculaCN();
                CalculaXB(varSaida);
                CalculaZ();
            SENÃO
                tipoSolução=ILIMITADA;
        FIMSE;
    SENÃO
        SE (variáveis não básicas ≠ 0) ENTÃO
            tipoSolução=OPTIMO;
        SENÃO
            tipoSolução=OPTIMO_MULTIPLO;
        FIMSE;
    FIMSE;
    SE (fase=1 e (tipoSolução=OPTIMO || tipoSolução=OPTIMO_MULTIPLO)) ENTÃO
        fase=2;
        SE (variaveis artificiais = 0) ENTÃO
            TrocarFunçõesObjectivos();
            EliminaVarArtificiaisNãoBasicas();
            Colocar problema na forma correcta;
            CalculaZ();
            tipoSolução=0;
        SENÃO
            tipoSolução=IMPOSSIVEL;
        FIMSE;
    FIMSE;
    FIMFAZ;
FIM;

```

**Listagem 4.8** - Algoritmo Simplex Revisto.

De realçar que os inevitáveis problemas de arredondamento produzidos pelos computadores são controlados pela função *ConstroiBInvertido* uma vez que  $\mathbf{B}^{-1}$  é periodicamente calculada através da inversão directa de  $\mathbf{B}$ .

Como será usada análise pos-óptimo não são eliminadas restrições redundantes nem variáveis artificiais no final da fase 1 do método das duas fases (explicado no apêndice A).

#### 4.2.4 Análise Pos-Ótimo

Uma vez obtida a solução ótima para um modelo LP, é frequentemente necessário achar uma nova solução ótima para uma versão ligeiramente diferente do modelo original. Uma alternativa seria reaplicar o algoritmo usado para a resolução do problema linear, neste caso o método Simplex Revisto<sup>13</sup>, desde o início para cada nova versão do modelo original. Esta abordagem exigiria bastante tempo de computação. Uma alternativa bem mais atractiva será usar-se análise pos-ótimo, isto é, análise feita após obtenção de uma solução ótima para a versão inicial do modelo. Dos vários fins a que a análise pos-ótimo se destina, é de particular interesse para este trabalho analisar o efeito na solução ótima de alterações a determinados parâmetros. A grande vantagem desta abordagem é que, usando informação final do modelo original, nenhuma ou apenas algumas iterações serão necessárias para se refazer a optimização do problema agora modificado, e para se obter uma nova solução ótima.

#### ANÁLISE POS-ÓTIMO APLICADA AO PROBLEMA DE ENCAMINHAMENTO

Como já foi mencionado anteriormente, uma das características das redes *multi-hop* é a configuração da topologia lógica ser afectada apenas a intervalos relativamente grandes. Como a abordagem escolhida neste trabalho trata o problema da criação da topologia lógica separadamente do problema de encaminhamento do tráfego, é possível, tirando partido das vantagens da análise pos-ótimo, reajustar frequentemente a distribuição do tráfego na topologia lógica conforme a matriz de tráfego do momento. Assim, é proposto que uma rede funcione em duas fases distintas:

1. Determinação da topologia lógica mais adequada à rede física e à distribuição relativa de tráfego. Como os valores de tráfego mudam durante o tempo de vida de uma topologia lógica, pode usar-se para esta fase uma matriz de tráfego que represente,

---

<sup>13</sup> O método Simplex é, como já foi dito, o ideal para fazer análise pos-ótimo. Este foi um factor determinante na sua escolha.

por exemplo, a média dos valores de tráfego previstos para esse período. A redefinição da topologia lógica deve ser feita a intervalos relativamente grandes uma vez que a sua alteração implica também uma resolução de raiz do problema de encaminhamento.

2. Encaminhamento do tráfego na topologia lógica. Após determinada a topologia lógica, e usando a mesma matriz de tráfego médio, é resolvido o problema de otimização de tráfego e encontrada uma solução ótima. Como a rede será confrontada não com os valores médios de tráfego mas sim com valores que variam ao longo desse período, deverá ser aplicada frequentemente análise pos-ótimo de forma a determinar a redistribuição de tráfego mais adequada a cada momento. Dito de outra forma, o encaminhamento deve ser ajustado de forma ótima às variações de tráfego registadas enquanto se mantiver a topologia lógica.

Se a análise pos-ótimo não fosse usada, a distribuição de tráfego inicialmente escolhida manter-se-ia durante todo período de permanência da topologia lógica. Para redes com tráfego que varia bastante ao longo do tempo (tráfego tipo *burst*) esta situação poderia provocar níveis de congestionamento bastante elevados uma vez que o cenário de tráfego do momento pode ser bastante diferente da matriz de tráfego médio para a qual o problema tinha sido otimizado. Com análise pos-ótimo é possível haver sempre uma situação ótima no que diz respeito ao encaminhamento do tráfego.

**ALTERAÇÃO DO MODELO INICIAL**

Consideremos o problema de encaminhamento de tráfego.

<b>Objectivo:</b>	$\min \lambda_{max}$	
<b>Sujeito às restrições de:</b>		
Congestionamento:	$\lambda_{ij} - \lambda_{max} \leq 0$	$\forall i, j$
Reencaminhamento:	$\sum_j \lambda_{ij}^{sd} - \sum_j \lambda_{ji}^{sd} = \begin{cases} \lambda^{sd} & , \text{ se } s=i \\ -\lambda^{sd} & , \text{ se } d=i \quad \forall s, d, i \\ 0 & , \text{ c. contrário} \end{cases}$	
Limite de atraso:	$\sum_{i,j} \lambda_{ij}^{sd} d_{ij} \leq \lambda^{sd} \alpha d_{min}^{sd} \quad \forall s, d$	
	$\lambda_{ij}^{sd}, \lambda_{ij}, \lambda_{max} \geq 0 \quad \forall i, j, s, d$	

[4.14]

Suponhamos que o método Simplex tinha sido usado para obtenção de um encaminhamento óptimo com base na matriz de tráfego médio. Consideremos ainda que se pretendia aferir os efeitos de uma alteração nos valores de tráfego. Neste caso apenas os termos independentes das restrições do modelo inicial iriam sofrer alterações, ou seja, o novo problema de encaminhamento ficaria:

<b>Objectivo:</b>	$\min \lambda_{max}$	
<b>Sujeito às restrições de:</b>		
Congestionamento:	$\lambda_{ij} - \lambda_{max} \leq 0$	$\forall i, j$
Reencaminhamento:	$\sum_j \lambda_{ij}^{sd} - \sum_j \lambda_{ji}^{sd} = \begin{cases} \bar{\lambda}^{sd} & , \text{ se } s=i \\ -\bar{\lambda}^{sd} & , \text{ se } d=i \quad \forall s, d, i \\ 0 & , \text{ c. contrário} \end{cases}$	
Limite de atraso:	$\sum_{i,j} \lambda_{ij}^{sd} d_{ij} \leq \bar{\lambda}^{sd} \alpha d_{min}^{sd} \quad \forall s, d$	
	$\lambda_{ij}^{sd}, \lambda_{ij}, \lambda_{max} \geq 0 \quad \forall i, j, s, d$	

[4.15]

Usando a notação matricial do Simplex Revisto (ver apêndice A) passaríamos a ter um novo vector de termos independentes:

$$\bar{\mathbf{b}} = \begin{bmatrix} \bar{b}_1 \\ \bar{b}_2 \\ \vdots \\ \bar{b}_m \end{bmatrix} \quad [4.16]$$

Uma alteração deste género apenas irá ter reflexos nos valores finais das variáveis básicas e, conseqüentemente, no valor final de  $Z$ . É importante lembrar que a solução do problema é lida directamente nas variáveis básicas uma vez que o Simplex atribui o valor zero às variáveis não básicas.

Os valores finais das variáveis básicas do novo problema de encaminhamento seriam obtidos fazendo

$$\bar{\mathbf{x}}_B = \mathbf{B}^{-1} \bar{\mathbf{b}} \quad [4.17]$$

onde  $\mathbf{B}^{-1}$  é a inversa da matriz básica obtida na última iteração do problema de encaminhamento original [25]. O valor final de  $Z$  (congestionamento) para o novo problema de encaminhamento seria obtido fazendo

$$Z = \mathbf{c}_B \bar{\mathbf{x}}_B = \mathbf{c}_B \mathbf{B}^{-1} \bar{\mathbf{b}} \quad [4.18]$$

onde  $\mathbf{c}_B$  representa os coeficientes das variáveis básicas na função objectivo [25]. Uma vez aplicadas estas fórmulas duas situações podem surgir.

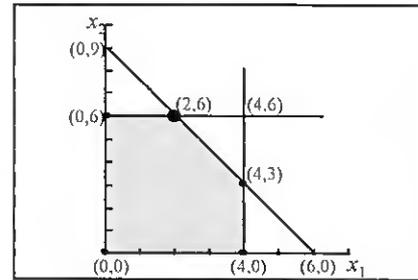
1. Todas as variáveis básicas na nova solução óptima têm valor positivo. Neste caso a nova solução obtida é admissível. Estamos perante o novo óptimo.
2. Existem variáveis básicas na nova solução óptima com valor negativo. Se esta situação surgir a nova solução obtida não é admissível uma vez que as restrições  $x_j \geq 0$  não estão a ser respeitadas. Neste caso deverá proceder-se a algumas iterações do algoritmo Simplex até todas as variáveis básicas terem valor positivo.

Assim, usando a informação final do problema original é possível saber qual o efeito de uma alteração no vector dos termos independentes sem ser necessário resolver o problema de

otimização desde o início. Isto só é possível porque a estrutura do problema se mantém igual.

Como ilustração consideremos o exemplo utilizado no apêndice A.

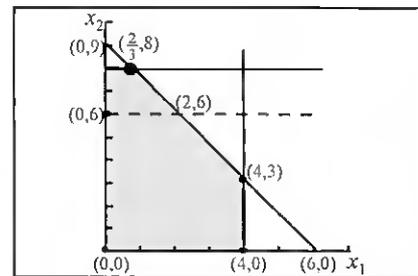
$$\begin{array}{l}
 \text{Maximizar: } Z=3x_1+5x_2 \\
 \text{Sujeito a:} \\
 \quad x_1 \leq 4 \\
 \quad 2x_2 \leq 12 \\
 \quad 3x_1+2x_2 \leq 18 \\
 \quad x_1 \geq 0, x_2 \geq 0
 \end{array}
 \quad [4.19]$$



**Fig. 4.6** - Problema LP exemplo.

A solução ótima encontra-se no vértice (2,6). Suponhamos que o modelo inicial tinha sofrido a seguinte alteração na segunda restrição.

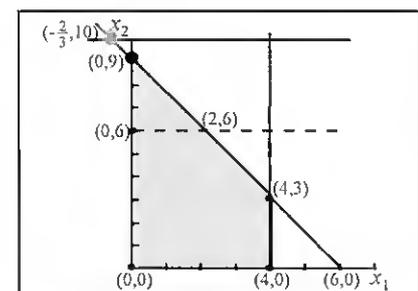
$$\begin{array}{l}
 \text{Maximizar: } Z=3x_1+5x_2 \\
 \text{Sujeito a:} \\
 \quad x_1 \leq 4 \\
 \quad 2x_2 \leq 16 \\
 \quad 3x_1+2x_2 \leq 18 \\
 \quad x_1 \geq 0, x_2 \geq 0
 \end{array}
 \quad [4.20]$$



**Fig. 4.6 a)** - Problema LP exemplo: efeitos de uma alteração ao modelo inicial.

A nova solução ótima encontra-se no vértice  $(\frac{2}{3}, 8)$ . Repare-se que toda a estrutura do modelo permanece igual, houve simplesmente uma ampliação da área de soluções admissíveis. Assim, não é necessário que o algoritmo percorra novamente todos os vértices para encontrar a melhor solução. O vértice ótimo continua a estar no cruzamento das rectas da segunda e terceira restrições. Suponhamos agora que a alteração tinha sido a seguinte:

$$\begin{array}{l}
 \text{Maximizar: } Z=3x_1+5x_2 \\
 \text{Sujeito a:} \\
 \quad x_1 \leq 4 \\
 \quad 2x_2 \leq 20 \\
 \quad 3x_1+2x_2 \leq 18 \\
 \quad x_1 \geq 0, x_2 \geq 0
 \end{array}
 \quad [4.21]$$



**Fig. 4.6 b)** - Problema LP exemplo: efeitos de uma nova alteração ao modelo inicial.

A nova solução óptima está no vértice  $(-\frac{2}{3}, 10)$ . Neste caso estamos perante uma solução não admissível uma vez que a restrição  $x_1 \geq 0$  não é respeitada. Neste caso será necessária apenas uma iteração para que o algoritmo Simplex determine a solução óptima no vértice  $(0, 9)$ , essa sim admissível.

Já foi dito anteriormente que o algoritmo não deve eliminar variáveis artificiais nem restrições redundantes no final da fase 1 do método das duas fases, caso se pretenda fazer análise pos-óptimo. Isto prende-se com o facto de as dimensões das matrizes terem que permanecer intactas para que seja possível realizar as necessárias operações matemáticas entre matrizes.

A listagem seguinte mostra em traços gerais como foi implementada a análise pos-óptimo.

```

Função: AnálisePosOptimo
INICIO
    tipoSolucao=0;
    novoTI=novo vector de termos independentes;
    Binvertido=inversa da matriz básica;
    XB=Binvertido*novoTI;
    FAZ
        varSaida=0;
        varEntrada=0;
        valorVarBasica=0;
        PARA CADA restrição=1...nTotalRestrições FAZ
            SE(XB[restrição]<valorVarBasica) ENTAO
                valorVarBasica=XB[restrição];
                varSaida=restrição;
        FIMSE;
    FIMFAZ;
    SE (varSaida ≠ 0) ENTAO
        varEntrada=EntradaNaBase(outVar);
        SE (varEntrada=0) ENTAO
            tipoSolução=IMPOSSIVEL;
        SENAO
            TrocaVariaveis(varSaida, varEntrada);
            ConstroiBinvertido(varSaida);
            CalculaCN();
            CalculaXB(varSaida);
            CalculaZ();
        FIMSE;
    FIMSE;
    ENQUANTO (varSaida ≠ 0 AND tipoSolução ≠ IMPOSSIVEL)
FIM,

```

**Listagem 4.9** - Algoritmo da análise pos-óptimo.

### 4.3 SUMÁRIO

Foi feita uma descrição de toda a concepção e implementação do sistema de optimização. Para a criação da topologia lógica foram implementados três algoritmos heurísticos: o HLDA, que atribui segmentos lógicos a pares de nós por ordem decrescente de tráfego, o MLDA que cria um par de segmentos lógicos para cada segmento físico e gera os restantes segmentos de acordo com os critérios do HLDA, e o TANC que se preocupa em ligar primeiro todos os nós usando o caminho com menor atraso e gera os restantes segmentos lógicos usando os critérios do HLDA. O TANC foi proposto neste trabalho com o objectivo de corrigir os aspectos negativos do HLDA e MLDA – criação de situações de *loop* e definição do algoritmo apenas para um *degree* lógico superior ou igual ao *degree* físico respectivamente. Foram também implementadas as topologias lógicas regulares *ShuffleNet* e *de Bruijn Graph*.

Foi implementado um algoritmo para formulação do problema de encaminhamento do tráfego e o algoritmo de optimização Simplex Revisto para resolução desse problema de encaminhamento. O Simplex possui as vantagens de o seu código ser conhecido e permitir análise pos-óptimo, o que permitirá reajustar o tráfego de acordo com a matriz de tráfego do momento, evitando níveis de congestionamento muito elevados para tráfego tipo *burst*. Beneficiando destes aspectos foi proposto que uma rede funcionasse em duas fases distintas: determinação da topologia lógica mais adequada à rede física e à matriz de tráfego com redefinição a intervalos relativamente grandes, e encaminhamento do tráfego na topologia lógica com uma redistribuição frequente para responder às variações de tráfego registadas durante a permanência da topologia lógica.

Todo o código foi implementado de raiz usando programação orientada a objectos.

## 5. SIMULAÇÃO DE TRÁFEGO EM REDES DE COMUNICAÇÃO

### 5.1 INTRODUÇÃO

Compreender a natureza do tráfego existente na rede em estudo e seleccionar um modelo adequado são factores que determinam o sucesso da implementação prática de um simulador de redes de comunicação. A modelização de tráfego é, por isso, um elemento chave na simulação de redes de comunicações [29]. Assim, no caso da transmissão ser feita por pacotes, é necessário definir na simulação qual o modelo que melhor descreve a chegada dos pacotes. Os modelos de tráfego que têm sido mais usados em redes de computadores são baseados no processo de *Poisson*. Estes possuem a vantagem de se basearem num processo bem compreendido e matematicamente tratável [30]. Contudo, estudos recentemente feitos nos Laboratórios de Pesquisa da Bellcore provaram que são inadequados para simular o tráfego das redes actuais. Nesse estudo foi observado tráfego de redes LAN tipo Ethernet<sup>14</sup>, tráfego de pacotes RDIS<sup>15</sup> num sistema telefónico público e tráfego gerado por serviços de vídeo VBR<sup>16</sup> [30]. Estas conclusões prendem-se com o facto de o tráfego nas redes flutuar ao longo do tempo e de, mesmo em segmentos pouco utilizados, ser possível que ocorram atrasos devido a congestionamento. Estas flutuações podem ocorrer em períodos de tempo muito curtos dando origem ao conceito de tráfego tipo *burst*. Como os picos de tráfego podem ser 5 vezes superiores à utilização média, um utilizador que queira transmitir dados nesse período poderá ter que esperar bastante.

Um processo de *Poisson*, pelas suas características, não permite a representação deste tipo de tráfego e, por isso, não é consistente com os dados observados no estudo da Bellcore. Neste estudo ficou provado que são necessários processos *renewal* com comportamento *self-similar* para modelar o tráfego existente nas redes de comunicação actuais e futuras [30].

---

<sup>14</sup> Rede local multi-acesso com topologia bus e forma de acesso CSMA/CD para regular o tráfego. A informação nestas redes é enviada em *frames* de tamanho variável que contêm dados e informação de controlo.

<sup>15</sup> Rede Digital com Integração de Serviços. Rede capaz de transmitir de forma totalmente digital dados que vão desde voz a comunicações entre computadores, música e vídeo.

<sup>16</sup> As *frames* VBR (*Variable Bit Rate*) possuem tamanho variável e aleatório.

## 5.2 PROCESSOS *RENEWAL*

Num processo *renewal* os tempos entre eventos sucessivos são independentes e identicamente distribuídos (IID) [31].

**Definição:** Seja  $\{N(t), t \geq 0\}$  um processo de contagem e  $X_n$  o tempo entre os eventos  $(n-1)$  e  $n$  deste processo,  $n \geq 1$ . Se a sequência de variáveis aleatórias não negativas  $\{X_1, X_2, \dots\}$  é independente e identicamente distribuída, então o processo de contagem  $\{N(t), t \geq 0\}$  é chamado processo *renewal*.

Um processo *renewal* é então um processo de contagem em que o tempo que decorre até surgir o primeiro evento tem uma distribuição qualquer  $F$ , o tempo entre o primeiro e o segundo evento, independentemente do tempo do primeiro evento, tem a mesma distribuição  $F$ , etc. Um exemplo de um processo *renewal* é o processo de *Poisson* em que  $F$  é uma distribuição exponencial.

## 5.3 PROCESSOS COM COMPORTAMENTO *SELF-SIMILAR*

Um comportamento *self-similar* tem a ver com o facto de em diferentes escalas temporais existirem comportamentos similares. Por exemplo, um processo de *Poisson* não possui um comportamento *self-similar* uma vez que este vai suavizando à medida que observamos o seu comportamento médio ao longo de grandes períodos. Processos *self-similar* são invariantes no que diz respeito ao período de tempo durante o qual são observados. Quando aplicado a modelos de tráfego, este comportamento significa que em qualquer escala de tempo serão obtidos *bursts*. Este comportamento é um factor fundamental a ter em consideração no desenho, controlo e análise de performance de redes de alta velocidade [32].

Para simular um processo *renewal* que exhibe um comportamento *self-similar* é necessário que os tempos entre eventos tenham variância infinita. A variância descreve quanto da massa probabilística da distribuição está perto da média, ou seja, quanto menor for a variância de uma distribuição mais os seus valores se aproximam do valor médio. Assim, as distribuições que

possuem variância finita têm pouco peso nas caudas, ao contrário das distribuições com variância infinita que têm muito peso nas suas caudas, sendo por isso normalmente chamadas de *heavy tailed* (cauda pesada) [30].

## 5.4 MODELOS ESTATÍSTICOS DE TRÁFEGO

As distribuições que produziram resultados consistentes com os dados observados na Bellcore são a *Pareto* e *t<sub>2</sub>-Student* com média finita e variância infinita [30]. Neste trabalho optou-se pela distribuição *Pareto* por estar implementada na *package* de simulação de redes COMNET III<sup>17</sup> disponível.

### 5.4.1 Distribuição *Pareto*

Esta distribuição é definida pela seguinte função de densidade [33]:

$$f_X(x; x_0, \theta) = \frac{\theta}{x_0} \left( \frac{x_0}{x} \right)^{\theta+1} I_{(x_0, \infty)}(x) \quad [5.1]$$

onde

$$I_{(x_0, \infty)}(x) = \begin{cases} 1, & \text{se } x_0 \leq x < \infty \\ 0, & \text{caso contrario} \end{cases}, \theta > 0 \text{ e } x_0 > 0. \quad [5.2]$$

Uma vez que esta distribuição possui as seguintes propriedades:

- $0 < \theta \leq 1$ , distribuição com média e variância infinitas;
- $1 < \theta \leq 2$ , distribuição com média finita e variância infinita;
- $\theta > 2$ , distribuição com média e variância finitas.

a média e variância são dadas respectivamente por:

$$\frac{\theta x_0}{\theta - 1} \text{ para } \theta > 1 \quad \text{e} \quad \frac{\theta x_0^2}{\theta - 2} - \left( \frac{\theta x_0}{\theta - 1} \right)^2 \text{ para } \theta > 2 \quad [5.3]$$

A influência destes dois parâmetros na distribuição está ilustrada no seguinte gráfico.

<sup>17</sup> Copyright © 1994-1997 CACI Products Company

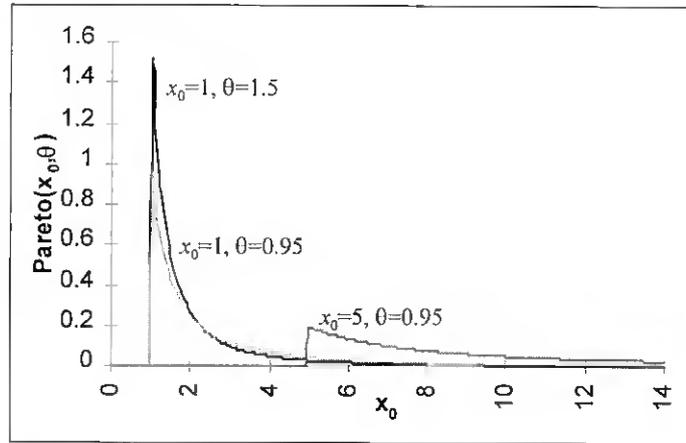


Fig. 5.1 - Distribuição Pareto.

O parâmetro  $x_0$  especifica onde começa a distribuição enquanto o parâmetro  $\theta$  especifica quão rápida é a queda da cauda.

### 5.5 EXEMPLO ILUSTRATIVO

Como exemplo ilustrativo iremos analisar o estudo realizado na rede do centro de computação (*Computer Centre Network*) da Universidade de Auckland em [30]. Na altura das experiências esta rede incluía uma sub-rede que ligava todos os escritórios do pessoal docente e um agrupamento VAX, uma sub-rede com estações Unix, uma sub-rede com os servidores de Netware do centro de computação e outra sub-rede que servia uma sala de PCs com Netware e computadores Macintosh. A figura 5.2 contém amostras retiradas durante 1000s da sub-rede do pessoal docente e na figura 5.3 é mostrado o respectivo histograma.

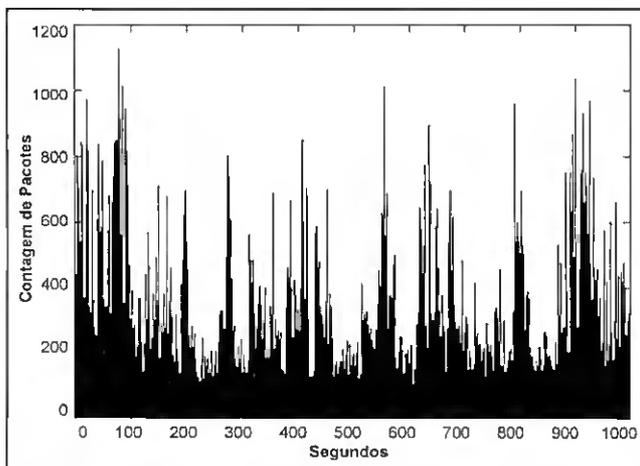


Fig. 5.2 - Amostras da rede do Centro de Computação da Universidade de Auckland.

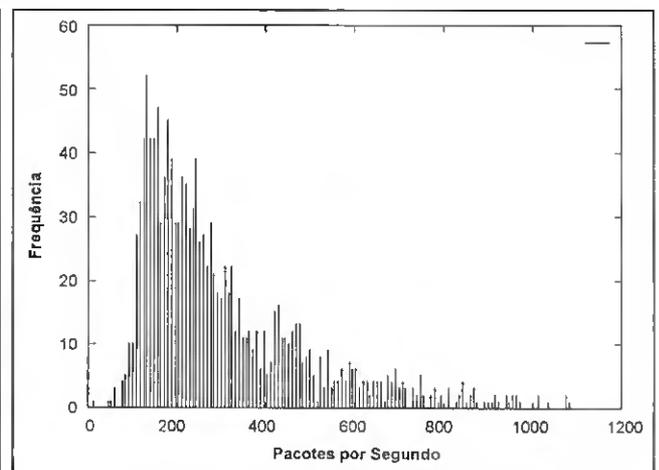


Fig. 5.3 - Histograma das amostras da rede do Centro de Computação da Universidade de Auckland.

Estes gráficos apresentam uma larga gama de fluxos de tráfego que incluem *bursts* em grande número e tamanho. A variância de decaimento lento associada ao comportamento *self-similar* pode ser vista no seguinte gráfico.

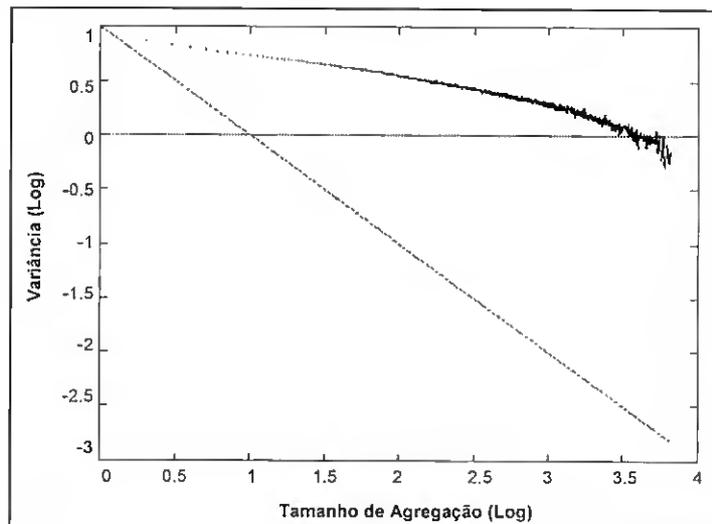


Fig. 5.4 - Variância de decaimento lento das amostras da rede do Centro de Computação da Universidade de Auckland.

O gráfico mostra uma linha de declive  $y = -x$  para referência (o declive teórico de um processo *renewal* genérico é  $y = x^{-1}$ , o que dá  $y = -x$  tomando o logaritmo).

### SIMULAÇÃO DA DISTRIBUIÇÃO DE PARETO

As figuras seguintes mostram os resultados simulados usando a distribuição de *Pareto* e o respectivo histograma.

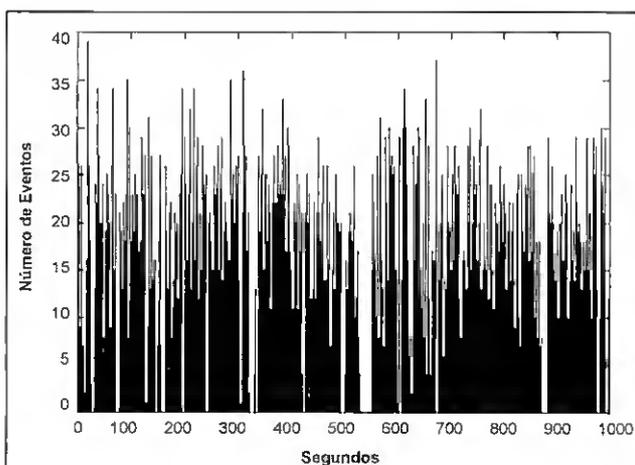


Fig. 5.5 - Simulação da distribuição de *Pareto*.

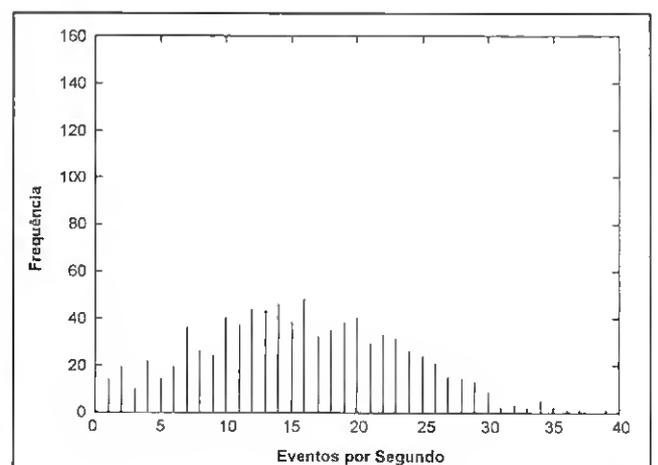


Fig. 5.6 - Histograma da simulação da distribuição de *Pareto*.

O histograma apresenta uma larga gama de níveis de tráfego o que está de acordo com os resultados reais obtidos na rede. O decaimento da variância é o seguinte.

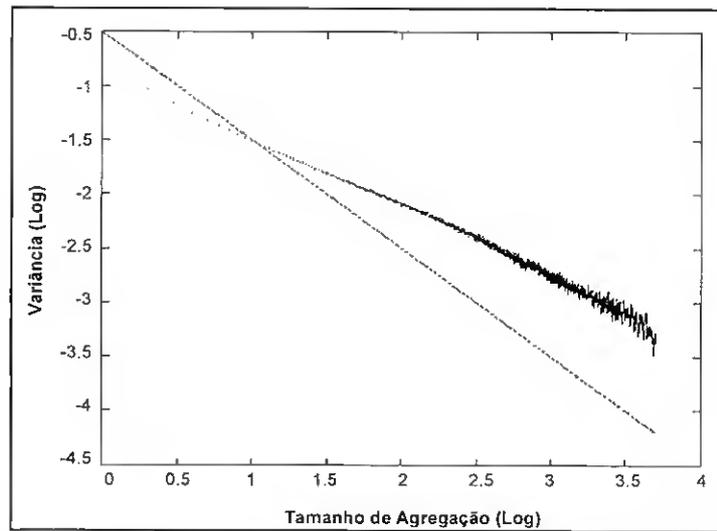


Fig. 5.7 - Variância de decaimento lento da simulação da distribuição de *Pareto*.

O decaimento ocorre a uma taxa inferior a  $x^{-1}$ , verificando-se que os processos *renewal* de variância infinita exibem comportamento *self-similar* [30].

#### SIMULAÇÃO DA DISTRIBUIÇÃO DE POISSON

Foi simulado um processo de *Poisson* com taxa  $\lambda=0.05$  pacotes por milissegundo, simulados durante um período de 1000s. Isto resulta numa chegada de pacotes de distribuição exponencial com  $\mu=20$ . Esta simulação e o seu histograma são mostrados abaixo.

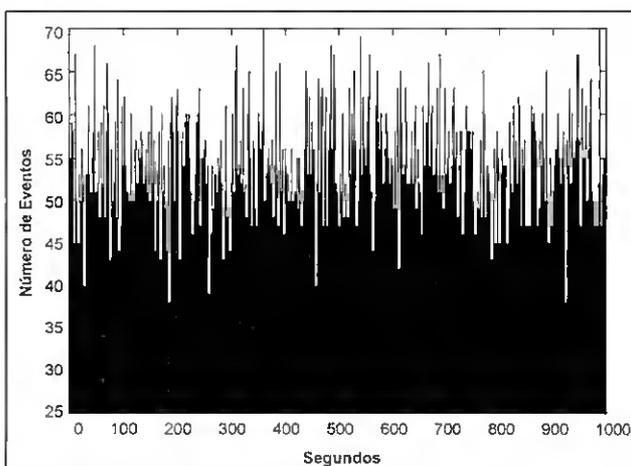


Fig. 5.8 - Simulação da distribuição de *Poisson*.

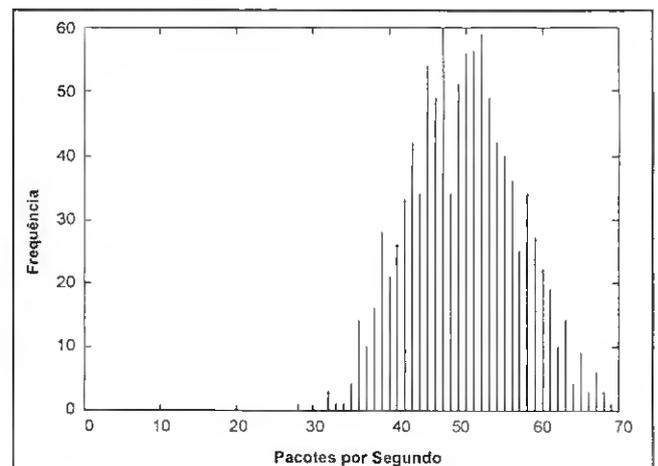


Fig. 5.9 - Histograma da simulação da distribuição de *Poisson*.

O decaimento da variância é o seguinte:

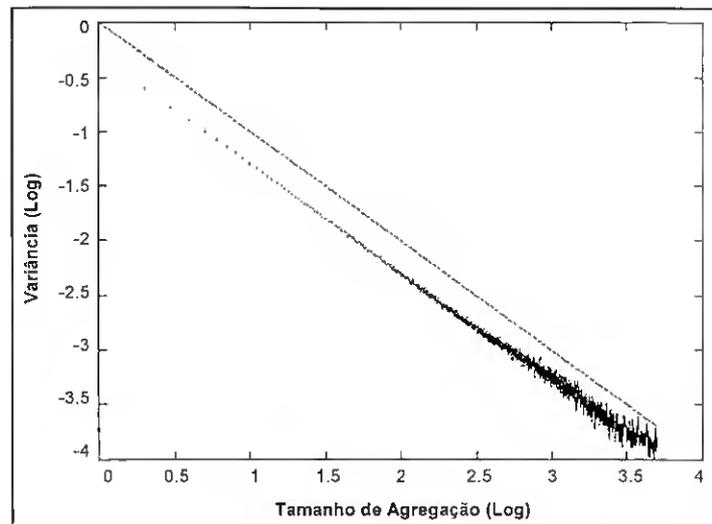


Fig. 5.10 - Variância de decaimento lento da simulação da distribuição de *Poisson*.

O decaimento decresce como uma linha de declive -1, logo a variância decai a uma taxa de  $x^{-1}$ . Confirma-se assim que o processo de *Poisson* não produz resultados consistentes com os obtidos experimentalmente na rede [30].

## 5.6 SUMÁRIO

A escolha do modelo que melhor descreve a chegada dos pacotes é um elemento chave na simulação de redes de comunicação já que determina o sucesso da implementação prática de uma simulação. Estudos recentes da Bellcore mostraram que o processo de *Poisson*, em que se baseiam muitos modelos, é inadequado para simular tráfego de redes, sendo necessários processos *renewal* com comportamento *self-similar* para modelar o tráfego nas redes de comunicações actuais e futuras. Para simular um processo *renewal* que exhibe um comportamento *self-similar* são necessárias distribuições de variância infinita. As distribuições que produziram resultados consistentes com os dados observados na Bellcore são a *Pareto* e *t<sub>2</sub>-Student* com média finita e variância infinita. Neste trabalho optou-se pela distribuição *Pareto*. Neste capítulo os modelos matemáticos do tráfego foram confrontados com tráfego experimental e validados.

## 6. APLICAÇÃO DOS ALGORITMOS DE OPTIMIZAÇÃO A UMA REDE WDM

### 6.1 INTRODUÇÃO

Neste capítulo serão feitos estudos comparativos do desempenho dos vários algoritmos heurísticos descritos no capítulo anterior para desenho da topologia lógica. Os parâmetros de rede que serão utilizados no estudo serão: análise dos níveis de congestionamento obtidos, grau de utilização de comprimentos de onda, tempos de computação e restrições de atraso. A rede física alvo do estudo é a NSFNET, frequentemente usada [22]. A matriz de tráfego utilizada é composta pelos valores médios do tráfego simulado de acordo com a distribuição de *Pareto* ao longo de 1000s. Será estudado o comportamento da rede face aos valores dinâmicos do tráfego ao longo do período considerado (1000 matrizes de tráfego diferentes). O estudo também será estendido às topologias lógicas *ShuffleNet* e de *Bruijn Graph*.

Para a realização das simulações foi usado um processador Alpha 21164 a 466 MHz num DEC 4100 com 512 Mb de memória RAM. Para este efeito o código foi recompilado em Unix usando o *cxx*, compilador de C++ da DEC.

### 6.2 REDE FÍSICA EM ESTUDO

A rede usada nos estudos comparativos de performance foi uma rede de 14 nós conhecida como NSFNET. Esta rede, alvo frequente de estudo, foi escolhida por ter um número razoável de nós. A seguinte figura mostra a estrutura física desta rede incluindo os atrasos relativos associados aos segmentos físicos [22].

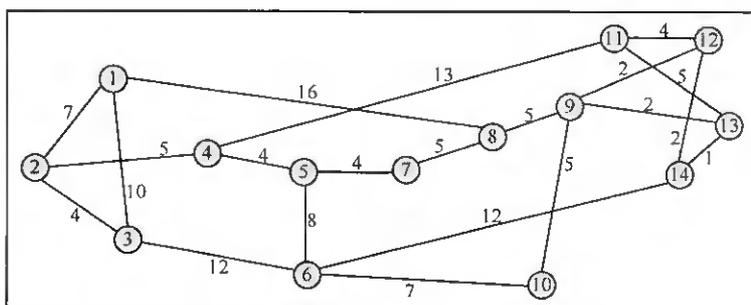


Fig. 6.1 - Rede NSFNET de 14 nós.

### 6.3 MATRIZES DE TRÁFEGO

O cenário de tráfego foi gerado usando a distribuição de *Pareto* uma vez que permite gerar dados com média finita e variância infinita ( $1 < \theta \leq 2$ ), dados consistentes com os observados numa rede real [30]. Trabalhos nesta área utilizam normalmente os valores médios de tráfego entre os nós. Neste trabalho foi usada uma matriz de tráfego que representa a média do tráfego simulado ao longo de 1000s.

ORIGEM \ DESTINO	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1		119	222	2	10	12	10	6	28	67	239	31	14	19
2	11		37	12	13	5	11	31	34	34	87	267	82	10
3	4	2		29	19	20	36	28	29	187	58	16	32	23
4	2	42	90		274	42	34	4	22	15	36	14	4	21
5	39	68	21	247		4	118	2	18	2	10	8	7	3
6	11	40	48	11	28		26	98	241	41	3	3	6	101
7	80	33	4	7	8	443		246	17	10	6	2	102	239
8	5	19	53	20	23	35	29		10	14	17	32	3	19
9	38	87	9	430	9	58	4	55		2	59	70	4	3
10	4	27	10	202	4	4	4	14	20		2	93	28	46
11	109	8	4	6	6	124	3	55	36	12		7	22	17
12	179	6	42	54	5	38	14	12	33	10	6		279	85
13	5	4	15	36	4	4	4	10	13	19	50	4		90
14	47	5	103	33	56	4	5	16	34	11	4	58	7	

Tabela 6.1 - Matriz de tráfego médio simulado ao longo de 1000s.

Por exemplo, a média do fluxo de tráfego que parte do nó 6 e que se destina ao nó 9 é de 241 pacotes por segundo, enquanto que a média do fluxo de tráfego que parte do nó 9 e se destina ao nó 6 é de 58.

## 6.4 ANÁLISE DOS RESULTADOS

### 6.4.1 Níveis de Congestionamento

O problema LP de encaminhado de tráfego tem como objectivo minimizar o congestionamento na topologia lógica previamente escolhida por um algoritmo heurístico. Os níveis de congestionamento obtidos pelas topologias lógicas geradas pelos diferentes algoritmos heurísticos foram os seguintes:

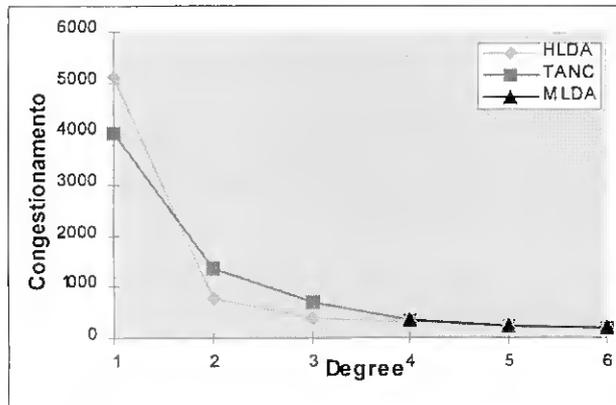


Fig. 6.2 - Níveis de congestionamento dos algoritmos geradores de topologias lógicas.

Degree	HLDA	TANC	MLDA
1	5120,0	4017,00	-
2	788,40	1352,00	-
3	399,00	719,30	-
4	299,25	354,00	337,50
5	239,40	245,30	245,30
6	199,50	199,50	199,50

Tabela 6.2 - Valores de congestionamento dos algoritmos geradores de topologias lógicas.

O algoritmo MLDA está apenas definido para um *degree* lógico superior ou igual ao *degree* físico, neste caso  $degree \text{ lógico} \geq 4$ .

Estes foram os resultados obtidos tanto para uma formulação desagregada como agregada do problema e sem limitações de atraso dos pacotes. Em [24] mostra-se que a forma agregada torna o problema mais leve mas, em contrapartida, apresenta valores de congestionamento maiores que a forma desagregada. Isto tem a ver com o facto de em [24] o problema ser considerado na sua globalidade, isto é, desenho da topologia lógica e encaminhamento do tráfego num só problema MILP. Nas simulações realizadas isto não se verificou uma vez que apenas foi considerado o problema de encaminhamento de tráfego.

O gráfico mostra que para um *degree* de 1 o algoritmo TANC é o que apresenta menor nível de congestionamento. Para um *degree* maior ou igual a 2 o HLDA é o que apresenta melhor performance. O facto de o algoritmo HLDA possuir maior congestionamento para um *degree* de 1 prende-se com o facto de este algoritmo gerar *loops*, isolando os nós intervenientes do resto da rede. As figuras abaixo mostram este aspecto.

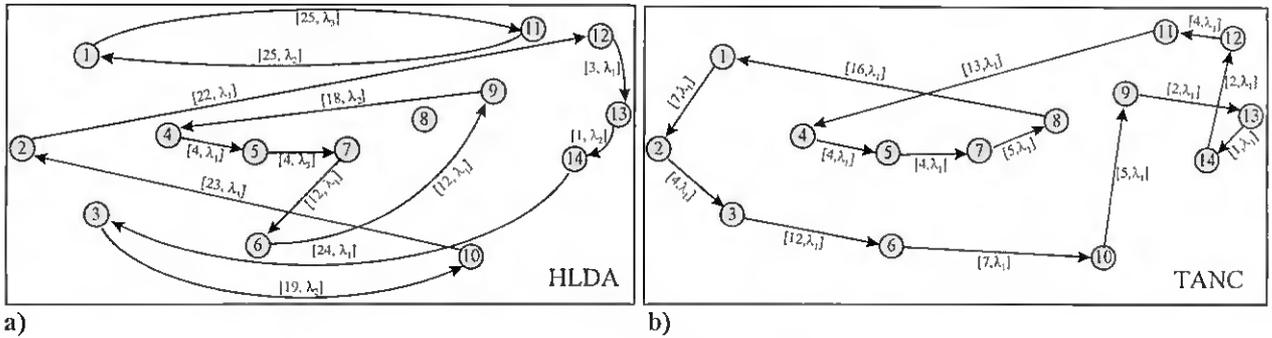


Fig. 6.3 - Topologias lógicas geradas pelos algoritmos para um *degree* igual a 1: a) HLDA; b) TANC.

As figuras mostram, para cada segmento lógico, o atraso de propagação associado e comprimento de onda usado.

Para o *degree* em causa, o algoritmo HLDA gera os seguintes *loops*: [1-11-1], [2-12-13-14-3-10-2] e [4-5-7-6-9-4], responsáveis pelo aumentando do congestionamento na rede. O nó 1, por exemplo, só pode enviar tráfego ao nó 11, ficando incomunicável com os restantes nós. Note-se que o nó 8 ficou isolado da rede por não haver receptores e emissores livres nos restantes nós da rede.

Para um *degree* igual ou superior a 2 o HLDA apresenta menor congestionamento que os outros algoritmos devido ao maior número de segmentos e inexistência de *loops*.

Note-se que, dado que algoritmo HLDA gera a topologia lógica em função da matriz de tráfego, pode acontecer, para uma outra situação de tráfego, não haver qualquer *loop* para um *degree* igual a 1 (implica diminuição do nível de congestionamento) ou haver *loops* para um *degree* igual ou superior a 2 (implica aumento do nível de congestionamento).

**6.4.2 Utilização de Comprimentos de Onda**

O número de comprimentos de onda que podem ser disponibilizados numa rede WDM é limitado. Assim, é importante analisar o seu consumo por parte de cada algoritmo, ou seja, quantos

comprimentos de onda foram necessários para cada algoritmo gerar a sua topologia lógica. O gráfico que se segue mostra essa informação para os vários *degrees*.

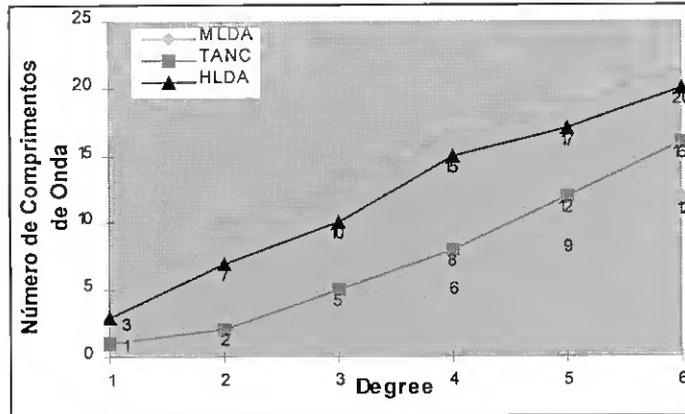


Fig. 6.4 - Consumo de comprimentos de onda dos algoritmos geradores de topologias lógicas.

Dos algoritmos em estudo o HLDA é o que requer maior número de comprimentos de onda. Isto deve-se ao facto de os segmentos lógicos serem atribuídos aos pares com maior tráfego entre si, mesmo que os segmentos físicos usados já possuam muitos comprimentos de onda em uso. Os algoritmos TANC e MLDA exigem menos comprimentos de onda uma vez que os primeiros segmentos lógicos atribuídos são constituídos apenas por um segmento físico.

Para uma rede WDM é sem dúvida importante que o algoritmo gerador da topologia lógica consuma poucos comprimentos de onda, permitindo um maior número de segmentos lógicos e, conseqüentemente, um menor congestionamento. Os gráficos confirmam este facto.

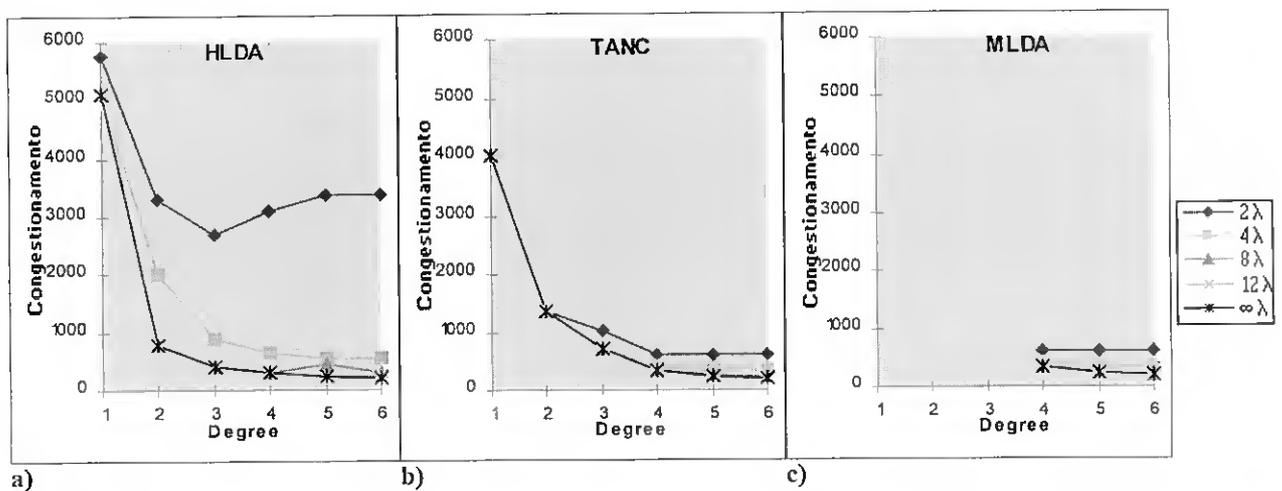


Fig. 6.5 - Níveis de congestionamento obtidos pelos algoritmos geradores de topologias lógicas para diferentes números de comprimentos de onda disponíveis: a) HLDA; b) TANC; c) MLDA.

O algoritmo HLDA, maior consumidor de comprimentos de onda, é realmente o que apresenta maior nível de congestionamento para um número limitado de comprimentos de onda disponíveis. Para um dado *degree* e número de comprimentos de onda disponíveis pequeno o algoritmo TANC apresenta um comportamento muito bom. Os algoritmos TANC e MLDA apresentam um comportamento semelhante para um *degree* superior ou igual a 4. Os gráficos seguintes mostram a performance de cada algoritmo para um número específico de comprimentos de onda disponíveis.

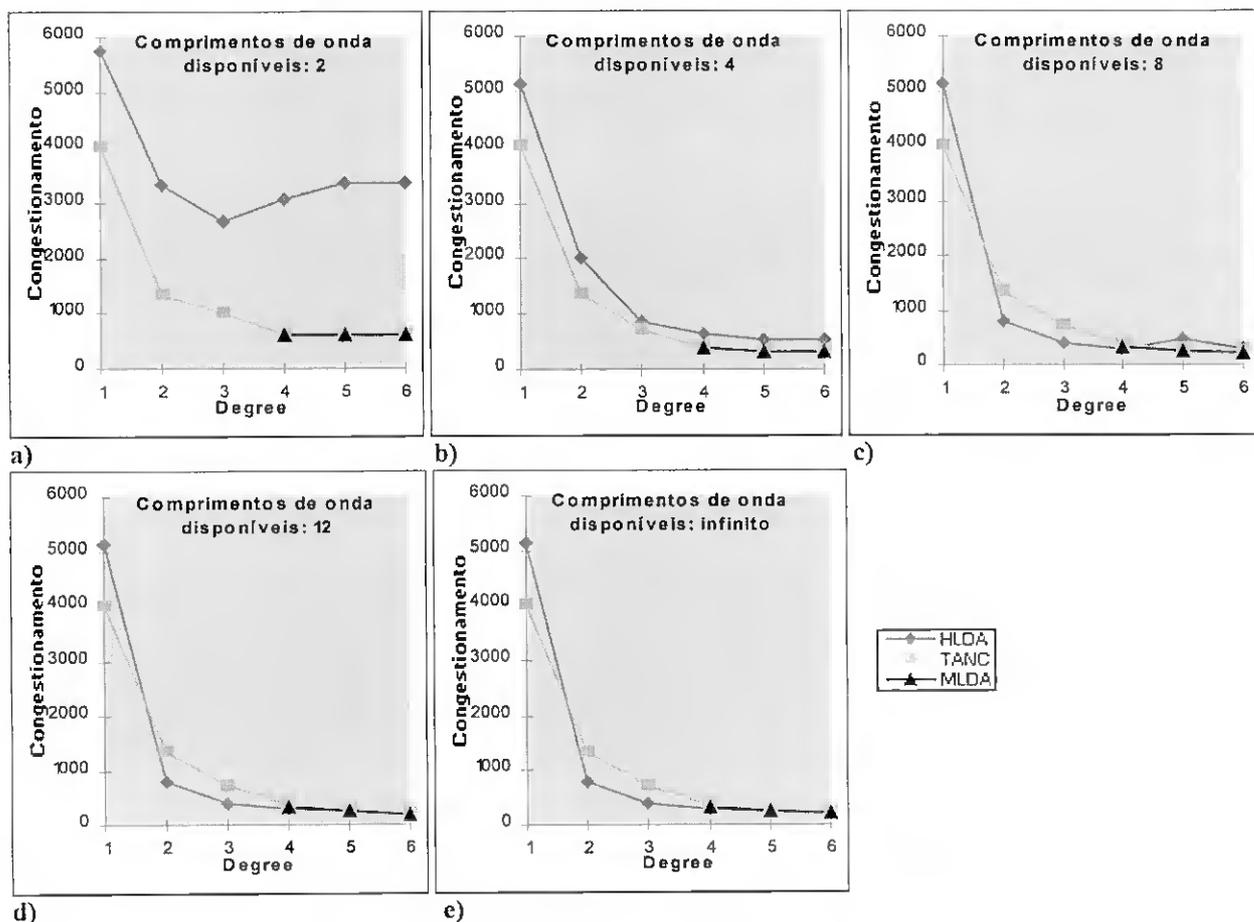


Fig. 6.6 - Níveis de congestionamento obtidos para um número de comprimentos de onda disponível: a) 2; b) 4; c) 8; d) 12; e) infinito.

Para 2 e 4 comprimentos de onda disponíveis o algoritmo HLDA apresenta níveis de congestionamento mais elevados que os obtidos pelos algoritmos TANC e MLDA. Para 8 ou mais comprimentos de onda disponíveis e um *degree* superior ou igual a 2 o algoritmo HLDA é o que possui melhor performance.

### 6.4.3 Tempos de Computação

O tempo de CPU consumido por uma simulação é determinado pelo número de variáveis e restrições do problema LP que, por sua vez, dependem do número de nós da rede, número de segmentos lógicos criados e conectividade escolhida. Para algoritmos que atribuem os segmentos lógicos em função da matriz de tráfego, esta tem influência directa nos tempos de computação obtidos.

Para que fosse possível analisar com rigor o efeito do número de nós e número de segmentos nos tempos de computação, foram feitas medições para um número crescente de nós numa rede física em anel usando uma matriz de tráfego uniforme. O algoritmo usado para gerar a topologia lógica foi o TANC uma vez que, para um *degree* de 1, este cria um anel lógico unindo todos os nós. Para um *degree* de 2 são criados dois anéis lógicos em sentido inverso. Esta conectividade é a ideal para a análise que se pretende uma vez que existe a garantia de todos os nós estarem comunicáveis. Se assim não fosse, como poderia ser o caso de uma topologia gerada pelo algoritmo HLDA, o número de variáveis e restrições do problema, e logo o tempo de computação, dependeria do número de pares de nós comunicáveis e caminhos possíveis entre eles. O gráfico seguinte mostra os tempos obtidos para uma formulação do problema na forma desagregada.

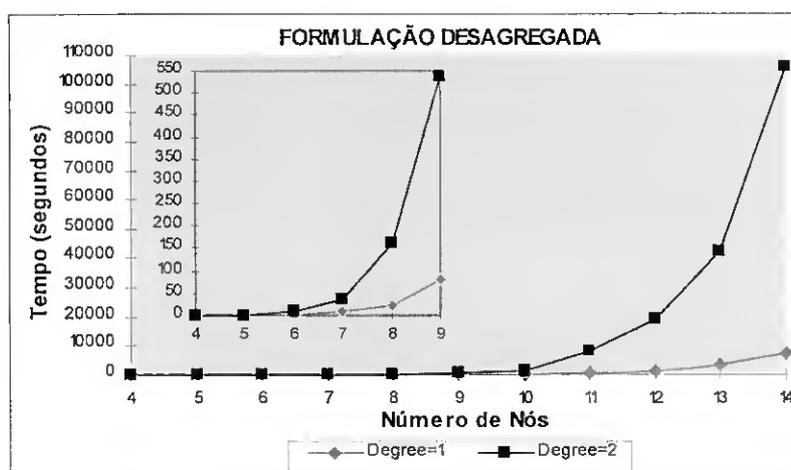


Fig. 6.7 - Tempos de computação para um número crescente de nós numa rede física em anel usando o algoritmo TANC e formulação desagregada.

O número de variáveis e restrições dos respectivos problemas LP são os seguintes:

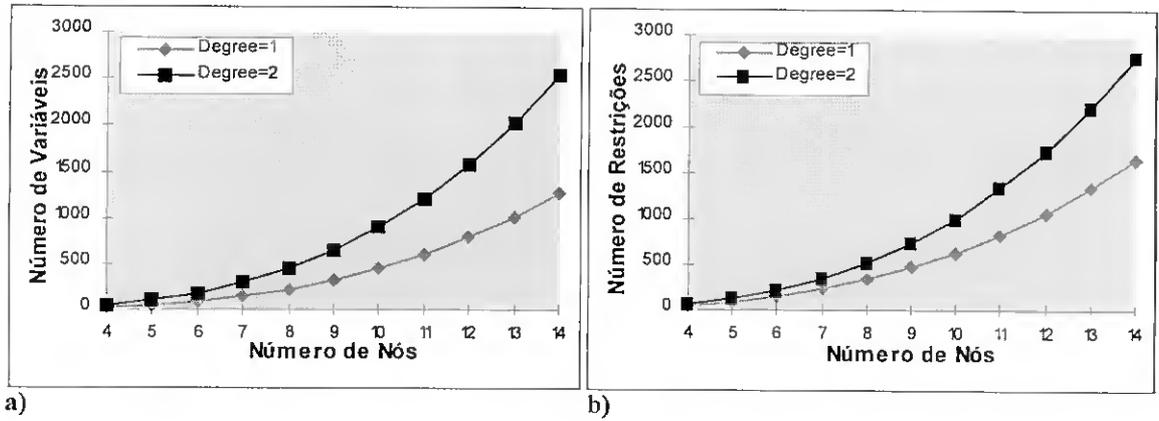


Fig. 6.8 - Problema LP para um número crescente de nós numa rede física em anel usando o algoritmo TANC e tipo de formulação desagregada: a) Variáveis; b) Restrições.

O número de variáveis do problema LP pode ser representado pela expressão:

$$nV = \left( \frac{x^3}{2} - \frac{x^2}{2} \right) \times Degree \tag{6.1}$$

Ou seja, o número de variáveis para o *degree* 2 é igual ao dobro das obtidas para o *degree* 1. O número de restrições para um *degree* de 1 e 2 pode ser representado respectivamente pelas expressões:

$$nR_1 = \frac{x^3}{2} + \frac{3x^2}{2} - x - 1 \tag{6.2}$$

$$nR_2 = x^3 + x - 1 \tag{6.3}$$

Os 3 polinómios anteriores foram obtidos por interpolação de Lagrange e são muito úteis para uma previsão do grau de complexidade de um problema. Os gráficos abaixo mostram uma previsão do número de variáveis e restrições até 40 nós.

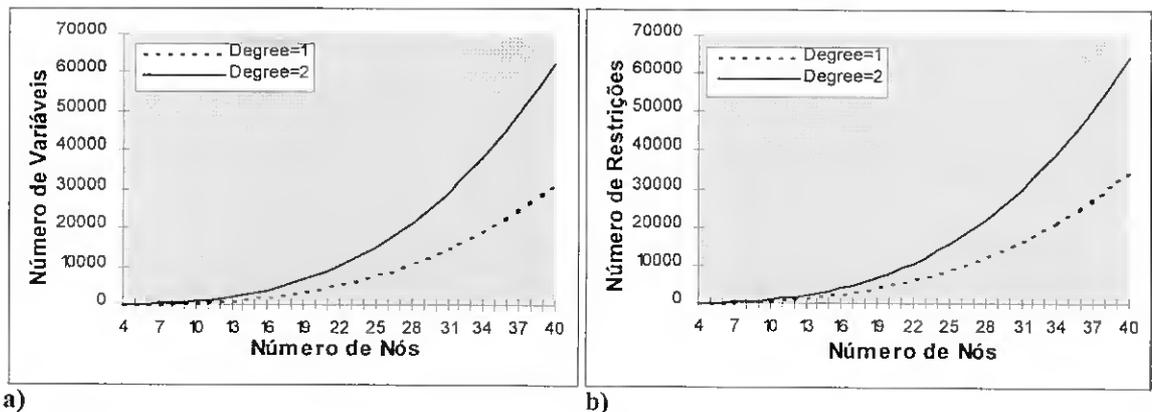


Fig. 6.9 - Previsão do número de variáveis e restrições do problema LP para um número crescente de nós numa rede física em anel usando o algoritmo TANC e tipo de formulação desagregada: a) Variáveis; b) Restrições.

Os mesmos dados para uma formulação agregada são os seguintes:

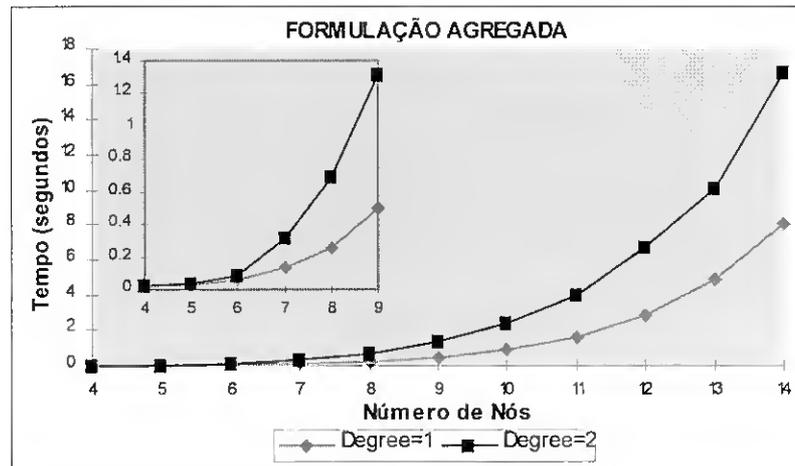


Fig. 6.10 - Tempos de computação para um número crescente de nós numa rede física em anel usando o algoritmo TANC e formulação agregada.

O número de variáveis e restrições dos respectivos problemas LP são os seguintes:

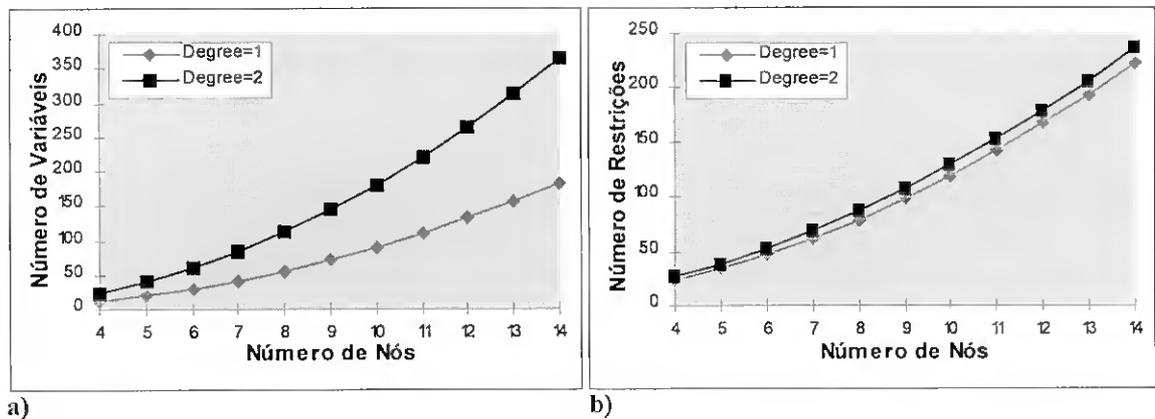


Fig. 6.11 - Problema LP para um número crescente de nós numa rede física em anel usando o algoritmo TANC e tipo de formulação agregada: a) Variáveis; b) Restrições.

O número de variáveis do problema LP na forma agregada pode ser representado pela expressão:

$$nV = (x^2 - x) \times Degree \tag{6.4}$$

O número de variáveis para o *degree* 2 é também igual ao dobro das obtidas para o *degree* 1. O número de restrições para um *degree* de 1 e 2 pode ser representado respectivamente pelas expressões:

$$nR_1 = x^2 + 2x - 1 \tag{6.5}$$

$$nR_2 = x^2 + 3x - 1 \tag{6.6}$$

Uma previsão do grau de complexidade do problema até 40 nós é ilustrado de seguida.

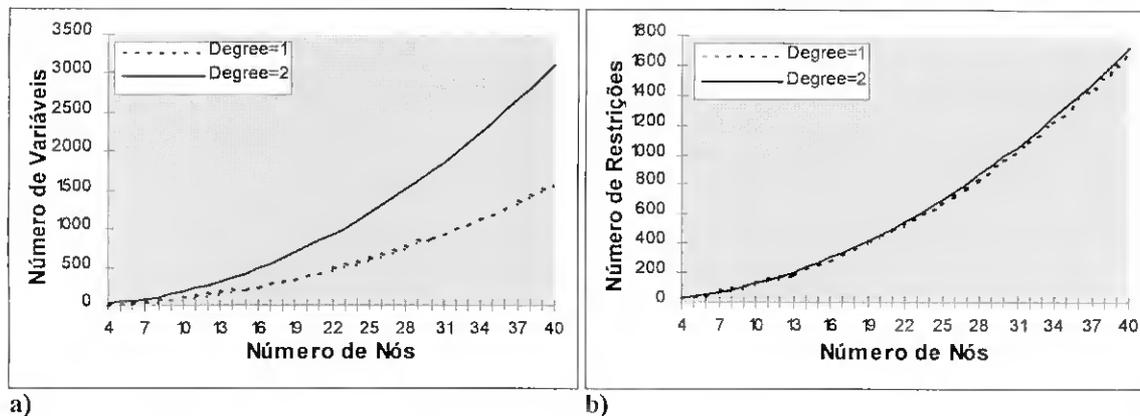


Fig. 6.12 - Previsão do número de variáveis e restrições do problema LP para um número crescente de nós numa rede física em anel usando o algoritmo TANC e tipo de formulação agregada: a) Variáveis; b) Restrições.

Repare-se na grande diferença do número de variáveis e restrições dos problemas LP para as abordagens agregada e desagregada, e conseqüentemente no tempo de computação. A abordagem agregada é sem dúvida muito mais tratável.

Para a rede NSFNET em estudo foram obtidos os seguintes tempos:

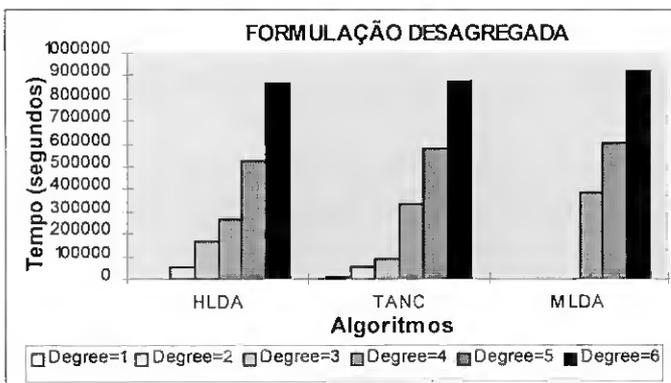


Fig. 6.13 - Tempos de computação obtidos nos estudos feitos na NSFNET usando formulação desagregada.

Degree	HLDA	TANC	MLDA
1	00:00:07	01:41:14	-
2	15:30:46	14:20:23	-
3	1d e 21:29:24	23:42:43	-
4	3d e 02:01:06	3d e 19:41:18	4d e 10:20:13
5	6d e 03:16:40	6d e 17:10:05	6d e 23:27:13
6	10d e 02:13:02	10d e 04:42:08	10d e 15:23:01

Tabela 6.3 - Valores dos tempos de computação obtidos nos estudos feitos na NSFNET usando formulação desagregada.

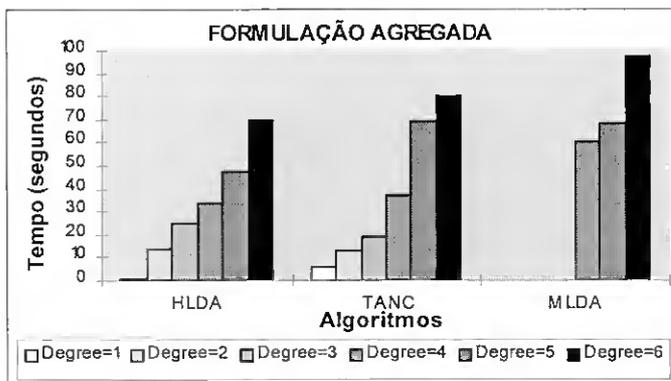


Fig. 6.14 - Tempos de computação obtidos nos estudos feitos na NSFNET usando formulação agregada.

Degree	HLDA	TANC	MLDA
1	00:00:01	00:00:6	-
2	00:00:14	00:00:13	-
3	00:00:25	00:00:19	-
4	00:00:34	00:00:37	00:01:00
5	00:00:47	00:01:09	00:01:08
6	00:01:10	00:01:20	00:01:37

Tabela 6.4 - Valores dos tempos de computação obtidos nos estudos feitos na NSFNET usando formulação agregada.

#### 6.4.4 Restrições de Atraso

Como já foi referido anteriormente a abordagem escolhida neste trabalho tem como objectivo minimizar o congestionamento na rede, restringindo ao mesmo tempo o atraso associado a cada par fonte-destino. As simulações feitas para análise do congestionamento não impõem um limite ao atraso que um par fonte-destino pode sofrer. Esta opção pode resultar em encaminhamentos de tráfego com longos atrasos. Desta forma é analisada em seguida a performance dos algoritmos para várias restrições de atraso.

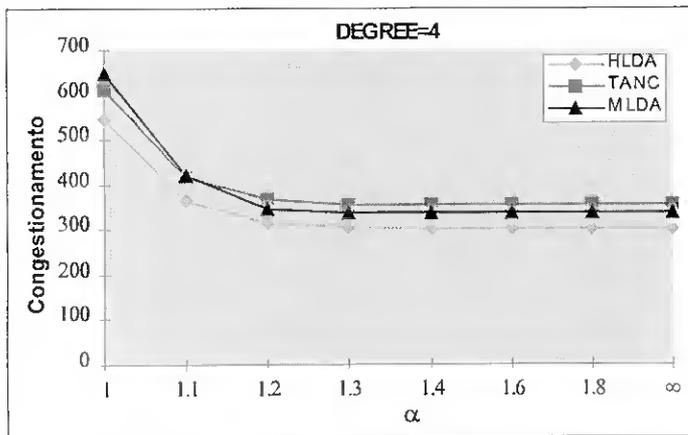


Fig. 6.15 - Níveis de congestionamento para um *degree* de 4 e várias restrições de atraso.

$\alpha$	HLDA	TANC	MLDA
1	544.70	610.10	647.20
1,1	361.10	415.90	419.10
1,2	315.70	369.50	347.20
1,3	304.50	356.00	338.20
1,4	299.25	355.30	337.50
1,6	299.25	354,00	337.50
1,8	299.25	354,00	337.50
$\infty$	299,25	354,00	337.50

Tabela 6.5 - Valores dos níveis de congestionamento para um *degree* de 4 e várias restrições de atraso.

De lembrar que o atraso permitido a um par  $(s,d)$  está limitado a  $\alpha$  vezes o menor atraso possível entre esse par,  $\alpha \geq 1$ . Ou seja, quanto mais  $\alpha$  se aproxima de 1 mais tráfego terá que fluir no caminho com menor atraso. Esta parece ser a melhor forma de limitar o atraso já que cada par  $(s,d)$  terá o seu próprio limite, de certa forma ajustado à natureza do tráfego. Isto é, o tráfego local (curta distância entre a origem e o destino) terá uma restrição de atraso obrigatoriamente menor que o tráfego destinado a longas distâncias. Se em vez disso a restrição de atraso máximo fosse igual para toda a rede poderia acontecer que tráfego não chegasse ao seu destino apesar de existir caminho possível. Pelo facto de os limites serem valores relativos ao menor atraso possível de cada par  $(s,d)$ , a única informação que o gráfico anterior nos dá é que uma restrição de atraso mais apertada resulta num aumento do congestionamento, não sendo possível aferir qualquer informação sobre os atrasos

registados pelo tráfego. O gráfico seguinte mostra qual o atraso médio sofrido por um pacote para diferentes restrições de atraso.

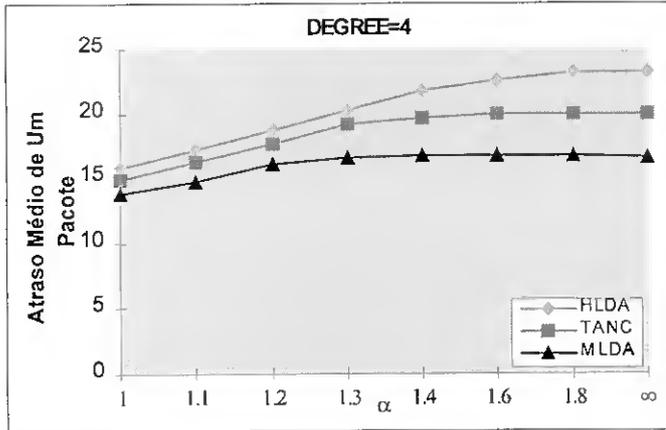


Fig. 6.16 - Atraso médio de um pacote para um *degree* de 4 e várias restrições de atraso.

$\alpha$	HLDA	TANC	MLDA
1	15,76	14,90	13,86
1,1	17,17	16,23	14,73
1,2	18,73	17,70	16,07
1,3	20,30	19,18	16,61
1,4	21,73	19,69	16,79
1,6	22,47	19,91	16,71
1,8	23,12	19,88	16,71
$\infty$	23,09	19,95	16,64

Tabela 6.6 - Valores dos atrasos médios de um pacote para um *degree* de 4 e várias restrições de atraso.

A topologia gerada pelo algoritmo HLDA parece ser a que origina maior atraso dos pacotes. Isto deve-se ao facto de os segmentos lógicos serem atribuídos aos nós sem existir uma consideração dos atrasos a eles associados. Os algoritmos TANC e MLDA apresentam menores atrasos pelo facto de os algoritmos começarem por atribuir segmentos lógicos constituídos apenas por um segmento físico. Note-se que para o *degree* em estudo o algoritmo HLDA, apesar de apresentar os maiores valores de atraso, é o que obtém menores valores de congestionamento.

#### 6.4.5 Performance da Rede usando Redistribuição Frequente do Tráfego

Como a rede vai ser confrontada não com os valores médios de tráfego usados na optimização mas sim com valores que variam ao longo do período considerado (1000s neste caso), torna-se importante analisar a performance da rede durante esse período. Para esse efeito foi usada análise pos-óptimo, análise feita após obtenção de uma solução óptima para a versão inicial do modelo, de forma a determinar a redistribuição de tráfego na topologia lógica conforme a matriz de tráfego do momento (uma das 1000 matrizes). Dito de outra forma, o encaminhamento será ajustado de forma óptima às variações de tráfego registadas. A grande vantagem desta abordagem é que, como foi demonstrado em 4.3.4, usando informação final do modelo original, nenhuma ou

apenas algumas iterações serão necessárias para se refazer a otimização do problema, agora modificado, e se obter uma nova solução ótima. Se a análise pos-ótimo não fosse usada, a distribuição de tráfego inicialmente escolhida manter-se-ia para todos os 1000s. Para redes com tráfego tipo *burst* esta situação podia provocar níveis de congestionamento bastante elevados. Assim, haverá sempre uma situação ótima no que diz respeito ao encaminhamento do tráfego. Nos gráficos que se seguem a análise pos-ótimo é aplicada à formulação agregada.

As redes lógicas geradas pelos algoritmos HLDA e TANC apresentam o seguinte comportamento para um *degree* igual a 1.

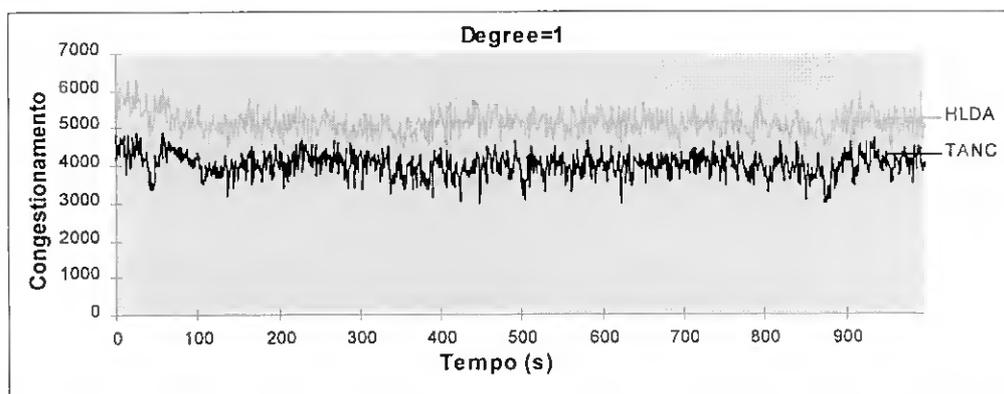


Fig. 6.17 - Comportamento das topologias lógicas geradas pelos algoritmos HLDA e TANC durante o período de 1000s para um *degree* igual a 1.

Os seguintes histogramas mostram a frequência dos valores de congestionamento obtidos ao longo dos 1000s.

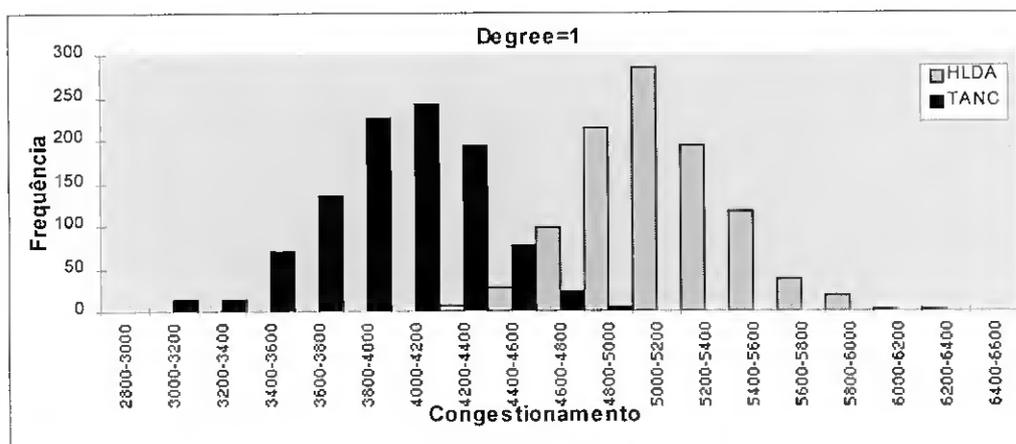


Fig. 6.18 - Histogramas dos comportamento das topologias lógicas geradas pelos algoritmos HLDA e TANC durante o período de 1000s para um *degree* igual a 1.

O algoritmo TANC apresenta de facto níveis de congestionamento bastante menores que o HLDA. O maior congestionamento que estes podem atingir é de 5000 para o algoritmo TANC e 6400 para o algoritmo HLDA.

As figuras que se seguem mostram a mesma informação para um *degree* igual a 2.

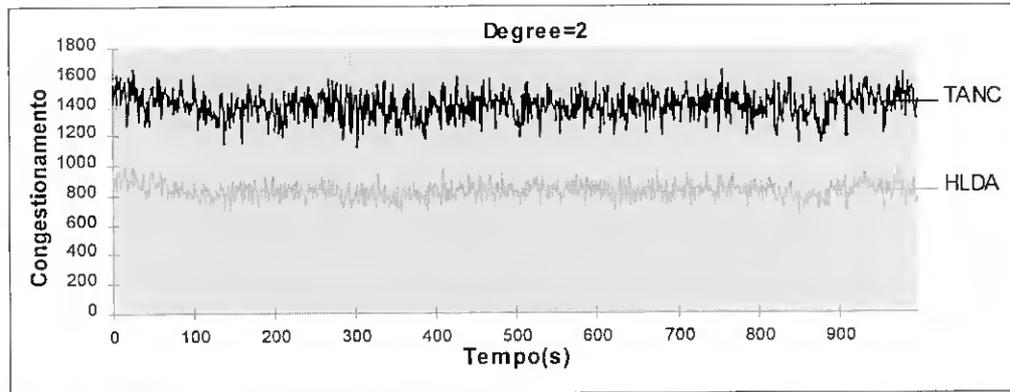


Fig. 6.19 - Comportamento das topologias lógicas geradas pelos algoritmos HLDA e TANC durante o período de 1000s para um *degree* igual a 2.

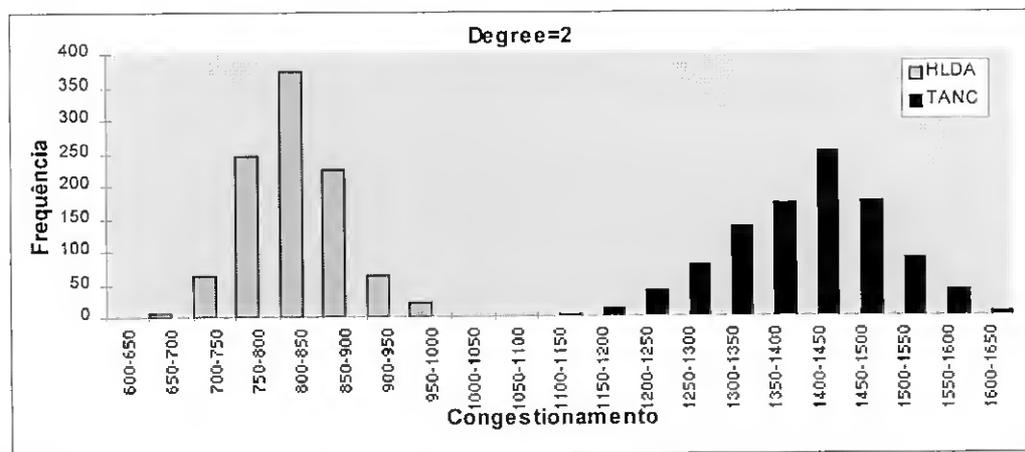


Fig. 6.20 - Histogramas dos comportamento das topologias lógicas geradas pelos algoritmos HLDA e TANC durante o período de 1000s para um *degree* igual a 2.

Para um *degree* de 2 o algoritmo HLDA é o que apresenta menores níveis de congestionamento. Este algoritmo apresenta também um comportamento mais estável, isto é, menor variância.

Para um *degree* igual a 6 foram obtidos os seguintes comportamentos:

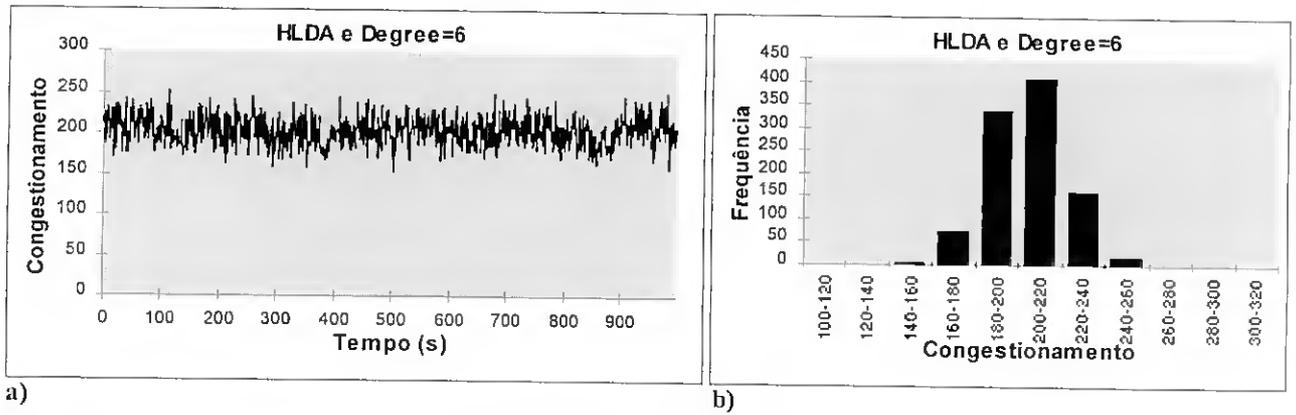


Fig. 6.21 - Comportamento da topologia lógica gerada pelo HLDA durante o período de 1000s para um *degree* igual a 6: a) Dados; b) Histograma.

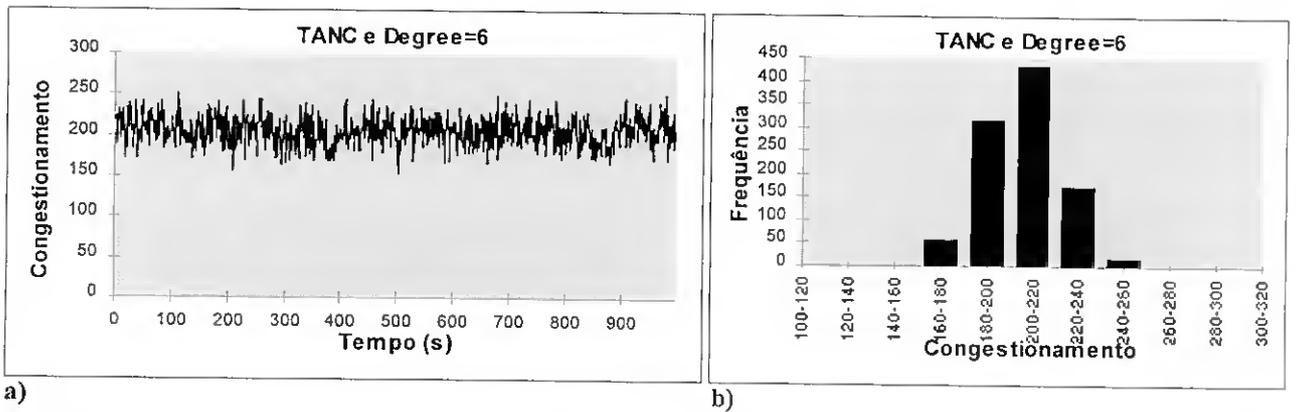


Fig. 6.22 - Comportamento da topologia lógica gerada pelo TANC durante o período de 1000s para um *degree* igual a 6: a) Dados; b) Histograma.

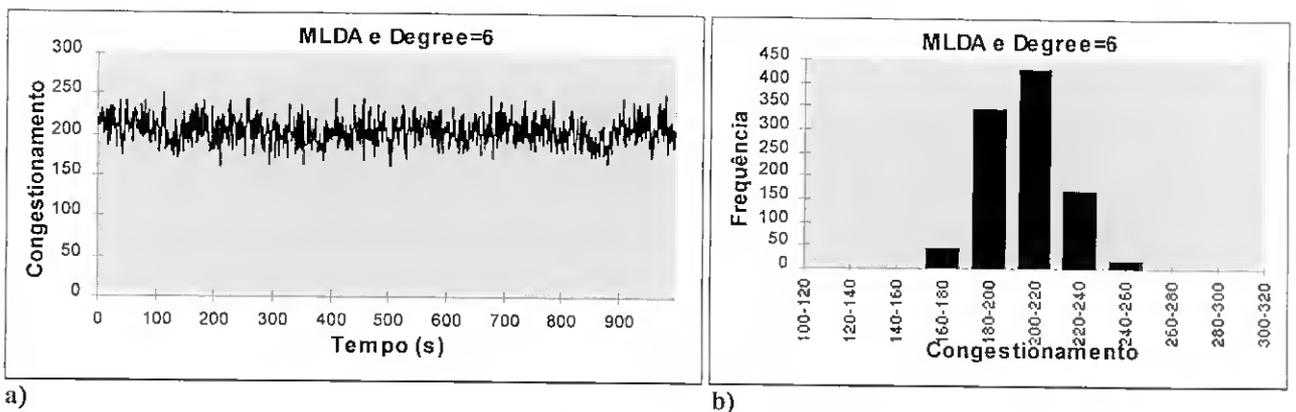


Fig. 6.23 - Comportamento da topologia lógica gerada pelo MLDA durante o período de 1000s para um *degree* igual a 6: a) Dados; b) Histograma.

O gráfico seguinte compara os três histogramas anteriores.

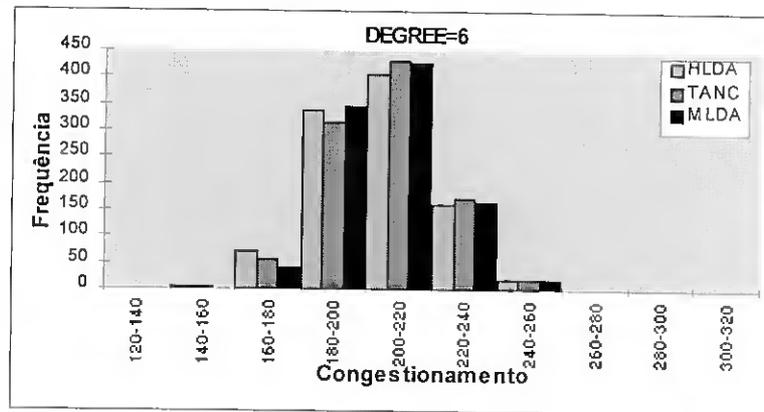


Fig. 6.24 - Comparação dos histogramas do comportamento das topologias lógicas geradas pelos algoritmos HLDA, TANC e MLDA durante o período de 1000s para um *degree* igual a 6.

Este histograma comparativo permite concluir que, apesar dos níveis de congestionamento obtidos usando a matriz de tráfego médio serem idênticos, o algoritmo HLDA apresenta uma performance melhor já que apresenta uma menor frequência para níveis de congestionamento elevados.

Os tempos de computação da análise pos-ótimo foram os seguintes:

<i>Degree</i>	HLDA	TANC	MLDA
1	00:00:12	00:01:13	-
2	00:23:18	00:09:14	-
3	01:20:26	00:32:29	-
4	02:18:08	01:53:47	02:11:46
5	03:23:17	03:21:18	04:09:07
6	04:54:00	05:31:05	06:20:25

Tabela 6.7 - Valores dos tempo de computação da análise pos-ótimo para um período de 1000s.

De salientar que estes tempos de computação correspondem ao cálculo da análise pos-ótimo para 1000 diferentes matrizes de tráfego. Ou seja, é o tempo gasto para calcular 1000 novas soluções ótimas. A título de exemplo, enquanto que para um *degree* de 2 o algoritmo HLDA demorou 14 segundos no cálculo da solução óptima (ver tabela 6.4), para o mesmo *degree* e algoritmo, a análise pos-ótimo encontrou, em média, um novo óptimo em cada  $(23m:18s)/1000 = 1,4$  segundos.

#### 6.4.6 Topologias Lógicas *ShuffleNet* e de *Bruijn Graph*

Nesta secção são comparadas as performances dos algoritmos geradores de topologias lógicas e das topologias lógicas regulares *ShuffleNet* e de *Bruijn Graph*. Como o número de nós nas redes

*Shuffle* e de *Bruijn* é determinado por dois parâmetros, não é possível usar um número arbitrário de nós, o número de nós usado tem que ser um número permitido por ambos os algoritmos. Desta forma foi usada a seguinte rede de 8 nós para estudo das topologias lógicas *shuffle* ( $p=2, k=2$ ) e de *Bruijn* ( $\delta=2, D=3$ ) de *degree* igual a 2.

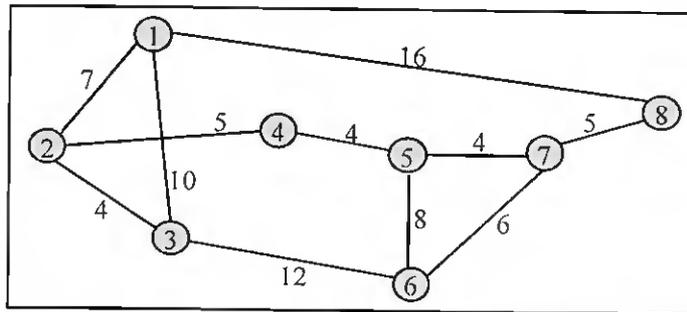


Fig. 6.25 - Topologia física para estudo das topologias lógicas *shuffle* ( $p=2, k=2$ ) e de *Bruijn* ( $\delta=2, D=3$ ).

Das matrizes de tráfego usadas uma contém parte do tráfego da rede NSFNET e a outra tráfego uniforme. Esta última contém valores muito próximos da média da matriz de tráfego não uniforme (que usa a distribuição *Pareto*).

ORIGEM \ DESTINO	1	2	3	4	5	6	7	8
1		119	222	2	10	12	10	6
2	11		37	12	13	5	11	31
3	4	2		29	19	20	36	28
4	2	42	90		274	42	34	4
5	39	68	21	247		4	118	2
6	11	40	48	11	28		26	98
7	80	33	4	7	8	443		246
8	5	19	53	20	23	35	29	

a)

ORIGEM \ DESTINO	1	2	3	4	5	6	7	8
1		51	52	51	51	51	52	51
2	51		51	51	52	52	52	52
3	52	52		50	51	51	52	52
4	52	52	50		50	52	52	52
5	52	52	52	51		52	52	52
6	52	51	51	51	52		52	52
7	52	52	52	50	51	52		52
8	52	52	52	51	51	51	52	

b)

Tabela 6.8 - Matrizes de tráfego médio para as topologias lógicas: a) *ShuffleNet*; b) *de Bruijn Graph*.

Os níveis de congestionamento sem limites de atraso para ambos os tráfegos são mostrados no seguinte gráfico.

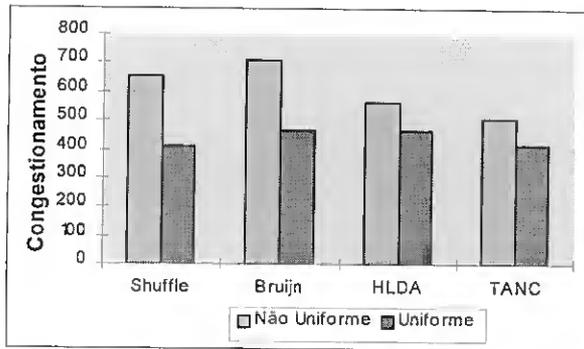


Fig. 6.26 - Níveis de congestionamento para uma rede de 8 nós e dois tipos de tráfego.

Topologias	Tráfego não uniforme	Tráfego uniforme
Shuffle	650,5	413,5
Bruijn	713,0	466,5
HLDA	566,0	464,5
TANC	505,0	415,0

Tabela 6.9 - Valores de congestionamento para uma rede de 8 nós e dois tipos de tráfego.

Os resultados obtidos revelam que as topologias lógicas regulares *Shuffle* e *Bruijn* não têm bom desempenho para um tráfego tipo *burst*. Para tráfego uniforme a topologia *Shuffle* apresenta o melhor resultado.

O consumo de comprimentos de onda foi o seguinte:

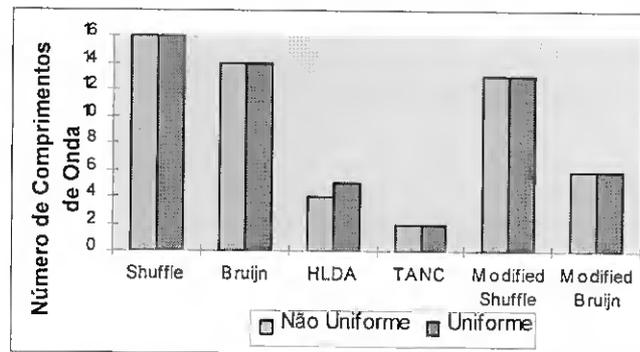


Fig. 6.27 - Consumo de comprimentos de onda para rede de 8 nós e dois tipos de tráfego.

O gráfico mostra também os algoritmos *Modified Shuffle* e *Modified Bruijn* que correspondem aos algoritmos *ShuffleNet* e *deBruijn* com reutilização de comprimentos de onda. Isto é, enquanto os originais reservam um comprimento de onda para cada segmento lógico, estes foram alterados de forma a reutilizarem comprimentos de onda já utilizados noutros segmentos físicos desde que não haja sobreposição. Mesmo com reutilização de comprimentos de onda os algoritmos *ShuffleNet* e *deBruijn* são os que consomem mais comprimentos de onda. O algoritmo HLDA apresenta um ligeiro aumento no consumo de comprimentos de onda para tráfego uniforme.

## 6.5 SUMÁRIO

A rede física em estudo é uma rede de 14 nós conhecida como NSFNET. O cenário de tráfego representa a média dos dados simulados usando a distribuição de *Pareto* ao longo de 1000s.

Assumindo que o número de comprimentos de onda é ilimitado e que não existem limitações no atraso que um pacote pode sofrer, as experiências mostram que a topologia gerada pelo algoritmo HLDA é a que apresenta menores níveis de congestionamento, excepto para um *degree* igual a 1 em que o menor congestionamento é conseguido pelo algoritmo TANC. Isto tem a ver com o facto de o HLDA gerar *loops* para um *degree* de 1. Como este algoritmo gera a topologia lógica em função da matriz de tráfego, para uma outra situação de tráfego poderá não haver qualquer *loop* para um *degree* igual a 1 (implica diminuição do nível de congestionamento) ou haver *loops* para um *degree* igual ou superior a 2 (implica aumento do nível de congestionamento). Assim, podem obter-se diferentes resultados para outras simulações.

Quanto ao grau de consumo de comprimentos de onda, o algoritmo HLDA é o maior consumidor. Para poucos comprimentos de onda disponíveis o HLDA apresenta níveis de congestionamento elevados devendo optar-se pelo algoritmo TANC ou MLDA.

O tempo de computação consumido por uma simulação é determinado pelo número de variáveis e restrições do problema LP que, por sua vez, dependem do número de nós da rede, número de segmentos lógicos criados e conectividade escolhida. As experiências revelam que, usando o algoritmo Simplex, os tempos de computação aumentam exponencialmente com a dimensão do problema. Entre a formulação agregada e desagregada ficou demonstrada a existência de uma desproporção considerável, sendo a abordagem agregada bem mais tratável.

Relativamente ao comportamento face a restrições de atraso confirmou-se que quanto mais se limita o atraso que os pacotes podem sofrer, maiores são os níveis de congestionamento obtidos. Isto é previsível já que os segmentos com menor atraso de propagação ficarão mais sobrecarregados. A topologia gerada pelo algoritmo HLDA é aquela em que os pacotes sofrem maior atraso. Se este for um aspecto importante deverá optar-se pela topologia TANC ou MLDA.

Confrontando a rede com os valores de tráfego usados para calcular a matriz média (1000 “fotografias” diferentes do tráfego), a rede apresenta oscilações que devem ser tomadas em consideração já que podem atingir eventualmente níveis incontroláveis. Para um *degree* igual a 6, apesar dos níveis de congestionamento obtidos para a matriz de tráfego médio serem idênticos, o algoritmo HLDA apresenta a melhor performance ao longo do período considerado.

No estudo das topologias *ShuffleNet* e *de Bruijn Graph* concluiu-se que estas são indicadas para tráfego uniforme e que, comparativamente com os outros algoritmos, consomem muitos comprimentos de onda, sendo inadequadas para redes *multi-hop* com tráfego tipo *burst*.

## 7. CONCLUSÃO

A técnica WDM, ao permitir um aumento drástico da capacidade dos sistemas de comunicação, surge como uma das tecnologias mais promissoras para a próxima geração de redes de alta velocidade. Quando aliada a componentes sintonizáveis permite embutir uma conectividade lógica numa rede física, conectividade lógica essa que pode ser alterada dinamicamente em resposta a variações de tráfego. Sendo certo que existem limitações físicas e económicas que condicionam o seu desenho, também surgem soluções para ultrapassar esses condicionamentos, sendo de especial interesse as redes *multi-hop* que, ao usarem nós intermédios, permitem reduzir substancialmente o número de emissores e/ou receptores necessários na rede.

O problema de desenho da topologia lógica e encaminhamento do tráfego numa rede *multi-hop*, pode ser formulado como um problema MILP cujo objectivo é minimizar o congestionamento (resultando numa maximização do *throughput*), abordagem que se mostrou intratável para problemas de grandes dimensões. A abordagem seguida neste trabalho dividiu o problema em duas partes: desenho da topologia lógica, resolvido com algoritmos heurísticos, e encaminhamento do tráfego, formulado como um problema de programação linear. A vantagem da abordagem adoptada é a rapidez de resolução e o facto de permitir análise pos-óptimo que possibilita um encaminhamento óptimo quase permanente perante alterações frequentes do tráfego. Concluiu-se ainda que a formulação desagregada e agregada apresentam níveis de congestionamento idênticos, sendo preferível o uso da última já que é de mais rápida execução.

Pelo exposto, propõe-se que uma rede funcione em duas fases distintas: determinação da topologia lógica mais adequada à rede física e à matriz de tráfego com redefinição a intervalos relativamente grandes, e encaminhamento do tráfego na topologia lógica com uma

redistribuição frequente para responder às variações de tráfego registadas durante a permanência da topologia lógica.

Para a realização das simulações foi usada a rede física NSFNET e um cenário de tráfego que representa a média dos dados simulados usando a distribuição de *Pareto*, uma das distribuições que permite simular um processo *renewal* de comportamento *self-similar* adequados à representação de tráfego em redes de comunicações. Destes estudos concluiu-se ainda que a escolha de um algoritmo deverá depender da realidade em causa e dos parâmetros considerados como críticos para uma determinada rede de comunicações. A utilização de análise pos-óptimo revelou-se bastante importante já que permite a obtenção de uma nova solução óptima em relativamente pouco tempo.

Como propostas para desenvolvimento futuro, é de considerar que alguma investigação se dedique ao aperfeiçoamento de algoritmos para distribuição de tráfego que permitam obter vantagens semelhantes às da aplicação da análise pos-óptimo. Desse modo, alguma atenção deverá ser dada ao desenvolvimento da algoritmia de optimização aplicada a redes, de modo a reduzir os tempos de computação consumidos e a alargar o âmbito da sua aplicação a redes mais complexas. É desejável que outras disciplinas como a Investigação Operacional, as Redes Neurais ou os Algoritmos Genéticos possam trazer as suas potencialidades ao aliciente domínio das telecomunicações.

## APÊNDICE A – OS MÉTODOS SIMPLEX E SIMPLEX REVISTO

### O SIMPLEX REVISTO

Para uma melhor compreensão do algoritmo Simplex, considere-se o seguinte problema e a sua respectiva visualização gráfica [25]:

Maximizar: $Z=3x_1+5x_2$  Sujeito a: $x_1 \leq 4$ $2x_2 \leq 12$ $3x_1+2x_2 \leq 18$  $x_1 \geq 0, x_2 \geq 0$
---

[A.1]

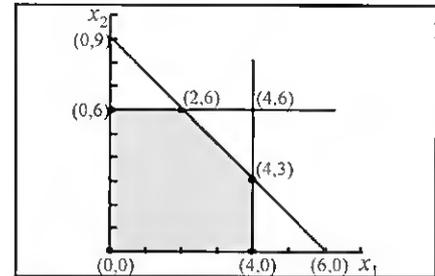


Fig. A.1 - Exemplo de um problema LP.

Como este problema tem somente duas variáveis, e por isso duas dimensões, é possível representá-lo graficamente. A região a cinzento representa o conjunto de soluções admissíveis onde os vértices possuem as seguintes propriedades:

1. Se existe uma solução ótima ela tem que ser um destes vértices.
2. Se existem várias soluções ótimas, então pelo menos duas têm que ser vértices adjacentes.
3. O número de vértices é finito.
4. Se um vértice não possui vértices adjacentes melhores (valor de  $Z$ ) então está-se perante a solução ótima.

O algoritmo Simplex explora estas quatro propriedades movendo-se apenas para um vértice adjacente que apresenta uma melhor solução, até ficar posicionado num vértice que tem melhor solução que os seus adjacentes (solução ótima). Em traços gerais, o algoritmo Simplex executa as seguintes etapas:

- 1- *Inicialização*. Começar num vértice.
- 2- *Iteração*. Mover para um melhor vértice. Repetir quantas vezes for necessário.
- 3- *Teste de Optimização*. O vértice corrente é o óptimo se os seus adjacentes têm um valor menor em  $Z$ .

**Inicialização**

Tanto para manipulação algébrica como para identificação dos vértices, torna-se mais simples lidar com equações do que com desigualdades. Desta forma, o primeiro passo é transformar todas as restrições em igualdades. Isto é possível através do uso de **variáveis de afastamento**. Considerando a restrição,

$$x_1 \leq 4, \tag{A.2}$$

a variável de afastamento para esta restrição é

$$x_3 = 4 - x_1, \tag{A.3}$$

e, logo,

$$x_1 + x_3 = 4. \tag{A.4}$$

A restrição original  $x_1 \leq 4$  mantém-se sempre que  $x_3 \geq 0$ . Assim,  $x_1 \leq 4$  é equivalente ao conjunto de restrições:  $x_1 + x_3 = 4$  e  $x_3 \geq 0$ . Se este procedimento for usado para o problema apresentado atrás, o modelo equivalente fica:

Maximizar: $Z - 3x_1 - 5x_2 = 0$  Sujeito a: $x_1 \quad \quad + x_3 \quad \quad = 4$ $\quad \quad 2x_2 \quad + x_4 \quad = 12$ $3x_1 + 2x_2 \quad \quad + x_5 = 18$  $x_j \geq 0, \text{ para } j=1,2,\dots,5$	[A.5]
---	-------

A este modelo é dado o nome de **forma aumentada** do problema. Um vértice admissível (que respeite as restrições) representado na forma aumentada é chamado de **solução básica admissível** (por exemplo, o vértice (0,6) na forma aumentada ficaria (0,6,4,0,6)).

Repare-se que neste novo modelo existem mais variáveis (5) que restrições (apenas 3). Isto dá-nos dois graus de liberdade uma vez que é possível atribuir um valor arbitrário a duas dessas variáveis com vista a resolver as 3 equações em ordem às 3 variáveis restantes. O Simplex usa zero como valor arbitrário. As variáveis a quem o Simplex atribui o valor zero são chamadas de **variáveis não básicas**, e as outras são chamadas de **variáveis básicas** (por exemplo, a solução básica admissível  $(0,6,4,0,6)$  tem como variáveis não básicas  $x_1$  e  $x_4$ ).

O primeiro vértice a ser considerado, solução básica admissível inicial, é a origem. Ou seja, para o exemplo anterior,  $(0,0,4,12,18)$ . Note-se que o motivo porque esta solução pode ser lida imediatamente prende-se com o facto de em cada restrição existir apenas uma variável básica com um coeficiente de 1. Adiante será visto que, ao mover-se de um vértice para outro, o Simplex encarrega-se (usando eliminação Gaussiana) de colocar as equações nesta forma conveniente.

### *Iteração*

Em cada iteração, o Simplex move-se da solução básica admissível corrente para uma solução básica admissível adjacente que seja melhor. Este movimento implica trocar uma variável não básica (**variável de entrada**) por uma variável básica (**variável de saída**) e identificar a nova solução básica admissível. Colocam-se, então, as seguintes questões:

1. **Qual a variável de entrada?** As candidatas a variável de entrada são todas as variáveis não básicas. Como os coeficientes de cada variável na função objectivo são as taxas a que  $Z$  irá variar à medida que cada variável muda, e como a nova solução básica admissível deverá resultar numa melhoria do valor da função objectivo (maior  $Z$  se o objectivo é maximizar e menor  $Z$  se o objectivo é minimizar), a variável de entrada é a que possui maior coeficiente (positivo se o objectivo é maximizar e negativo se o objectivo é minimizar) e por isso a taxa mais rápida.

2. **Qual a variável de saída?** As candidatas a variável de saída são todas as variáveis básicas. Quando o método Simplex escolhe uma variável de entrada, a interpretação geométrica é que está a escolher um dos segmentos, arestas, emanando do actual vértice. Aumentar esta variável a partir do zero corresponde a percorrer essa aresta. Quando a primeira variável básica chegar a zero estamos perante a variável de saída. Desta forma, a variável de saída é aquela cuja restrição impõe o menor limite superior de quanto a variável de entrada pode crescer. Para o exemplo que temos visto, se  $x_2$  é a variável de entrada:

Variável Básica	Equação	Limite Superior	
$x_3$	$x_3=4-x_1$	Não tem limite	
$x_4$	$x_4=12-2x_2$	$x_2 \leq \frac{12}{2} = 6 \leftarrow$ <b>Menor</b>	[A.6]
$x_5$	$x_5=18-3x_1-2x_2$	$x_2 \leq \frac{18}{2} = 9$	

Ou seja, na primeira restrição  $x_3$  fica sempre igual a 4 uma vez que  $x_1$  é zero (variável não básica), não impondo qualquer restrição ao valor que  $x_2$  pode tomar. Relativamente a  $x_4$  e  $x_5$ , estes impõem um limite superior a  $x_2$  para evitar que  $x_4$  e  $x_5$  tomem valores negativos e, por isso, não admissíveis. A variável de saída é então  $x_4$ .

3. **Como é identificada a nova solução básica admissível?** Uma vez achada a variável de entrada e a variável de saída, só é necessário achar os novos valores para as restantes variáveis básicas (isto porque, geometricamente, percorrer a aresta até encontrar a primeira variável básica a zero implica alterar o valor de todas as variáveis básicas). Para uma leitura directa da solução, o Simplex coloca o novo sistema de equações na forma conveniente (restrições com uma variável básica por equação com coeficiente 1, e função objectivo só em função das variáveis não básicas) usando a eliminação Gausseana. Para mais detalhes consultar [25].

*Teste de Optimização*

Para determinar se uma solução básica admissível é ótima a função objectivo é escrita somente em função das variáveis não básicas e com  $Z$  em evidência. A solução é ótima se todos os coeficientes da função objectivo são não positivos ( $\leq 0$ ) e o objectivo é maximizar, ou não negativos ( $\geq 0$ ) e o objectivo é minimizar. Isto porque escolher qualquer destas variáveis como variável de entrada só ia piorar a solução. Para o exemplo que temos vindo a seguir, na 2ª iteração o problema tem o seguinte aspecto na forma conveniente:

<p>Maximizar: <math>Z + \frac{3}{2}x_4 + x_5 = 36</math></p> <p>Sujeito a:</p> $x_3 + \frac{1}{3}x_4 - \frac{1}{3}x_5 = 2$ $x_2 + \frac{1}{2}x_4 = 6$ $x_1 - \frac{1}{3}x_4 + \frac{1}{3}x_5 = 2$ $x_j \geq 0, \text{ para } j=1,2,\dots,5$	[A.7]
---	-------

sendo  $x_4$  e  $x_5$  as variáveis não básicas. A função objectivo com  $Z$  em evidência e só em função das variáveis não básicas ficaria:  $Z = 36 - \frac{3}{2}x_4 - x_5$ . Como o objectivo é maximizar a função e nenhum dos coeficientes das variáveis não básicas é positivo, esta é a solução ótima. Uma vez que o problema está na forma conveniente, e como o Simplex atribui zero às variáveis não básicas, a solução pode ser obtida directamente:  $x_1 = 2, x_2 = 6$  ficando  $Z = 36$ .

**O SIMPLEX REVISTO**

A forma algébrica descrita anteriormente não é a mais eficiente para implementação em computador porque em cada iteração são feitos cálculos e são armazenadas variáveis desnecessárias. O Simplex Revisto surgiu para resolver o mesmo tipo de problemas usando a mesma filosofia, só que de forma mais eficiente quando executado em computador.

Como o Simplex Revisto usa manipulações matriciais é necessário descrever a notação utilizada. Para representar matrizes são usadas letras maiúsculas realçadas (por exemplo: **A**),

para vectores são usadas letras minúsculas realçadas (por exemplo: **b**) e letras em itálico para escalares (por exemplo:  $x_1$ ).

Usando matrizes, o modelo de programação linear geral fica na seguinte forma:

Maximizar: $Z = \mathbf{c}\mathbf{x}$ Sujeito a: $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ e $\mathbf{x} \geq \mathbf{0}$	[A.8]
--	-------

onde **c** é o vector linha

$$\mathbf{c} = [c_1, c_2, \dots, c_n] \quad [A.9]$$

**x**, **b** e **0** são os vectores coluna

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}, \quad \mathbf{0} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad [A.10]$$

e **A** é a matriz

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \dots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}. \quad [A.11]$$

Para obter a forma aumentada do problema é introduzido o vector coluna de variáveis de afastamento

$$\mathbf{x}_s = \begin{bmatrix} x_{n+1} \\ x_{n+2} \\ \vdots \\ x_{n+m} \end{bmatrix}, \quad [A.12]$$

de forma que as restrições ficam na seguinte forma:

$$[\mathbf{A}, \mathbf{I}] \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_s \end{bmatrix} = \mathbf{b} \quad \text{e} \quad \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_s \end{bmatrix} \geq \mathbf{0}, \quad [A.13]$$

onde **I** é a matriz identidade  $m \times n$  e o vector nulo **0** tem agora  $(n+m)$  elementos.

Como já foi referido, a abordagem do método Simplex é obter uma sequência de soluções básicas admissíveis até ser atingido o óptimo. A característica chave do Simplex

Revisto tem a ver com a forma como estas soluções básicas são calculadas após serem identificadas as variáveis básicas e não básicas. Uma vez conhecidas, a solução básica é a solução das  $m$  equações

$$[A, I] \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_s \end{bmatrix} = \mathbf{b}, \quad [A.14]$$

em que as  $n$  variáveis não básicas dos  $n+m$  elementos de

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{x}_s \end{bmatrix} \quad [A.15]$$

são colocadas a zero. Como estas  $n$  variáveis podem ser eliminadas, ficam  $m$  equações com  $m$  incógnitas (as variáveis básicas) que podem ser representadas por:

$$\mathbf{B}\mathbf{x}_B = \mathbf{b}, \quad [A.16]$$

onde o vector das variáveis básicas

$$\mathbf{x}_B = \begin{bmatrix} x_{B1} \\ x_{B2} \\ \vdots \\ x_{Bm} \end{bmatrix}, \quad [A.17]$$

é obtido eliminando as variáveis não básicas de

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{x}_s \end{bmatrix}, \quad [A.18]$$

e a matriz básica

$$\mathbf{B} = \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1m} \\ B_{21} & B_{22} & \cdots & B_{2m} \\ \vdots & \vdots & \cdots & \vdots \\ B_{m1} & B_{m2} & \cdots & B_{mm} \end{bmatrix}, \quad [A.19]$$

é obtida eliminando as colunas de  $[A, I]$  que dizem respeito às variáveis não básicas. Como  $\mathbf{B}^{-1}$  existe sempre, é possível multiplicar ambos os lados da expressão  $\mathbf{B}\mathbf{x}_B = \mathbf{b}$  por  $\mathbf{B}^{-1}$  ficando:

$$\mathbf{B}\mathbf{x}_B = \mathbf{b} \Leftrightarrow \mathbf{B}^{-1}\mathbf{B}\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b} \quad [A.20]$$

Como  $\mathbf{B}^{-1}\mathbf{B} = \mathbf{I}$ , a solução desejada para as variáveis básicas é:

$$\boxed{\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b}} \quad [A.21]$$

Se  $\mathbf{c}_B$  for um vector cujos coeficientes são os coeficientes da função objectivo que correspondem aos elementos de  $\mathbf{x}_B$ , o valor da função objectivo para esta solução básica é

$$Z = \mathbf{c}_B \mathbf{x}_B = \mathbf{c}_B \mathbf{B}^{-1} \mathbf{b} \quad [\text{A.22}]$$

Desta forma, em qualquer iteração, apenas é necessário saber quais são as variáveis básicas (após determinar qual a variável de entrada e de saída) e calcular  $\mathbf{B}^{-1}$ , para se obter o estado de todas as outras variáveis. Isto reduz, sem dúvida, o número de cálculos a executar e a quantidade de informação a armazenar em cada iteração.

Relativamente ao cálculo de  $\mathbf{B}^{-1}$  em cada iteração este pode ser feito usando uma função que inverta uma matriz. Contudo, uma vez que  $\mathbf{B}$  (e, conseqüentemente,  $\mathbf{B}^{-1}$ ) muda tão pouco de uma iteração para a outra, é muito mais eficiente calcular o novo  $\mathbf{B}^{-1}$  ( $\mathbf{B}_{\text{new}}^{-1}$ ) a partir do  $\mathbf{B}^{-1}$  da iteração anterior ( $\mathbf{B}_{\text{old}}^{-1}$ ). Para descrever como este cálculo é efectuado, são necessárias as seguintes definições:

$x_k$  = variável básica de entrada,

$a'_{ik}$  = coeficiente de  $x_k$  na restrição corrente  $i$  para  $i=1,2,\dots,m$ ,

$r$  = número da equação contendo a variável básica de saída.

O elemento na linha  $i$  e coluna  $j$  de  $\mathbf{B}_{\text{new}}^{-1}$  é:

$$(\mathbf{B}_{\text{new}}^{-1})_{ij} = \begin{cases} (\mathbf{B}_{\text{old}}^{-1})_{ij} - \frac{a'_{ik}}{a_{rk}} (\mathbf{B}_{\text{old}}^{-1})_{rj}, & \text{se } i \neq r \\ \frac{1}{a_{rk}} (\mathbf{B}_{\text{old}}^{-1})_{rj}, & \text{se } i = r \end{cases} \quad [\text{A.23}]$$

Em notação matricial fica:

$$\mathbf{B}_{\text{new}}^{-1} = \mathbf{E} \mathbf{B}_{\text{old}}^{-1}, \quad [\text{A.24}]$$

onde  $\mathbf{E}$  é a matriz identidade com a coluna número  $r$  substituída pelo vector:

$$\eta = \begin{bmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_m \end{bmatrix}, \text{ onde } \eta_i = \begin{cases} -\frac{a'_{ik}}{a_{rk}}, & \text{se } i \neq r \\ \frac{1}{a_{rk}}, & \text{se } i = r \end{cases} \quad [\text{A.25}]$$

Para além das vantagens já apresentadas e não menos importante do que aquelas, o Simplex Revisto permite controlar os inevitáveis problemas de arredondamento produzidos pelos computadores, calculando periodicamente  $\mathbf{B}^{-1}$  através da inversão directa de  $\mathbf{B}$ .

**O MÉTODO DAS DUAS FASES**

No exemplo simples que tem sido apresentado, após inseridas as variáveis de afastamento, o problema fica imediatamente na chamada forma conveniente, isto é, apenas uma variável básica por restrição com um coeficiente de 1. Mas como foi dito anteriormente, é legítimo o problema apresentar restrições de igualdade (=) e superioridade ( $\geq$ ), ou serem eliminadas as restrições de não negatividade das variáveis de decisão. Nestas situações o problema na forma aumentada pode não ficar na forma conveniente. Suponhamos, para ilustração, o seguinte exemplo e a sua correspondente forma aumentada:

Maximizar: $Z=3x_1+5x_2$  Sujeito a: $x_1 \leq 4$ $2x_2=12$ $3x_1+2x_2 \geq 18$  $x_1 \geq 0, x_2 \geq 0$
--

[A.26]

Maximizar: $Z-3x_1-5x_2=0$  Sujeito a: $x_1 + x_3 = 4$ $2x_2 = 12$ $3x_1+2x_2 - x_4 = 18$  $x_j \geq 0, \text{ para } j=1,2,\dots,5$
---

[A.27]

Infelizmente o problema na forma aumentada não se encontra na forma conveniente. Nestes casos é usada a técnica das **variáveis artificiais**<sup>18</sup>. Esta técnica introduz uma variável artificial em cada restrição que dela necessite. No exemplo, a segunda e terceira restrição ficariam:

$$\begin{aligned} 2x_2 + \bar{x}_5 &= 12 \\ 3x_1+2x_2 - x_4 + \bar{x}_6 &= 18 \end{aligned} \quad [\text{A.28}]$$

e  $\bar{x}_5 \geq 0, \bar{x}_6 \geq 0$

<sup>18</sup> As variáveis artificiais possuem uma barra superior.

Esta forma do problema é chamada **forma revista**. O efeito da introdução de uma variável artificial é expandir a região admissível. Uma solução admissível para o problema na forma revista é também solução para o problema original se as variáveis artificiais possuírem valor zero. Se fosse permitido ao algoritmo Simplex proceder normalmente até achar uma solução para o problema revista, no final não haveria garantia dessa solução ser a ótima para o problema original. Para que a solução do problema revista esteja dentro da região admissível do problema original pode usar-se o *método Big M* ou o *método das duas fases*. O método das duas fases é mais adequado para implementação em computadores apresentando as seguintes etapas:

1- *Inicialização*. Introdução das variáveis artificiais necessárias.

2- *Fase 1*. Usar o método Simplex para resolver o problema LP:

Objectivo: Minimizar  $Z$ =soma das variáveis artificiais.  
 Sujeito a: Restrições do problema revista.

A solução obtida para este problema (com  $Z=0$ ) será uma solução básica admissível para o problema original.

3- *Fase 2*. Abandonar as variáveis artificiais<sup>19</sup> (não importam visto serem todas zero).

Começando na solução básica admissível obtida na fase 1, usar o método Simplex para resolver o problema original.

Para o exemplo apresentado as funções objectivo seriam:

• *Fase 1*: Minimizar  $Z = \bar{x}_5 + \bar{x}_6$  [A.29]

• *Fase 2*: Maximizar  $Z = 3x_1 + 5x_2$  [A.30]

### SITUAÇÕES A CONSIDERAR

Existem várias situações a ter em consideração durante a execução do algoritmo Simplex e no método das duas fases. São elas:

---

<sup>19</sup> Adiante serão discutidas as situações: 1) variáveis artificiais  $>0$ ; 2) variáveis artificiais degeneradas; e 3) manter as variáveis artificiais como não básicas na fase 2 (e não permitindo que se tornem básicas) para posterior análise pos-ótimo.

- *Solução Degenerada.* Quando nalguma iteração a variável que sai da base possui valor zero, é necessário garantir que a variável que vai entrar para a base fique também com valor zero (visto não poder aumentar o seu valor sem tornar a variável de saída negativa). A estas variáveis é dado o nome de **variáveis degeneradas** e a uma solução que possua variáveis degeneradas é dado o nome de **solução degenerada**.
- *Múltiplas Soluções Óptimas.* Esta situação acontece quando no final do algoritmo Simplex existem variáveis não básicas com coeficiente zero.
- *Restrição Redundante.* Se no final da fase 1 existir alguma restrição com todos os coeficientes igual a zero, excepto o coeficiente da variável básica dessa restrição e excepto os coeficientes das variáveis artificiais, então essa é uma restrição redundante. As restrições redundantes podem ser eliminadas antes de se passar à fase 2 do método das duas fases. Estas não devem ser eliminadas caso se pretenda fazer análise pos-óptimo.
- *Solução Ilimitada.* Uma solução ilimitada significa que o valor da função objectivo pode ser melhorado indefinidamente, isto é, as restrições ao problema não colocam limitações ao valor de  $Z$ . Isto normalmente significa que o problema não foi bem formulado uma vez que no mundo real não são possíveis lucros ( $Z$ ) ilimitados. Esta situação pode ser detectada quando nalguma iteração a variável de entrada possui todos os seus coeficientes a zero ou negativos.
- *Solução Impossível.* O problema não tem solução se no final da primeira fase do método das duas fases existir alguma variável artificial com valor superior a zero ( $Z$  diferente de zero consequentemente).

## BIBLIOGRAFIA

- [1] N. A. Olsson, et. al. – 68.3 km transmission with 1.37 Tbit . km/s capacity using wavelength division multiplexing of ten single-frequency lasers at 1.5  $\mu\text{m}$  (Electron. Lett., vol. 21, 1985)
- [2] C. Lin, H. Kobrinski, A. Frenkel, and C.A. Brackett – Wavelength-tunable 16 optical channel transmission experiment at 2 Gbit/s and 600 Mbit/s for broadband subscriber distribution (Electron. Lett., vol. 24, 1988)
- [3] H. Toba, K. Oda, K. Nakanishi, N. Shibata, K. Nosu, N.Takato, and M. Fukuta – A 100-channel optical FDM transmission/distribution at 622 Mb/s over 50 km (IEEE, Journal of Lightwave Technology, vol. 8, 1990)
- [4] Tien-Pei Lee, et al. – Multiwavelength DFB laser Array Transmitters for ONTC Reconfigurable Optical Network Testbed (IEEE, Journal of Lightwave Technology, vol. 14, n° 6, June 1996)
- [5] A. R. Chraplyvy, et. al. – One-Third Terabit/s Transmission Through 150 km of Dispersion-Managed Fiber (IEEE, Photonics Technology Letters, vol. 7, n° 1, January 1995)
- [6] Matthias Berger, et. al – Pan-European Optical Networking using Wavelength Division Multiplexing (IEEE, Communications Magazine, vol. 35, n° 4, April 1997)
- [7] Jean-Luc Archambault and Stephen G. Grubb – Fiber Gratings in Lasers and Amplifiers (IEEE, Journal of Lightwave Technology, vol. 15, n° 8, August 1997)
- [8] Gerald R. Ash – Dynamic Routing in Telecommunications Networks (IEEE Communications Magazine, July 1995)
- [9] Gerald R. Ash – Dynamic Network Evolution, with Examples from AT&T's Evolving Dynamic Network (IEEE Communications Magazine, July 1995)
- [10] John M. Senior – Optical Fiber Communications (Prentice Hall, 1992)
- [11] John Gowar – Optical Communication Systems (Prentice Hall, 1993)
- [12] Leonid G. Kazovsky, Charles Barry, Michael Hickey, Ciro A. Noronha Jr., and Pierluigi Poggiolini – WDM Local Area Networks (IEEE LTS, May 1992)
- [13] Hiroshi Okamoto, et. al. – A Wavelength-Tunable Duplex Integrated Light Source for Fast Wavelength Switching (IEEE, Journal of Lightwave Technology, vol. 14, n° 6, June 1996)
- [14] Hiroshi Yasaka, et. al. – Repeated Wavelength Conversion of 10 Gb/s Signals and Converted Signal Gating Using Wavelength-Tunable Semiconductor Lasers (IEEE, Journal of Lightwave Technology, vol. 14, n° 6, June 1996)
- [15] Max Ming and Kang Liu – Principles and Applications of Optical Communications (Irwin 1996)

- [16] Ciro A. Noronha Jr. and Fouad A. Tobagi – Routing of Streams in WDM Reconfigurable Networks (Technical Report N° CSL-TR-94-649, December 1994)
- [17] Denis J. G. Mestdagh – Fundamentals of Multiaccess Optical Fiber Networks (Artech House 1995)
- [18] Haim Kobrinski and Kwok-Wai Cheung – Wavelength-Tunable Optical Filters: Applications and Technologies (IEEE Communications Magazine 1989)
- [19] Frank J. Janniello, Rajiv Ramaswami, and David G. Steinberg – A Prototype Circuit-Switched Multi-Wavelength Optical Metropolitan-Area Network (IEEE, Journal of Lightwave Technology, vol. 11, n° 5/6, May/June 1993)
- [20] Eric Hall, Jeff Kravitz, Rajiv Ramaswami, Marty Halvorson, Steve Tenbrink and Richard Thomsen – The Rainbow-II Gigabit Optical Network (IEEE, Journal on Selected Areas in Communications, vol. 14, n° 5, June 1996)
- [21] Richard E. Wagner, Rod C. Alferness, A. A. M. Saleh, and Matthew S. Goodman – MONET: Multiwavelength Optical Networking (IEEE, Journal of Lightwave Technology, vol. 14, n° 6, June 1996)
- [22] Rajiv Ramaswami and Kumar N. Sivarajan – Design of Logical Topologies for Wavelength-Routed Optical Networks (IEEE Journal on Selected Areas in Communications, vol. 14, no. 5, June 1996)
- [23] Jean-François P. Labourdette and Anthony S. Acampora – Logically Rearrangeable Multihop Lightwave Networks (IEEE Transactions on Communications, vol. 39, no. 8, August 1991)
- [24] D. Bienstock, O. Günlük – Computacional Experience with a Difficult Mixed-Integer Multicommodity Flow Problem (Mathematical Programming, vol. 68, 1995)
- [25] Frederick S. Hillier and Gerald J. Liberman – Introduction to Operations Research (McGraw-Hill, 1990)
- [26] Kumar N. Sivarajan – Multihop Logical Topologies For Gigabit Lightwave Networks (IEEE LTS, August 1992)
- [27] F. Ayadi, J. F. Hayes and M. Kavehrad – WDM Cross-Connected Star Topology for the Bilayered ShuffleNet (IEEE, Journal of Lightwave Technology, vol. 12, no.9, September 1994)
- [28] Michael Borella, et. al. – Optical Interconnects for Multiprocessor Architectures Using Wavelength-Division Multiplexing (Department of Computer Science, University of California, Davis, CA 95616)
- [29] Victor S. Frost and Benjamin Melamed – Traffic Modelling For Telecommunications Networks (IEEE Communications Magazine, March 1994)
- [30] Ross Alexander – Monitoring, Analysis and Simulation of Packet Switched Network Traffic (Master of Science Thesis, University of Auckland, January 1995)

- [31] Sheldon M. Ross – Introduction to Probability Models (Academic Press, 1993)
- [32] Will E. Leland and Murad S. Taqqu – On The Self-Similar Nature of Ethernet Traffic (ACM SigComm '93, San Francisco, CA, USA, September 1993)
- [33] A. Mood, F. Graybill and D. Boes – Introduction to the Theory of Statistics (McGraw-Hill, 1985)

