

Este exemplar corresponde à redação final da tese defendida por: e aprovada pela Comissão Julgada em / / *Helio Waldman* Orientador

Universidade Estadual de Campinas - UNICAMP
Faculdade de Engenharia Elétrica e de Computação

Otimização por Colônia de Formigas e sua Aplicação em Redes Ópticas

Autor: Gustavo Sousa Pavani
Orientador: Prof. Dr. Helio Waldman

Tese de Doutorado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Doutor em Engenharia Elétrica. Área de concentração: Telecomunicações e Telemática.

Banca Examinadora

Prof. Dr. Helio Waldman DECOM/FEEC/UNICAMP
Prof. Dr. Paulo Cardieri DECOM/FEEC/UNICAMP
Prof. Dr. Maurício Ferreira Magalhães DCA/FEEC/UNICAMP
Prof. Dr. Fernando José Von Zuben DCA/FEEC/UNICAMP
Prof. Dr. Amílcar Careli César DEE/EESC/USP
Dr. Marcos Rogério Salvador CPqD

18 de Setembro de 2006
Campinas, SP - Brasil

BIBLIOTECA CENTRAL
DESENVOLVIMENTO
COLEÇÃO
UNICAMP

UNIDADE BC
Nº CHAMADA T/UNICAMP
P288o
V _____ EX _____
TOMBO BC/ 70711
PROC. 16-P.00123.06
C 3
PREÇO 11.00
DATA 22/11/06
BIB-ID 391960

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

P288o Pavani, Gustavo Sousa
Otimização por colônia de formigas e sua aplicação em
redes ópticas / Gustavo Sousa Pavani. ---Campinas, SP:
[s.n.], 2006.

Orientador: Helio Waldman.
Tese (doutorado) - Universidade Estadual de Campinas,
Faculdade de Engenharia Elétrica e de Computação.

1. Comunicações óticas. 2. Inteligência artificial
distribuída. 3. Engenharia de tráfego. I. Waldman, Helio.
II. Universidade Estadual de Campinas, Faculdade de
Engenharia Elétrica e de Computação. III. Título.

Título em Inglês: Ant colony optimization and its applications on optical networks.
Palavras-chave em Inglês: Routing; Traffic engineering; Restoration; GMPLS control
plane.

Área de concentração: Telecomunicações e Telemática.

Titulação: Doutor em Engenharia Elétrica.

Banca examinadora: Paulo Cardieri, Maurício Ferreira Magalhães, Fernando José Von
Zuben, Amílcar Careli César e Marcos Rogério Salvador.

Data da defesa: 18/09/2006

The Road Not Taken

TWO roads diverged in a yellow wood,
And sorry I could not travel both
And be one traveler, long I stood
And looked down one as far as I could
To where it bent in the undergrowth;

Then took the other, as just as fair,
And having perhaps the better claim,
Because it was grassy and wanted wear;
Though as for that the passing there
Had worn them really about the same,

And both that morning equally lay
In leaves no step had trodden black.
Oh, I kept the first for another day!
Yet knowing how way leads on to way,
I doubted if I should ever come back.

I shall be telling this with a sigh
Somewhere ages and ages hence:
Two roads diverged in a wood, and I—
I took the one less traveled by,
And that has made all the difference.

Robert Frost. *Mountain Interval*, 1920.

Esta Tese contou com o suporte financeiro da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP).

Resumo

Este trabalho aborda o uso de otimização por colônia de formigas (ACO) em redes ópticas. Como essa técnica exhibe naturalmente capacidade de balanceamento de carga e auto-organização, além de prover controle distribuído na rede, apresentamos um algoritmo de roteamento baseado em formigas artificiais integrado a um plano de controle GMPLS, propondo seu uso nas seguintes aplicações: engenharia de tráfego e restauração em redes comutadas por caminhos ópticos e por pacotes ópticos, roteamento na presença de degenerações na camada física e gerenciamento integrado de caminhos ópticos e de recursos de processamento em arquiteturas de *grid*. O algoritmo proposto é comparado com técnicas tradicionais de roteamento baseado em topologia, como o roteamento por caminho mais curto. Apresentamos as vantagens e as limitações do algoritmo proposto em relação a essas técnicas para diferentes cenários, sendo que os resultados indicam que o algoritmo proposto neste trabalho é um excelente candidato para o controle de redes ópticas em substituição aos métodos tradicionais.

Palavras-chave: Otimização por colônia de formigas, roteamento, engenharia de tráfego, restauração, plano de controle GMPLS, redes ópticas, ruído ASE, *grids*, pacotes ópticos.

Abstract

This work discusses the usage of Ant Colony Optimization (ACO) in optical networking. Since this technique exhibits naturally the capacity of network load-balancing and self-organization, besides it provides a distributed control of the network, we present a routing algorithm based on artificial ants, which is integrated with a GMPLS control plane. We propose the following applications for it: traffic engineering and restoration for wavelength-routed and optical packet switched networks, routing aware of physical-layer impairments, and integrated management of processing resources and lightpaths in grid architectures. The proposed algorithm is compared to conventional routing techniques based on topology, such as shortest-path routing. We present the advantages and limitations of the proposed algorithm when compared to those techniques considering different scenarios. The results indicate that the proposed algorithm is an excellent candidate to manage and control optical networks in lieu of the conventional methods.

Keywords: Ant colony optimization, routing, traffic engineering, restoration, GMPLS control plane, optical networks, ASE noise, grids, optical packets.

Sumário

Resumo & Abstract	i
Lista de Figuras	vii
Lista de Tabelas	xi
1 Introdução	1
1.1 Escopo	4
2 Otimização por Colônia de Formigas	5
2.1 Linhas Gerais do Algoritmo AntNet	7
2.2 Estruturas de Dados	8
2.3 Descrição do Algoritmo Proposto	10
2.3.1 Construção das Soluções	11
2.3.2 Atualização das Estruturas de Dados	12
2.4 Exemplo de uma Iteração do Algoritmo	15
2.5 Restauração	18
2.5.1 No caso de Roteamento por Caminho Mais Curto	20
2.5.2 No caso do Algoritmo Proposto	21
2.6 Redes de Testes	22
3 Roteamento e Alocação de Comprimento de Onda (RWA)	25
3.1 Demonstração intuitiva do funcionamento da ACO no problema do RWA	26
3.2 Trabalhos prévios na literatura	29
3.3 Variações do Algoritmo Proposto	30
3.3.1 Restauração	30
3.4 Simulação	31
3.5 Resultados	32
3.5.1 Restauração	35
3.5.2 Considerações finais	39
4 Degenerações na Camada Física	41
4.1 RWA com Restrições de Potência (RWA-P)	43
4.1.1 Ruído ASE	44

4.1.2	Nível de sensibilidade	44
4.1.3	Cascata de Amplificadores: O Pré-Amplificador Equivalente	45
4.2	Arquitetura do Nó Óptico	46
4.2.1	Ganho do amplificador	47
4.3	Variações do Algoritmo Proposto	48
4.4	Simulação	49
4.5	Resultados	52
5	Gerenciamento de Recursos em Grids	55
5.1	Variações do Algoritmo Proposto	57
5.2	Arquitetura	58
5.3	Simulação	59
5.4	Resultados	60
6	Redes de Pacotes Ópticos	69
6.1	Resolução de Contenção em Nós OPS	70
6.2	Variações do Algoritmo Proposto	73
6.3	Simulação	75
6.3.1	Tráfego Poissoniano	75
6.3.2	Tráfego Auto-similar	77
6.4	Resultados	78
6.4.1	Tráfego Poissoniano	78
6.4.1.1	Restauração	82
6.4.2	Tráfego Auto-Similar	84
6.4.3	Considerações finais	88
7	Conclusões	91
A	Simulação de Tráfego	95
A.1	Tráfego Poissoniano com Duração Exponencialmente Distribuída	95
A.2	Tráfego Auto-Similar	96
B	O Padrão GMPLS	99
B.1	Cálculo de Rota e Alocação de Comprimento de Onda	101
B.1.1	Roteamento fixo	102
B.1.2	Roteamento fixo-alternado	102
B.1.3	Roteamento baseado em ACO	103
B.2	Sinalização	103
B.2.1	Recuperação de falhas	106
C	Lista de Trabalhos Publicados	107
C.1	Trabalhos Ligados Diretamente a Tese	107
C.2	Demais Trabalhos	108

SUMÁRIO	v
D Lista de Acrônimos	109
E Índice Remissivo de Autores	111
Referências Bibliográficas	113

Lista de Figuras

2.1	Exemplo da tabela de roteamento por ferormônio do nó 4.	9
2.2	Exemplo do modelo paramétrico estatístico do nó 4.	10
2.3	Função de amassamento com $a = 5$	14
2.4	Exemplo de uma viagem completa de um nó origem ao nó destino.	15
2.5	Estado atual das estruturas de dados dos nós 1 e 2.	15
2.6	Exemplo de seleção probabilística do próximo hop.	16
2.7	Rede NSFNet.	22
2.8	Rede NTTNet.	23
3.1	Uma rota descoberta por formigas (c) e duas rotas ineficientes (a e b).	28
3.2	Roteamento fixo-alternado versus roteamento baseado em ACO para a rede NSFNet.	32
3.3	Roteamento fixo-alternado versus roteamento baseado em ACO para a rede NTT-Net.	33
3.4	Hot-spots para a rede NTTNet com carga = 40 erlangs	34
3.5	Percentagem das conexões estabelecidas por comprimento da rota na rede NTT-Net com uma carga = 20 erlangs.	34
3.6	Influência dos parâmetros na capacidade de restauração da rede NSFNet em falha simples de nó para o algoritmo baseado em ACO.	35
3.7	Influência dos parâmetros na capacidade de restauração da rede NSFNet em falha simples de enlace para o algoritmo baseado em ACO.	36
3.8	Capacidade de restauração no caso de falha simples de nó para a rede NSFNet.	37
3.9	Capacidade de restauração no caso de falha simples de enlace para a rede NSFNet.	37
3.10	Capacidade de restauração no caso de falha simples de nó para a rede NTTNet.	38
3.11	Capacidade de restauração no caso de falha simples de enlace para a rede NTTNet.	38
4.1	Cascata de amplificadores.	45
4.2	Arquitetura do WRN.	47
4.3	Curvas de saturação para um amplificador óptico de $G_{max} = 20$ dB e $P^{sat} = 13,7$ mW.	48
4.4	Rede de teste para os algoritmos de RWA-P.	49
4.5	Roteamento fixo e fixo-alternado versus roteamento baseado em ACO.	52
4.6	Percentagem das conexões estabelecidas por comprimento da rota com uma carga igual a 10 erlangs.	53

5.1	Um sistema de grid gerenciado em duas camadas.	59
5.2	Probabilidade de bloqueio no grid para diferentes cargas.	61
5.3	Probabilidade de bloqueio total e relativa somente à fazenda de servidores para $W = 2$ comprimentos de onda.	61
5.4	Probabilidade de bloqueio total e relativa somente à fazenda de servidores para $W = 4$ comprimentos de onda.	62
5.5	Probabilidade de bloqueio total e relativa somente à fazenda de servidores para $W = 8$ comprimentos de onda.	62
5.6	Carga de trabalho total do grid para uma carga de 4 erlangs na rede óptica.	63
5.7	Carga de trabalho total do grid para $W = 8$ comprimentos de onda.	63
5.8	Probabilidade de bloqueio em função do número de processadores com um carga de 4 erlangs na rede óptica.	64
5.9	Probabilidade de bloqueio total e relativa somente à fazenda de servidores em função do número de processadores para $W = 2$ e carga na rede óptica igual a 4 erlangs.	65
5.10	Probabilidade de bloqueio total e relativa somente à fazenda de servidores em função do número de processadores para $W = 4$ e carga na rede óptica igual a 4 erlangs.	65
5.11	Probabilidade de bloqueio total e relativa somente à fazenda de servidores em função do número de processadores para $W = 8$ e carga na rede óptica igual a 4 erlangs.	66
5.12	Probabilidade de bloqueio em função da taxa global de lançamento de formigas para uma carga na rede óptica de 4 erlangs.	67
5.13	Overhead das formigas artificiais $W = 8$ comprimentos de onda e carga na rede óptica de 4 erlangs.	68
6.1	Exemplo de um nó OPS com $W = 4$ e $B = 4$	70
6.2	Algoritmo de WDS.	71
6.3	Probabilidade de perda versus unidade de atraso para carga = 6.4 erlangs.	72
6.4	Influência da ocupação média do buffer sobre a probabilidade de perda para tráfego poissoniano.	72
6.5	Influência da ocupação média do buffer sobre a probabilidade de perda para tráfego auto-similar.	73
6.6	A transformação da função de decaimento exponencial.	74
6.7	Roteamento por caminho mais curto versus roteamento baseado em ACO para a rede NSFNet com tráfego poissoniano.	78
6.8	Roteamento por caminho mais curto versus roteamento baseado em ACO para a rede NTTNet com tráfego poissoniano.	79
6.9	Utilização média de enlace na rede NSFNet com carga igual a 80 erlangs para tráfego poissoniano.	80
6.10	Utilização média de enlace na rede NTTNet com carga igual 40 erlangs para tráfego poissoniano.	81
6.11	Hot-spots com carga = 80 erlangs para tráfego poissoniano.	82

6.12	Percentagem de chegadas de pacote por comprimento da rota na rede NTTNet com carga = 80 erlangs para tráfego poissoniano.	82
6.13	Perda acumulada para restauração por caminho mais curto (SP) versus ACO.	83
6.14	Perda instantânea na rede NSFNet para restauração por caminho mais curto versus restauração baseada em ACO.	83
6.15	Perda instantânea na rede NTTNet para restauração por caminho mais curto versus restauração baseada em ACO.	84
6.16	Probabilidade de perda em um buffer com carga = 6.4 erlangs e tráfego poissoniano.	86
6.17	Probabilidade de perda em um buffer com carga = 6.4 erlangs e tráfego auto-similar.	86
6.18	Roteamento por caminho mais curto versus roteamento baseado em ACO para a rede NSFNet com tráfego auto-similar.	87
6.19	Roteamento por caminho mais curto versus roteamento baseado em ACO para a rede NTTNet com tráfego auto-similar.	87
6.20	Utilização média de enlace na rede NSFNet com carga = 80 erlangs para tráfego auto-similar.	88
6.21	Utilização média de enlace na rede NTTNet com carga = 40 erlangs para tráfego auto-similar.	89
6.22	Hot-spots com carga = 80 erlangs para tráfego auto-similar.	90
6.23	Percentagem de chegadas de pacote por comprimento da rota na rede NTTNet com carga = 80 erlangs para tráfego auto-similar.	90
B.1	Base de dados estado-enlace de cada nó óptico da rede exemplificada.	101
B.2	Cálculo da rota de um LSP do nó A ao nó C.	103
B.3	Exemplo de falha de estabelecimento de um LSP devido à violação da restrição de continuidade de comprimento de onda.	104
B.4	Exemplo de estabelecimento de um LSP. O nó de egresso seleciona o primeiro rótulo disponível para alocação do comprimento de onda.	105
B.5	Exemplo de sinalização de um LSP afetado por falha.	106

Lista de Tabelas

2.1	Parâmetros usados na simulação do exemplo.	16
3.1	Parâmetros usados na simulação dos algoritmos de RWA.	31
4.1	Parâmetros e seus respectivos valores usados no RWA-P.	51
4.2	Parâmetros relativos aos algoritmos baseados em ACO para o problema RWA-P.	51
5.1	Parâmetros usados na simulação do gerenciamento de recursos no grid.	60
6.1	Parâmetros usados na simulação de redes OPS com tráfego poissoniano.	76
6.2	Parâmetros usados na simulação de redes OPS com tráfego auto-similar.	77
6.3	Valores de média e variância para $k = 64$ e diferentes valores de p na distribuição BP.	85

Capítulo 1

Introdução

As taxas de bits nos sistemas de transmissão ópticos estão aumentando 1000 vezes por década desde os primeiros sistemas fabricados na década de 70. Em comparação, o poder de processamento segue a lei de Moore, que postula um crescimento de 100 vezes por década. Como resultado, o gargalo das redes de comunicação passou dos enlaces para os nós, de forma que novas formas de gerenciamento da rede são necessárias para enfrentar esse gargalo.

Em redes ópticas, essa limitação se manifesta como bloqueio de conexões em redes comutadas por caminhos ópticos – Wavelength Routed Optical Network (WRON) ou em perda de pacotes em redes comutadas por pacotes ópticos – Optical Packet Switching (OPS).

A redução de perdas de pacotes ou do bloqueio de caminhos ópticos pode ser alcançada com o uso de tecnologias de Engenharia de Tráfego – Traffic Engineering (TE) [1]. As técnicas de TE tem como objetivo otimizar o desempenho de redes de comunicação. Essas técnicas podem ser orientadas ao tráfego ou aos recursos. No primeiro caso, a meta é melhorar a qualidade de serviço – Quality of Service (QoS) das conexões ou dos fluxos de pacotes. No segundo caso, foca-se no uso eficiente de recursos.

A minimização do congestionamento é o principal objetivo da engenharia de tráfego tanto orientada a tráfego quanto a recurso. Concerne a redução do congestionamento quando o

tráfego é mapeado em enlaces sobrecarregados enquanto outros enlaces possíveis permanecem sub-utilizados. Em geral, uma estratégia baseada em balanceamento de carga reduz o congestionamento causado por uma alocação ineficiente de recursos, o que é comumente o caso dos algoritmos baseados em topologia, como o roteamento por caminho mais curto – Shortest Path (SP), que não consideram as características do tráfego ou a banda disponível nos enlaces.

Outro aspecto importante em redes ópticas é o gerenciamento de falhas. Uma única falha de enlace pode prejudicar milhares ou até mesmo milhões de usuários. Estratégias para se achar rotas alternativas são necessárias em caso de falhas de enlace ou nó.

As falhas de nó são bem mais raras que as falhas de enlace, devido ao uso de redundância nos nós. Entretanto, esse tipo de falha é bem mais difícil de ser gerenciada, pois causa falhas simultâneas em todos os enlaces adjacentes, de forma que a rede não tem tempo para recalcular um plano de proteção como quando trata de múltiplas falhas seqüenciais. Assim, como 100% de proteção contra múltiplas falhas é extremamente ineficiente do ponto de visto de uso de recursos, restauração deve ser usada.

De fato, quando múltiplas falhas acontecem, elas podem afetar as conexões de trabalho e de reserva, mesmo que as rotas dessas conexões pertençam a diferentes grupos de risco compartilhado de enlace – Shared Risk Link Group (SRLG). Assim, um esquema de proteção pode não recuperar uma conexão na presença de múltiplas falhas, mas um esquema de restauração tem mais chance de achar um caminho alternativo para as conexões afetadas.

Neste contexto, propomos um sistema baseado em otimização por colônia de formigas – Ant Colony Optimization (ACO) como uma interessante alternativa para engenharia de tráfego e restauração.

O termo ACO [2] é usado para se referir à classe de algoritmos que foram inspirados no processo de procura e coleta de alimentos por formigas naturais para a otimização de problemas de difícil solução ou que necessitam de controle distribuído. Esses algoritmos exibem inteligência de coletividade ou de enxame – Swarm Intelligence (SI) [3], que é uma propriedade de sistemas

compostos por agentes individualmente não inteligentes e com capacidade limitada exibirem coletivamente um comportamento inteligente.

Nas formigas, SI surge através de um tipo de comunicação indireta entre elas, que é chamada de estigmergia [4]. Estigmergia é um termo derivado das palavras gregas *stigma* (sinal) e *ergon* (ação), que foi originalmente definido como a “estimulação de trabalhadores pelo desempenho alcançado”. De fato, as formigas depositam um hormônio volátil, chamado ferormônio, que age como um estimulante para outras formigas. Quando uma formiga procura por comida, ela deixa pequenas quantidades de ferormônio para marcar a trilha, possibilitando que outras formigas consigam seguir esse mesmo caminho. Assim, algo (ferormônio) que não é relacionado diretamente a uma tarefa é depositado no ambiente, mas é usado para influenciar o comportamento posterior relacionado a essa tarefa.

Dessa forma, neste trabalho, formigas são representadas por agentes¹ móveis² [5, 6] que, ao modificar os níveis de ferormônio artificiais em tabelas dos nós da rede, são capazes de, por exemplo, otimizar de forma global e distribuída o uso dessa mesma rede.

Uma vez que ambas engenharia de tráfego e restauração necessitam de cálculo de rota na rede e que essas funções podem ser feitas através de níveis de ferormônio no roteamento baseado em formigas, elas são consideradas em conjunto neste trabalho.

Esta Tese é organizada da seguinte maneira:

Capítulo 1 é esta introdução, o qual fornece um panorama deste trabalho.

Capítulo 2 apresenta os algoritmos baseados em ACO que serão usados ao longo deste trabalho.

Capítulo 3 trata do gerenciamento de WRON. Através de um plano de controle Generalized Multi-Protocol Label Switching (GMPLS) integrado, são apresentados algoritmos

¹Agente (de computador) pode ser definido como uma entidade computacional que age autonomamente e é capaz de reagir a eventos externos, de tomar decisões de acordo com o seu ambiente e de cooperar com outros agentes.

²Agentes móveis são agentes capazes de migrar de forma autônoma pela rede.

de roteamento com capacidade de sobrevivência a falhas.

Capítulo 4 propõe um algoritmo capaz de estabelecer caminhos ópticos com restrições de camada física, baseando-se nas técnicas apresentadas no Capítulo 3.

Capítulo 5 detalha uma arquitetura de gerenciamento de recursos em grids, onde a rede óptica é tratada de forma similar aos recursos de processamento e armazenamento. Também faz uso da proposta introduzida no Capítulo 3.

Capítulo 6 introduz uma plataforma integrada para gerenciamento de redes OPS, a qual prevê o roteamento adaptativo com tolerância a falhas.

Capítulo 7 apresenta as conclusões obtidas neste trabalho.

1.1 Escopo

Este trabalho é limitado a uma área controlada por uma instância de plano de controle GMPLS, de forma a constituir um único sistema autônomo – Autonomous System (AS), ou seja, o roteamento inter-domínio não foi considerado.

O escopo deste trabalho também não cobre os aspectos ligados à segurança do sistema de roteamento por formigas. De fato, um ataque por um agente mal-intencionado que consiga inserir formigas falsas pode comprometer completamente o roteamento do sistema.

Finalmente, não trata da disponibilidade de redes ópticas, pois além de requerer uma modelagem analítica específica, a disponibilidade depende mais da topologia da rede em si e do tipo de demanda de tráfego do que a disponibilidade de recursos [7].

Capítulo 2

Otimização por Colônia de Formigas

A ACO foi desenvolvida inicialmente para tratar de problemas em que se deseja descobrir o caminho de custo mínimo em um grafo $G = (N, A)$, onde N é o conjunto de nós e A é o conjunto de arestas desse grafo. Os dois pontos entre os quais se quer estabelecer um caminho de custo mínimo são chamados de origem e destino. Quando o custo das arestas é dado por seu comprimento, o problema de caminho de custo mínimo é o mesmo que o problema de caminho mais curto¹.

Baseando-se nas características das formigas reais, as formigas artificiais devem ter uma forma limitada de memória, de tal maneira que sejam capazes de guardar as rotas percorridas por elas. Assim, através de sua memória, as formigas implementam um certo número de comportamentos que permitem que elas construam soluções de forma eficiente para o problema de caminho de custo mínimo. Esses comportamentos são:

1. Construção da solução de forma probabilística, através dos caminhos de ferormônio, pela formiga no caminho de ida;
2. Retorno da formiga pelo mesmo caminho de ida com a atualização dos níveis de fe-

¹Tradicionalmente, em redes ópticas, os termos problema de caminho de custo mínimo e problema de caminho mais curto têm o mesmo significado, sendo mais comum o emprego do segundo termo.

ferormônio;

3. Avaliação da qualidade das soluções geradas e o uso da qualidade da solução para se determinar a quantidade de ferormônio a ser depositada. Esta avaliação é feita de forma concomitante ao retorno da formiga.

Assim, através de um processo iterativo, cada formiga constrói uma solução usando dois tipos de informações locais: informação específica do problema e informação adicionada pelas formigas durante iterações anteriores do algoritmo (ferormônio). De fato, enquanto produz a solução, cada formiga coleta informações sobre as características do problema a ser resolvido e sobre o seu próprio desempenho e utiliza essa informação para modificar a representação do problema, conforme visto localmente por outras formigas. Dessa maneira, a representação do problema é modificada de tal forma que a informação contida nas soluções obtidas anteriormente possa ser explorada para a obtenção de soluções melhoradas.

Em redes de comunicação, o problema é achar os caminhos de custo mínimo entre todos os nós da rede. A cada aresta de A é associado um custo d_{ij} , onde i é o nó origem da aresta e j é o nó destino da aresta. Se os custos d_{ij} são fixos, esse problema é reduzido a um conjunto de problemas de caminho de custo mínimo, cada qual podendo ser resolvido eficientemente através de algoritmos de tempo polinomial, como o algoritmo de Dijkstra [8]. Entretanto, em redes de telecomunicação, a topologia da rede e esses custos variam no tempo, devido a mudanças do tráfego, congestionamento ou falhas de componentes da rede.

O uso de ACO em redes de comunicação permite um roteamento adaptativo ao tráfego ou ao congestionamento na rede. Entre as técnicas de ACO usadas nesse caso, é possível citar AntNet [9, 10], Ants-Routing [11] e Accelerated Ants Routing [12].

AntNet foi usado no roteamento de redes IP convencionais, isto é, do tipo *store-and-forward*, provando ser superior a seis algoritmos de roteamento no estado da arte em redes IP. Por causa dessa superioridade, AntNet serviu de base para este trabalho, recebendo as modificações necessárias para operar junto aos problemas apresentados nesta Tese.

2.1 Linhas Gerais do Algoritmo AntNet

De forma geral, o algoritmo AntNet e suas características principais podem ser resumidas da seguinte maneira:

- Em intervalos regulares e de forma concorrente ao tráfego de dados, de cada nó da rede, formigas artificiais são lançadas de forma assíncrona em direção aos nós destino escolhidos de forma aleatória, levando em conta a distribuição do tráfego.
- As formigas artificiais agem de forma concorrente e independente, se comunicando através dos ferormônios que elas lêem ou escrevem localmente nos nós, ou seja, a comunicação se faz através da estigmergia.
- Cada formiga artificial procura por um caminho de mínimo custo entre seus nós origem e destino.
- A cada nó intermediário visitado pela formiga artificial, uma estratégia estocástica é usada para decidir qual o próximo nó a ser visitado. Essa estratégia faz uso dos níveis de ferormônio locais, informação heurística local (específica do problema) e da memória da formiga.
- Enquanto se movem, as formigas artificiais coletam uma identificação dos nós visitados.
- Quando as formigas chegam no nó de destino, as formigas artificiais retornam ao nó de origem seguindo o mesmo caminho da ida, mas na direção oposta.
- Durante o retorno da formiga ao nó de origem, os modelos paramétricos e os níveis de ferormônio são modificados pelas formigas artificiais como função do caminho percorrido e de sua avaliação em cada nó intermediário.
- Quando chegam ao nó de origem, as formigas artificiais são descartadas pelo sistema.

2.2 Estruturas de Dados

As estruturas de dados do AntNet foram ligeiramente modificadas para este trabalho, pois o algoritmo AntNet original previa que a informação a ser incorporada pelas formigas era o atraso sofrido pela formiga a cada *hop*. Esse tipo de informação não se aplica em redes do tipo WRON ou OPS. A métrica mais comumente usada é a distância, em número de *hops*, entre um par origem-destino. No segundo caso, há um problema extra, pois o pacote deve ser opticamente processado e uma seqüência de atrasos seria muito difícil de ser adicionada ou extraída a cada nó visitado.

Dessa maneira, no algoritmo proposto para este trabalho, a formiga carregará somente um rótulo do nó visitado. De fato, a minimização do número de *hops* de uma conexão através de algum critério específico demonstrou ser uma heurística excelente para se reduzir o bloqueio total em redes do tipo WRON ou OPS, além de facilitar o processamento completamente óptico da formiga [13] em redes do tipo OPS, de forma que consideramos, sem perda de generalidade, que o tempo desse processamento é muito pequeno e pode ser desprezado nos cálculos.

Dessa forma, a cada nó intermediário i , as seguintes estruturas de dados são mantidas:

1. Tabela de roteamento por ferormônio \mathcal{T}^i : É uma tabela contendo uma linha para cada destino d da rede e uma coluna para cada nó vizinho com o propósito de guardar os valores de ferormônio. O elemento τ_{nd}^i indica a probabilidade de se selecionar um enlace para se alcançar o destino d , de acordo com o que se aprendeu através das visitas das formigas. A soma de cada linha deve ser igual a 1, isto é, $\sum_{n \in \mathcal{N}_i} \tau_{nd}^i = 1$, onde \mathcal{N}_i representa o conjunto de vizinhos do nó i . A Figura 2.1 ilustra um exemplo de tabela de roteamento por ferormônio.
2. Modelo paramétrico estatístico \mathcal{M}^i : É uma tabela contendo a tupla $\langle \mu_d^i, \sigma_d^i, E_d^i \rangle$ para cada destino d da rede, onde μ_d^i representa o comprimento médio dos caminhos seguidos pelas formigas do nó atual i até o destino d , σ_d^i é o desvio padrão desses

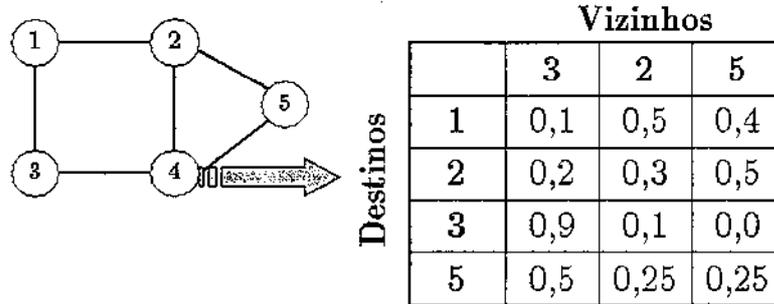


Figura 2.1: Exemplo da tabela de roteamento por ferormônio do nó 4.

caminhos e E_d^i é o melhor valor de comprimento de caminho descoberto para o destino d , dentro da janela não-deslizante de w observações, conforme descrito no algoritmo AntNet original. Não-deslizante significa que, quando o número de observações alcança $w+1$, todos os valores acumulados são re-inicializados, isto é, $E_d^i = \mu_d^i$ (comprimento do caminho da $(w+1)$ -ésima observação), $\sigma_d^i = 0$ e o contador de observações recebe 1. A Figura 2.2 ilustra um exemplo de modelo paramétrico estatístico.

Para cada destino d , μ_d^i e σ_d^i dão uma representação do tamanho do caminho do nó i ao nó d e de sua estabilidade. Essas estatísticas são atualizadas usando um modelo exponencial [14]:

$$\mu_d^i \leftarrow \mu_d^i + \eta(o_{i \rightarrow d} - \mu_d^i) \quad (2.1)$$

$$\sigma_d^i \leftarrow \sigma_d^i + \eta(|o_{i \rightarrow d} - \mu_d^i| - \sigma_d^i), \quad (2.2)$$

onde $o_{i \rightarrow d}$ é a nova observação, que é a distância entre os nós i e d , e η é o peso do modelo exponencial, que pondera o número das mais recentes amostras que irão realmente influenciar a média. O número de observações efetivas é aproximadamente $5/\eta$ [9].

O número de valores na janela é calculado como $w = 5(c/\eta)$, onde $c \in (0, 1]$

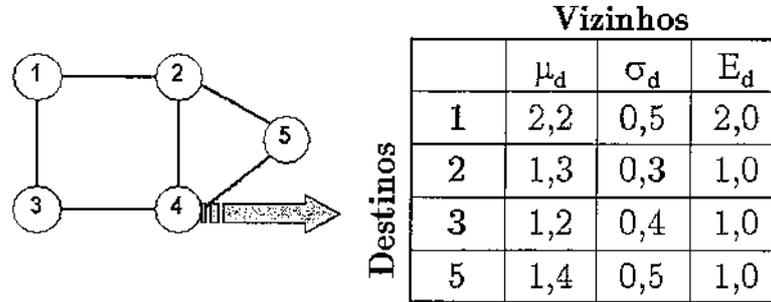


Figura 2.2: Exemplo do modelo paramétrico estatístico do nó 4.

é um fator de redução. Assim, a janela é atualizada em um intervalo menor que o usado para os estimadores de média e desvio padrão, de forma que o valor de E_d^i e dos estimadores refiram-se ao mesmo conjunto de observações.

\mathcal{M} e \mathcal{T} podem ser vistos como memórias locais para os nós, capturando aspectos diferentes da dinâmica da rede. O modelo \mathcal{M} mantém estimadores absolutos de distância para todos os nós, enquanto a matriz de roteamento por ferormônios dá medidas de utilidade relativas ao se escolher o próximo *hop* para se alcançar um dado destino.

2.3 Descrição do Algoritmo Proposto

O algoritmo proposto generaliza algumas partes do algoritmo original AntNet, sendo que as variantes necessárias para acomodar os diferentes usos do roteamento por formigas são explicados separadamente no capítulo apropriado.

O algoritmo proposto é composto por duas fases: construção das soluções e atualização das estruturas de dados, que serão descritas nas próximas sub-seções.

2.3.1 Construção das Soluções

Em intervalos regulares ($1/R_{ants}$), uma formiga $F_{s \rightarrow d}$ é enviada de um nó aleatório s da rede a um outro nó aleatório d da rede. Durante sua viagem de s para d , a formiga seleciona o próximo *hop* usando um esquema aleatório que leva em consideração as probabilidades de se selecionar o próximo nó vizinho, que é dado pelos níveis de ferormônio em cada enlace, e uma métrica heurística local h_{nd} , que é dependente do problema.

A formiga adiciona a identificação ou rótulo de cada nó visitado, colocando-o em sua memória $V_{s \rightarrow i}$, a qual serve como memória tabu [15].

Se entre os vizinhos do nó onde a formiga está sendo processada houver nós que ainda não foram visitados, a escolha do próximo nó é feita de forma aleatória. Assim, as probabilidades de cada nó candidato n a ser o próximo *hop* é dada pela expressão [9]:

$$p_{nd} = \begin{cases} \frac{\tau_{nd}}{(1 + \alpha)\Sigma_\tau} + \frac{\alpha h_{nd}}{(1 + \alpha)\Sigma_h} & , \forall n \in \mathcal{N}_i \wedge n \notin V_{s \rightarrow i} \\ 0 & , \text{ caso contrário,} \end{cases} \quad (2.3)$$

onde $\Sigma_\tau = \sum_{n \notin V_{s \rightarrow i}} \tau_{nd}$, $\Sigma_h = \sum_{n \notin V_{s \rightarrow i}} h_{nd}$ e α dá a ênfase entre os níveis de ferormônio (memória de longo prazo) e a métrica fornecida pela heurística (memória de curto prazo).

Caso todos os nós vizinhos já tenham sido visitados, isso indica que a formiga entrou em um laço. O próximo nó a ser visitado é também escolhido de forma aleatória, onde as probabilidades de cada nó candidato é dado pela seguinte expressão:

$$p_{nd} = \begin{cases} \frac{\tau_{nd}}{\Sigma_\tau} & , \forall n \in \mathcal{N}_i \wedge (n \neq v_{i-1} \vee |\mathcal{N}_i| = 1) \\ 0 & , \text{ caso contrário,} \end{cases} \quad (2.4)$$

onde v_{i-1} representa o último nó visitado.

Neste caso, após a seleção do nó n , o laço é removido da memória da formiga.

Antes que o tráfego de dados comece a chegar na rede, é necessário um intervalo I_{warmup} ,

onde somente formigas são permitidas na rede, de forma que seja possível a exploração da rede e a inicialização das tabelas de roteamento. Na prática, isso permite a configuração das tabelas de roteamento por ferormônio com os caminhos de custo mínimo quando ainda não há tráfego a ser roteado.

Se a formiga artificial não consegue alcançar o nó destino em um número pré-estabelecido de *hops*, ela é descartada.

2.3.2 Atualização das Estruturas de Dados

Quando a formiga chega em seu nó destino d , a formiga artificial $F_{s \rightarrow d}$ transforma-se na formiga $B_{d \rightarrow s}$ e assume o papel de atualizar as estruturas de dados dos nós visitados, cujas identificações estão guardadas na memória $V_{s \rightarrow d}$. Assim, fazendo o caminho reverso ao da ida, de cada nó intermediário i até o nó origem s , a formiga $B_{d \rightarrow s}$ atualiza o modelo paramétrico local de tráfego \mathcal{M}^i e a matriz de roteamento por ferormônio T^i , para todas as entradas correspondentes a d .

Além disso, também são efetivadas as atualizações correspondentes a todos os nós $d' \in V_{i \rightarrow d}$, $d' \neq d$ nos segmentos de caminho ($i \rightarrow d'$) percorridos pela formiga $F_{s \rightarrow d}$ após ter visitado o nó i . Se o tamanho desse segmento de caminho é estatisticamente bom, então as entradas de \mathcal{M}^i e T^i correspondentes a d' também são atualizadas.

O que define se um segmento de caminho é estatisticamente bom é se $dist(V_{i \rightarrow d'}) < I_{sup}^{d'}$ [9], onde $dist()$ é uma função que dá a distância, em número de *hops*, do caminho seguido pela formiga e I_{sup} é o estimador superior obtido através das relações de Tchebycheff, que permitem a definição de um intervalo de confiança de uma variável aleatória que segue qualquer tipo de distribuição. O estimador inferior é igual a $E_{d'}$. O estimador $I_{sup}^{d'}$ é expresso pela seguinte fórmula:

$$I_{sup}^{d'} = \mu_{d'} + \frac{1}{\sqrt{(1-\gamma)}} \frac{\sigma_{d'}}{\sqrt{w}}, \quad (2.5)$$

onde γ é o coeficiente do nível de confiança.

Dessa forma, o modelo paramétrico local é atualizado usando-se as Equações 2.1 e 2.2, onde $o_{i \rightarrow d} = \text{dist}(V_{i \rightarrow d})$. Se $o_{i \rightarrow d} < E_d^i$, então $E_d^i = o_{i \rightarrow d}$. O mesmo processo é repetido para os d' cujos segmentos de caminho foram considerados estatisticamente bons.

Após a atualização de \mathcal{M}^i , o reforço adaptativo r_d é calculado para se atualizar a tabela de roteamento por ferormônio \mathcal{T}^i [9]:

$$r_d = c_1 \left(\frac{E_d}{\text{dist}(V_{i \rightarrow d})} \right) + c_2 \left(\frac{I_{sup}^d - E_d}{(I_{sup}^d - E_d) + (\text{dist}(V_{i \rightarrow d}) - E_d)} \right) \quad (2.6)$$

O primeiro termo da Equação 2.6 simplesmente avalia a razão entre a distância da melhor rota dentro da janela de observações e a distância atual percorrida pela formiga. O segundo termo avalia o quão longe essa distância atual está do intervalo de confiança. É importante ressaltar que o segundo termo deve ser considerado igual a zero quando $\text{dist}(V_{i \rightarrow d}) = I_{sup}^d = E_d$. Os coeficientes c_1 e c_2 são parâmetros que medem a importância de cada termo.

O valor obtido de r_d é limitado a 0,9 para evitar estagnação e o seu valor é “amassado” usando a seguinte expressão[9]:

$$r_d = \frac{s(r_d)}{s(1)} \text{ e } s(x) = \left(1 + \exp \left(\frac{a}{x|\mathcal{N}_i|} \right) \right)^{-1}, \quad (2.7)$$

onde a é o coeficiente amplificador da função.

A função de amassamento permite que o sistema recompense bons valores de r , enquanto satura os valores ruins de r . Como pode ser observado na Figura 2.3, a escala é comprimida para os valores mais baixos de reforço e expandida para os valores mais altos. Dessa forma, a ênfase é dada para os melhores resultados de reforço.

Caso o nó vizinho m esteja no caminho da formiga, ele recebe um reforço positivo:

$$\tau_{md} \leftarrow \tau_{md} + r_d(1 - \tau_{md}) \quad (2.8)$$

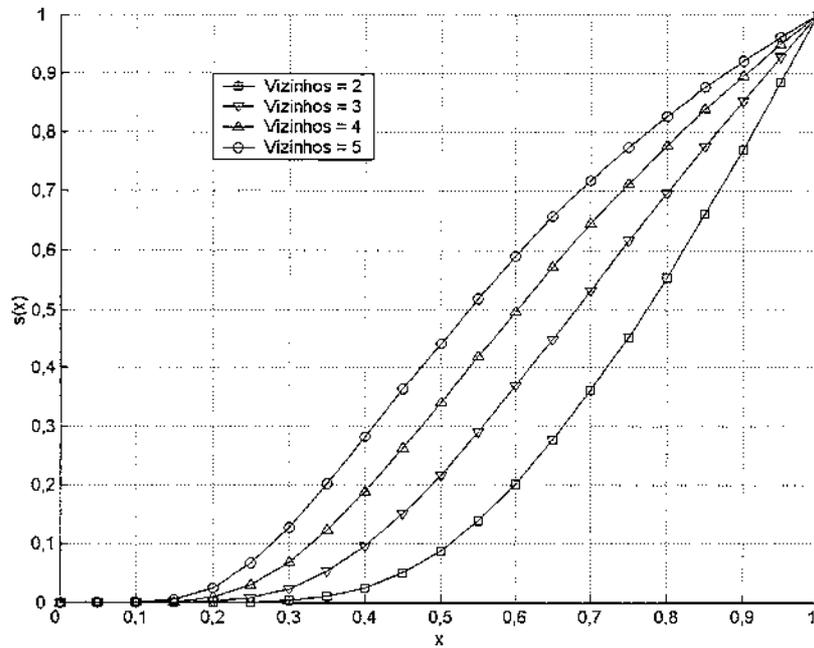


Figura 2.3: Função de amassamento com $a = 5$.

Por outro lado, os outros nós vizinhos recebem um reforço negativo:

$$\tau_{nd} \leftarrow \tau_{nd} - r_d \tau_{nd}, \forall n \in \mathcal{N}_i, n \neq m, \quad (2.9)$$

Assim como para o modelo paramétrico \mathcal{M}^i , o processo de atualização de T^i é repetido para os d' considerados estatisticamente bons.

Como o algoritmo implementa uma estratégia de deposição privilegiada de ferormônio, as chances de uma formiga reforçar um caminho que seja não-ótimo ou que esteja congestionado segundo alguma métrica é baixo. Dessa maneira, é possível prescindir de estratégias que usam ruído ou evaporação para evitar a estagnação [16], sem comprometer a eficiência do algoritmo.

Finalmente, embora \mathcal{M}^i e T^i não possuam informações sobre o congestionamento da rede, o fato de haver mais formigas que escolhem o caminho menos congestionado resulta em uma realimentação do sistema que acaba reforçando os níveis de ferormônio desses caminhos.

2.4 Exemplo de uma Iteração do Algoritmo

A formiga $F_{1 \rightarrow 5}$ é enviada do nó 1 com destino ao nó 5, como pode ser visto na Figura 2.4.

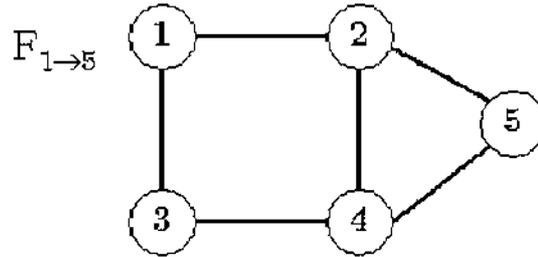


Figura 2.4: Exemplo de uma viagem completa de um nó origem ao nó destino.

A Figura 2.5 mostra o estado atual das tabelas de roteamento por ferormônio e dos modelos paramétricos estatísticos dos nós 1 e 2.

	2	3
2	0,9	0,1
3	0,1	0,9
4	0,5	0,5
5	0,7	0,3

	μ_d	σ_d	E_d
2	1,2	0,2	1,0
3	1,1	0,1	1,0
4	2,3	0,3	2,0
5	2,4	0,5	2,0

(a) Nó 1

	1	4	5
1	0,9	0,1	0
3	0,5	0,5	0
4	0,1	0,7	0,2
5	0,01	0,1	0,89

	μ_d	σ_d	E_d
1	1,2	0,2	1,0
3	2,3	0,4	2,0
4	1,2	0,2	1,0
5	1,1	0,1	1,0

(b) Nó 2

Figura 2.5: Estado atual das estruturas de dados dos nós 1 e 2.

A Tabela 2.1 contém os parâmetros usados na simulação deste exemplo.

Tabela 2.1: Parâmetros usados na simulação do exemplo.

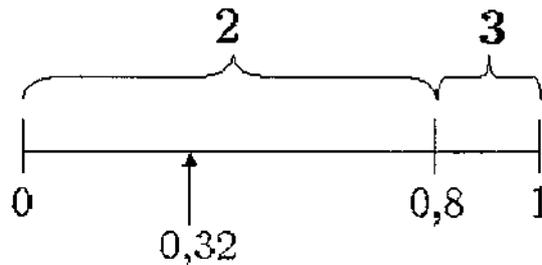
Parâmetro	Símbolo	Valor
Correção para o roteamento das formigas	α	0,5
Peso do modelo exponencial para a janela	η	0,005
Redutor para a janela do modelo paramétrico	c	0,3
Primeiro peso para o reforço adaptativo	c_1	0,7
Segundo peso para o reforço adaptativo	c_2	0,3
Nível de confiança para o reforço	γ	0,65
Amplificador da função de “amassamento”	a	5

A Equação 2.3 é usada para se definir as probabilidades a serem usadas na escolha do próximo *hop*. Supondo que no nó 1, $h_{25} = 0,6$ e $h_{35} = 0,4$, as probabilidades de se escolher o próximo *hop* são:

$$p_{25} = \frac{0,9}{(1 + 0,5) \times (0,9 + 0,1)} + \frac{0,5 \times 0,6}{(1 + 0,5) \times (0,6 + 0,4)} = 0,8$$

$$p_{35} = \frac{0,1}{(1 + 0,5) \times (0,9 + 0,1)} + \frac{0,5 \times 0,4}{(1 + 0,5) \times (0,6 + 0,4)} = 0,2$$

Um número aleatório entre 0 e 1 é sorteado e o seu valor é igual a 0,32. Usamos um esquema de seleção proporcional, alinhando em ordem crescente de vizinhos, até a soma ser igual a 1. Observando a Figura 2.6, que representa esse esquema, vemos que o vizinho 2 foi escolhido como próximo *hop*. A formiga adiciona esse vizinho à sua memória ($V_{1 \rightarrow 5}$) e migra para esse mesmo nó.

Figura 2.6: Exemplo de seleção probabilística do próximo *hop*.

No nó 2, o mesmo processo é repetido. Supondo que no nó 2, $h_{15} = 0,2$, $h_{45} = 0,3$ e $h_{55} = 0,5$, as probabilidades de se escolher o próximo *hop* são:

$$p_{15} = 0, \text{ nó 1 está na lista tabu, pois foi visitado anteriormente.}$$

$$p_{45} = \frac{0,1}{(1 + 0,5) \times (0,89 + 0,1)} + \frac{0,5 \times 0,3}{(1 + 0,5) \times (0,3 + 0,5)} = 0,1923$$

$$p_{55} = \frac{0,89}{(1 + 0,5) \times (0,89 + 0,1)} + \frac{0,5 \times 0,5}{(1 + 0,5) \times (0,3 + 0,5)} = 0,8077$$

Supondo que o número aleatório sorteado é igual à 0,79, então o próximo *hop* será o nó 5, que é adicionado pela formiga à sua memória. No próximo *hop*, como a formiga chegou em seu destino, ela se transforma na formiga $B_{5,-1}$. Neste ponto, a formiga deve retornar pela mesma rota do caminho de ida, atualizando as estruturas de dados dos nós visitados. Assim, ela migra para o primeiro nó intermediário, que é o nó 2.

Primeiramente, o modelo paramétrico estatístico é atualizado. Como $dist(V_{2-5}) = 1,0$, temos:

$$\mu_5^2 \leftarrow 1,1 + 0,005 \times (1 - 1,1) = 1,0995$$

$$\sigma_5^2 \leftarrow 0,1 + 0,005 \times (|1 - 1,1| - 0,1) = 0,1$$

Logo $I_{sup}^5 = 1,0995 + (0,1/\sqrt{(1 - 0,65) \times 300}) = 1,1971$. Assim, o reforço será igual a:

$$r_5 = 0,7 \times \frac{1}{1} + 0,3 \times \frac{1,1971 - 1}{(1,1971 - 1) + (1 - 1)} = 1$$

Em seguida, o reforço é limitado à 0,9 e o seu valor é “amassado”, resultando em um valor igual a 0,8539 ($a = 3$ vizinhos).

Na linha referente ao destino 5, o nó vizinho 5 recebe um reforço positivo:

$$\tau_{55} \leftarrow 0,89 + 0,8539 \times (1 - 0,89) = 0,9839$$

Enquanto os outros nós vizinhos (1 e 4) recebem um reforço negativo:

$$\tau_{15} \leftarrow 0,01 - 0,8539 \times 0,01 = 0,0015$$

$$\tau_{45} \leftarrow 0,1 - 0,8539 \times 0,1 = 0,0146$$

Como a formiga já atualizou o nó 2, o próximo nó cujas estruturas de dados devem ser atualizadas é o nó 1, para o qual a formiga migra.

Nesse nó, além da atualização referente ao nó destino 5, as entradas referentes ao nó destino 2 (tabela de roteamento e modelo paramétrico estatístico) também podem ser atualizadas se o caminho seguido até o nó 2 for estatisticamente bom.

O estimador superior para esse destino intermediário é:

$$I_{sup}^2 = 1,2 + (0,2/\sqrt{(1 - 0,65) \times 300}) = 1,2195$$

Uma vez que $dist(V_{2 \rightarrow 5}) = 1 < I_{sup}^2$, então serão feitas as atualizações das entradas referentes ao nó destino 5 e também das entradas referentes ao nó destino 2, conforme descrito anteriormente.

Finalmente, após essas atualizações, a formiga é descartada, pois já retornou ao nó origem.

2.5 Restauração

Restauração é o processo de recuperação das conexões ou rotas atingidas por uma falha, através da utilização da capacidade livre da rede após a ocorrência dessa falha.

A restauração não deve ser confundida com proteção, na qual os recursos necessários para o gerenciamento da falha já estão alocados e/ou operantes antes de sua ocorrência. Na proteção 1+1, por exemplo, para cada caminho óptico estabelecido existe um outro caminho óptico de reserva em uma rota disjunta ao primeiro caminho.

A restauração pode ser aplicada em diferentes níveis na rede. Um Label Switched Path (LSP) pode ter recuperação local, por segmento ou fim-a-fim [17].

Na recuperação local ou de enlace, a restauração é executada pelos nós à montante e à jusante do enlace que falhou, de maneira que o tráfego seja comutado para uma outra rota entre esses dois nós. Na recuperação por segmento, o tráfego do segmento atingido pela falha é comutado para outro segmento alternativo. Finalmente, na recuperação fim-a-fim, os nós origem e destino de cada conexão atingida pela falha são responsáveis pela sua recuperação, a fim de que o tráfego afetado seja comutado para rotas alternativas em torno da falha. Como a restauração fim-a-fim é a mais eficiente [17], ela foi usada neste trabalho.

Na restauração de um LSP, a rota do LSP alternativo pode já estar calculada antes da falha para agilizar o processo de restauração da conexão, sendo que este caso é chamado de “re-provisionamento de LSP”. Isso é interessante para o caso de falhas simples de enlace, mas no caso de uma segunda falha consecutiva esse processo pode não ter sucesso. Quando se calcula as rotas dos LSPs alternativos após a detecção e/ou notificação da falha, usa-se o termo “re-roteamento completo de LSP” [18]. Neste trabalho, consideramos o re-roteamento completo, mas assumimos que o cálculo das rotas alternativas tem um tempo de processamento que pode ser desprezado.

O re-roteamento completo tende a ser o mais eficiente em termos de uso de recursos e mais robusto no caso de múltiplas falhas, embora tenha um tempo de recuperação maior do que em esquemas de proteção. Entretanto, nenhuma aplicação requer a garantia de 50 ms da proteção 1+1 [19], sendo que uma interrupção de 2 segundos tem pouco impacto e é gerenciável pela maioria dos protocolos de transporte, a não ser a sinalização SS7, que é afetada após 200 ms de desconexão.

Há duas maneiras de se executar a operação de restauração: *hard* e *soft*. No primeiro caso, os recursos do LSP atingido pela falha são liberados antes do estabelecimento do LSP alternativo, enquanto no segundo caso os recursos são liberados após o estabelecimento. Essas estratégias

também são conhecidas como “break-before-make” e “make-before-break”, respectivamente. Em nosso caso, utilizamos a estratégia *hard* por ser de mais simples implementação.

Neste trabalho, as seguintes operações devem ser executadas em caso de falha [20, 18]:

Detecção da Falha Neste trabalho, assumiu-se que a detecção é feita de forma bidirecional, de forma que a falha é detectada pelos receptores dos nós mais próximos da montante e da jusante da falha. Em redes ópticas, a detecção é feita na camada física, sendo que o método mais usual é através da perda de luz – Loss Of Light (LOL).

Localização da Falha A localização da falha é feita através da comunicação entre os nós para determinar onde a falha ocorreu.

Notificação da Falha Quando os nós isolam a falha, esta é notificada aos nós responsáveis por seu tratamento.

Recuperação É executada pelos algoritmos de restauração, de forma a recuperar o tráfego atingida pela falha.

Os tempos para detecção e localização da falha foram considerados em conjunto e a sua soma é igual a I_{detect} .

A falha de nó é um caso especial de falhas múltiplas de enlace, onde todos os enlaces conectados ao nó falham simultaneamente.

A notificação da falha e o cálculo das rotas alternativas para a recuperação do tráfego são detalhados nas próximas sub-seções.

2.5.1 No caso de Roteamento por Caminho Mais Curto

Supondo que um protocolo do tipo enlace-estado está sendo usado, por exemplo o Open Shortest Path First with Traffic Engineering (OSPF-TE) [21, 22], os nós afetados pela falha (à montante e à jusante) notificam a falha aos demais nós da rede, inundando a rede com a nova

topologia física. Assim, quando cada nó recebe a mudança na topologia, ele re-calcula suas tabelas de roteamento.

2.5.2 No caso do Algoritmo Proposto

Após a falha do enlace l_{im} , no nó intermediário i , os valores da tabela de roteamento por ferormônio são atualizados de acordo com a seguinte expressão:

$$\tau_{nd} = \frac{\tau_{nd}}{\Sigma_{\tau}}, \forall n \in \mathcal{N}_i \wedge n \neq m, \quad (2.10)$$

onde $\Sigma_{\tau} = \sum_{n \neq m} \tau_{nd}$. Essa expressão é equivalente àquela apresentada em [23], mas escrita de uma forma mais clara.

Então, a linha m da tabela de roteamento por ferormônio é removida no nó i , pois o nó m não é mais um nó vizinho.

No caso de redes WRON, os nós afetados pela falha (à montante e à jusante) notificam somente os nós origem e destino das conexões afetadas pela falha, que tentam recuperar essas conexões. Mais detalhes da sinalização envolvida neste processo podem ser encontrados no Apêndice B.

No caso das redes tipo OPS, por outro lado, os cálculos de atualização em caso de falha são feitos apenas pelos nós afetados diretamente pela falha, isto é, falhas são gerenciadas de forma local. Isso permite uma rápida recuperação do tráfego afetado pela falha. Basta lembrar que, no caso de roteamento por caminho mais curto, todos os nós de rede devem ser notificados através dos protocolos de roteamento da mudança da topologia da rede.

2.6 Redes de Testes

Duas redes comumente usadas para a avaliação de desempenho de redes ópticas serviram de base para o teste do algoritmo proposto nesta Tese.

A primeira rede é conhecida como NSFNet, que era a antiga rede Internet de núcleo nos EUA no final dos anos 80 e início dos anos 90, sendo apresentada na Figura 2.7. É composta por 14 nós e 21 enlaces bi-direcionais. É bem balanceada² e tem o comprimento médio de caminho mais curto entre todos os pares origem-destino igual a 2,2.

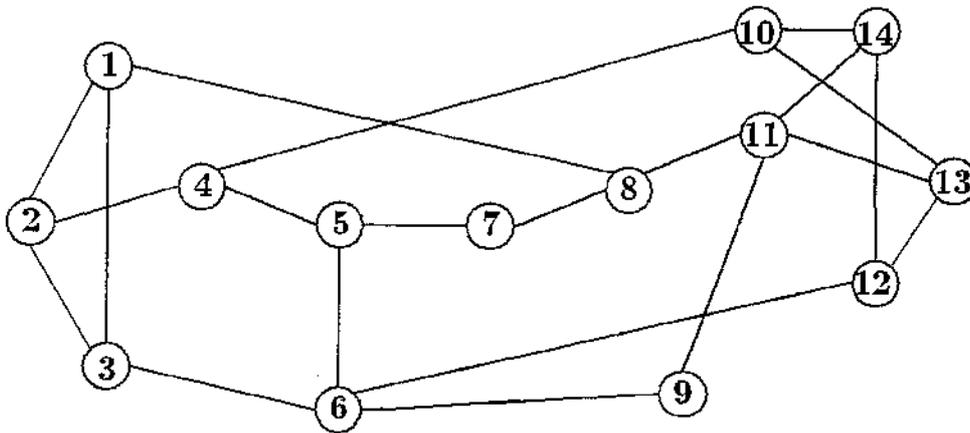


Figura 2.7: Rede NSFNet.

A outra rede usada nos testes é a NTTNet, pertencente à *Nippon Telephone and Telegraph*, que é mostrada na Figura 2.8. É formada por 57 nós e 81 enlaces bi-direcionais. Ao contrário da NSFNet, esta rede não é bem balanceada. Seu comprimento médio de caminho mais curto entre todos os pares origem-destino é igual a 6,5.

²Uma rede é dita bem balanceada se a distribuição dos caminhos mais curtos entre todos os pares origem-destino tem uma variância pequena.

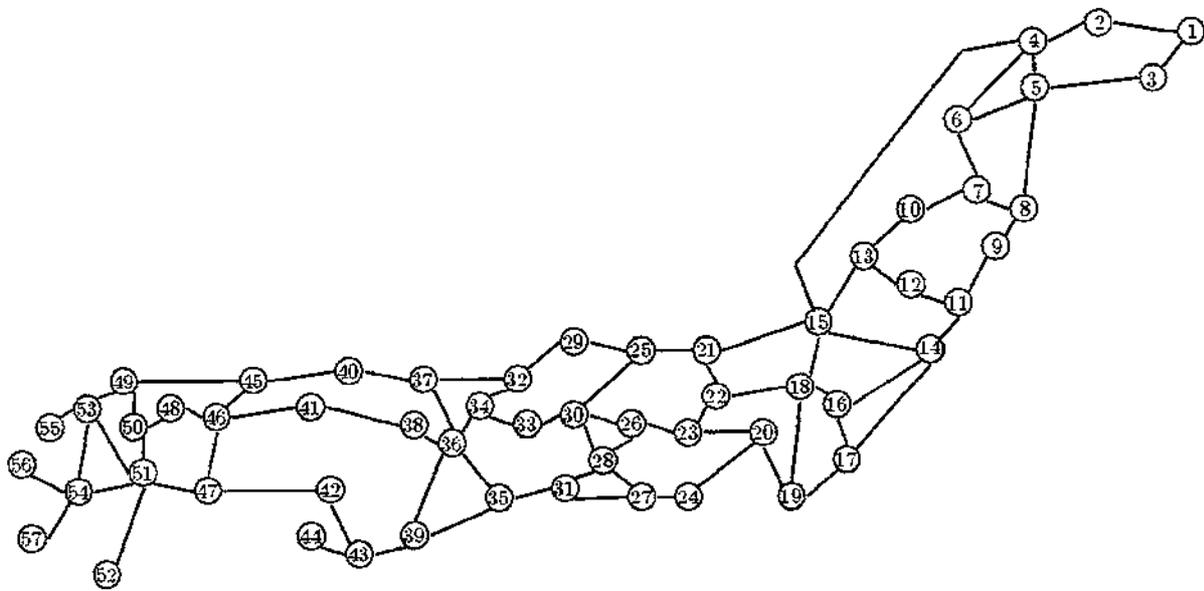


Figura 2.8: Rede NTTNet.

Capítulo 3

Roteamento e Alocação de Comprimento de Onda (RWA)

O algoritmo de roteamento e alocação de comprimento de onda – Routing and Wavelength Assignment (RWA) tenta estabelecer um caminho óptico¹ para cada requisição de conexão [24]. Tipicamente, as requisições de conexão podem ser de três tipos: estática, incremental ou dinâmica.

No cenário estático, todas as requisições são conhecidas com antecedência e o objetivo do algoritmo é, por exemplo, minimizar a probabilidade de bloqueio dado um número de comprimentos de onda disponíveis. No tráfego incremental, as requisições chegam seqüencialmente e os caminhos ópticos são permanentes. Finalmente, no cenário dinâmico, um caminho óptico é estabelecido à medida que as requisições chegam e permanecem ativos por uma duração finita. Redes dinâmicas de caminhos ópticos são remanescentes das tradicionais redes comutadas por circuitos e, de fato, são classificadas como tais. Em geral, o objetivo dos casos incremental e dinâmico é minimizar a probabilidade total de bloqueio de conexões.

Na falta de conversores de comprimento de onda nos nós, as rotas devem obedecer à

¹Caminho óptico é um sinal óptico que vai de um nó origem a um nó destino utilizando um comprimento de onda nos enlaces que ligam esses nós.

restrição de continuidade de comprimento de onda, isto é, um caminho óptico deve ter o mesmo comprimento em todos os enlaces atravessados por ele, o que resulta em uma probabilidade de bloqueio maior.

Como o problema de RWA é difícil de ser resolvido, por ser do tipo NP-difícil [25], em geral, ele é dividido em dois sub-problemas separados, o que permite a simplificação do algoritmo: o sub-problema de roteamento e o sub-problema de alocação de comprimento de onda, cada qual com várias heurísticas propostas [24].

Neste Capítulo, apresentamos um algoritmo baseado em ACO que pode ser usado para resolver o sub-problema de roteamento.

3.1 Demonstração intuitiva do funcionamento da ACO no problema do RWA

Em topologias lineares sem conversão de comprimento de onda, o problema do RWA é reduzido à alocação do comprimento de onda. Em cada plano disponível de comprimento de onda, uma requisição de caminho óptico deve ser alocada em um espaço vazio entre duas rotas ativas vizinhas. Em [26], foi mostrado que a probabilidade de bloqueio é minimizada quando o comprimento de onda é alocado para um espaço vazio que por sua vez é contido em outros maiores. Se nenhum espaço vazio é contido em outro, então é melhor alocar o plano de comprimento de onda onde a nova rota é inserida mais assimetricamente no espaço vazio, ou seja, o mais próximo possível de uma rota ativa vizinha. Em outras palavras, boas rotas devem ficar próximas de obstáculos pré-existentes. Não há nenhuma razão para que isto não seja válido em redes emaranhadas também, mas é muito mais difícil descobrir métricas adequadas para a melhor e mais assimétrica inserção de uma rota requisitada em um espaço vazio na rede emaranhada. Uma estratégia baseada em formigas pode dar uma resposta apropriada para esse problema.

Formigas são míopes, de forma que é necessário outra maneira delas se guiarem no ambi-

ente. Essa maneira é o uso de ferormônios, que são depositados pelas formigas que previamente passaram pelo mesmo ponto. Direções com maior quantidade de ferormônio tendem a ser favorecidas na procura por comida e/ou no retorno para o formigueiro. Quando as formigas encontram um obstáculo enquanto procuram por uma fonte de comida, suas opções são reduzidas, de tal maneira que o ferormônio tende a acumular-se na vizinhança dos obstáculos. Presumivelmente, esta é uma boa característica para rotas de formigas, pois isso reduz a probabilidade de que a rota seja pisada por algum animal.

De fato, essa atração aos obstáculos é também uma boa característica para redes de comunicação, como provado matematicamente para topologias lineares em [26]. A razão é que enlaces e nós próximos a enlaces ativos (obstáculos) são mais prováveis de participarem de requisições de rotas bloqueadas, mesmo se os recursos estiverem livres. Como custam menos em termos de probabilidade de bloqueio adicional para suportar novas conexões, eles são bons candidatos para fazerem parte de novas rotas alocadas sob um critério de mínimo custo.

Como exemplo, consideremos a topologia mostrada na Figura 3.1. Seja uma requisição de uma conexão feita do nó formigueiro A para algum recurso que é atualmente disponível em B , que é o nó com a fonte de comida. Note que a rota mais curta de A para B deve conter 4 passos à direita e 4 passos para cima. Como esses passos podem ser executados em qualquer ordem, há $\binom{8}{4} = 70$ rotas possíveis, de forma que um algoritmo simples de caminho mais curto pode selecionar qualquer uma delas. Entretanto, algumas alocações são muito melhores que outras quando se leva em consideração a probabilidade de bloqueio produzida para futuras requisições.

Considere, por exemplo, a alocação produzida quando se alterna passos para a direita e para cima, o que parece ser natural para uma agente míope, como mostrado na Figura 3.1a. A rota divide a rede em duas metades distintas, resultando em uma probabilidade de bloqueio de cerca de $1/2$, o que é uma performance muito pífia. Por outro lado, uma rota em L, como a mostrada na Figura 3.1b, exaure as portas do nó C , de forma a bloquear 15 dos 120 pares possíveis, resultando em uma probabilidade de bloqueio de $1/8$ sob tráfico uniforme. Alguém

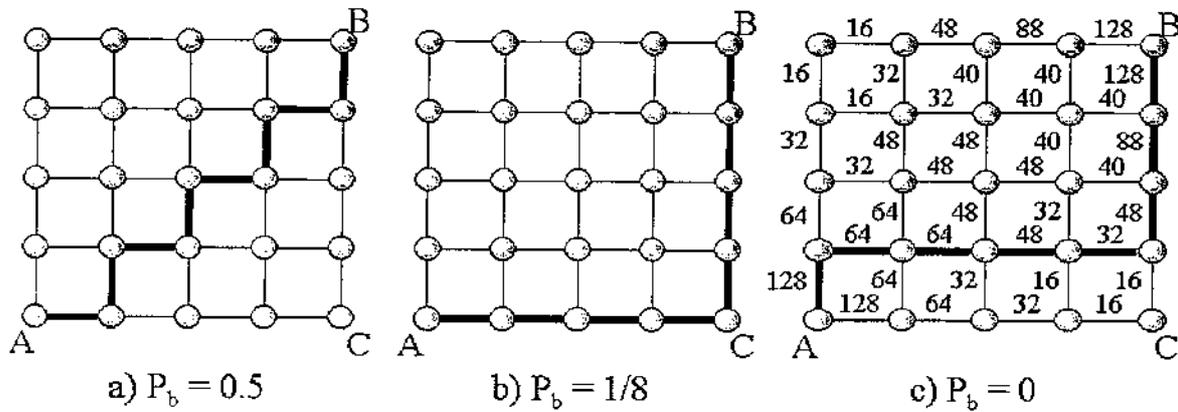


Figura 3.1: Uma rota descoberta por formigas (c) e duas rotas ineficientes (a e b).

pode se perguntar: existe alguma escolha que mantém a probabilidade de bloqueio em zero após a alocação da rota?

Como um algoritmo baseado no comportamento das formigas pode ser usado para resolver este problema? Para procurar algum recurso necessário (“comida”), o nó A libera simultaneamente um número N de formigas que estão programadas para executar movimentos para cima e para a direita de forma aleatória até que encontrem comida, e retornem através do caminho com maior quantidade de ferormônio, enquanto reservam recursos para o estabelecimento de um caminho óptico ou para uma rajada passar.

Para conveniência e sem perda de generalidade, seja $N = 2^8 = 256$ formigas. A Figura 3.1c mostra o número médio de formigas indo através de cada enlace antes que todas se encontrem no nó B . Este número é agora a quantidade de ferormônio depositado no enlace e se torna uma métrica a ser usada no roteamento. Quando as formigas retornam para o formigueiro, elas favorecerem os enlaces com maior quantidade de ferormônio, de forma a seguir a rota mostrada na Figura 3.1c. Note que a rota resultante mantém a probabilidade de bloqueio em zero e uma boa conectividade da fonte de comida à maioria dos nós da rede. Nos primeiros quatro passos em sua procura por comida, o ferormônio tende a se espalhar de uma forma binomial ou gaussiana, concentrando-se em torno da diagonal. Após o quinto passo, entretanto, as formigas começam a

bater nos limites da rede (como se fosse um espaço vazio livre no plano de comprimento de onda de uma rede maior), acumulando, assim, ferormônio nessas regiões e gerando uma boa escolha de rota. Isto significa que a miopia individual de cada formiga pode levar a uma visão geral coletiva usando-se um comportamento simples, mas organizado.

3.2 Trabalhos prévios na literatura

Existem poucos trabalhos que usam a ACO como plataforma para algoritmos de RWA e a lista apresentada nesta Seção não é exaustiva. Entretanto, nenhum desses trabalhos apresentou realmente como um sistema baseado em ACO pode ser implementado em uma rede óptica.

O primeiro exemplo do uso de ACO no problema de RWA tratou do caso estático [27]. Usando diferentes colônias de formigas que são repelidas entre elas, o algoritmo apresentado tentava minimizar o número de comprimentos de onda necessários na rede enquanto mantinha o tamanho das rotas curtos.

Outro exemplo de algoritmo baseado em ACO para o caso estático foi apresentado em [28], onde nenhuma capacidade de conversão de comprimento de onda é assumida.

O primeiro artigo que encontramos e que focava no problema de RWA dinâmico foi [29]. Este trabalho incorpora o tamanho da rota e informação de congestionamento nas decisões de roteamento. Um número fixo de formigas é lançado quando uma requisição chega de forma que não há formigas circulando na rede entre requisições de conexão.

Em [30, 31], formigas estão continuamente sendo lançadas, independentemente da chegada de conexões. O principal ponto negativo presente nesses trabalhos é que cada formiga deve portar em sua memória o estado dos comprimentos de onda de cada enlace, além de uma identificação de cada nó visitado. O algoritmo proposto nesta Tese mostra que somente a identificação dos nós visitados é suficiente para se alcançar um roteamento que usa informações de congestionamento.

3.3 Variações do Algoritmo Proposto

A métrica heurística local leva em conta o complemento do congestionamento de cada enlace vizinho. Assim, ela é calculada da seguinte forma:

$$h_{nd} = \frac{l_n^a}{W}, \quad (3.1)$$

onde l_n^a é o número de comprimentos de onda disponíveis no vizinho n e W é o número total de comprimentos de onda implementados no enlace.

3.3.1 Restauração

Quando ocorre uma falha, as tabelas de roteamento por ferormônio adjacentes à falha são atualizadas conforme descrito na Seção 2.5. Entretanto, somente essa atualização não é suficiente para um bom re-roteamento das conexões afetadas pela falha, uma vez que a recuperação é fim-a-fim, enquanto a atualização é feita localmente para o enlace afetado, de forma que a solução final pode não ser ótima.

Assim, uma ou mais novas tentativas podem ser feitas após um intervalo de tempo $I_{holdoff}$ pequeno, possibilitando a exploração da rede em busca de rotas alternativas. Para aumentar a eficiência desse processo, os nós origens das conexões afetadas pela falha, imediatamente após a notificação da falha, começam a lançar formigas com uma taxa $R_{ants}^{failure}$ em direção ao nó destino.

Essas novas tentativas tendem a melhorar a capacidade de restauração da rede, sendo que normalmente são necessárias poucas tentativas extras para se obter bons resultados [17]. Para compararmos o desempenho dessa estratégia, o mesmo número de rotas alternativas no roteamento fixo-alternado será usado como valor das tentativas do algoritmo baseado em ACO.

3.4 Simulação

Consideramos um tráfego Poissoniano homogêneo com um perfil espacial uniforme. A duração de cada caminho óptico segue uma distribuição exponencial com o valor médio de 100 segundos. Há W comprimentos de onda disponíveis por enlace na rede.

Para a rede NSFNet, usamos 10^7 requisições para o roteamento fixo-alternado e 10^6 requisições para o roteamento baseado em ACO. E para a rede NTTNet, usamos 10^6 requisições para o roteamento fixo-alternado e 10^5 para o roteamento baseado em ACO. Em todos os casos, os valores de probabilidade de bloqueio já haviam convergido.

Para o sub-problema de alocação de comprimento de onda, usamos uma estratégia *first-fit* para ambos os algoritmos, pois estamos comparando a eficiência do roteamento.

O mecanismo de estabelecimento de caminhos ópticos com o auxílio de um plano de controle GMPLS é detalhado no Apêndice B.

Os parâmetros usados nas simulações são mostrados na Tabela 3.1:

Tabela 3.1: Parâmetros usados na simulação dos algoritmos de RWA.

Parâmetro	Símbolo	Valor
Taxa global de lançamento de formigas	R_{ants}	100 formigas/s
Intervalo para chegada das requisições	I_{warmup}	100 s
Número de comprimentos de onda por enlace	W	8
Correção para o roteamento das formigas	α	0,6
Peso do modelo exponencial para a janela	η	0,005
Redutor para a janela do modelo paramétrico	c	0,3
Primeiro peso para o reforço adaptativo	c_1	0,7
Segundo peso para o reforço adaptativo	c_2	0,3
Nível de confiança para o reforço	γ	0,65
Amplificador da função de “amassamento”	a	5
Tempo para detectar e localizar uma falha	I_{detect}	10 ms
Tempo de espera para nova tentativa de restauração	$I_{holdoff}$	200 ms
Taxa de lançamento de formigas pelos nós afetados por falha	$R_{ants}^{failure}$	1000 formigas/s

Foram feitos inúmeros testes para se avaliar a influência dos parâmetros apresentados na

Tabela 3.1. Não há muita diferença entre os resultados obtidos quando se mantém $\alpha \in [0,5; 0,7]$, $c_1 \in [0,6; 0,7]$, $c_2 \in [0,3; 0,4]$ e $a \in [5; 7]$. Os parâmetros usados neste problema são muito próximos aos parâmetros propostos no algoritmo AntNet original.

Além disso, o aumento da taxa de lançamento de formigas na rede (R_{ants}) além do valor apresentado na Tabela 3.1 não melhora o resultados obtidos.

Assim, os valores de parâmetros que apresentaram os melhores resultados na simulação estão mostrados na Tabela 3.1 e são os mesmos para as duas topologias testadas. Isso é um indicativo que esses valores tendem a ser robustos para diferentes topologias e/ou diferentes cargas na rede.

3.5 Resultados

Primeiramente, usamos a rede NSFNet para testar o desempenho do algoritmo baseado em ACO em relação aos roteamentos fixo e fixo-alternado, como podemos observar na Figura 3.2.

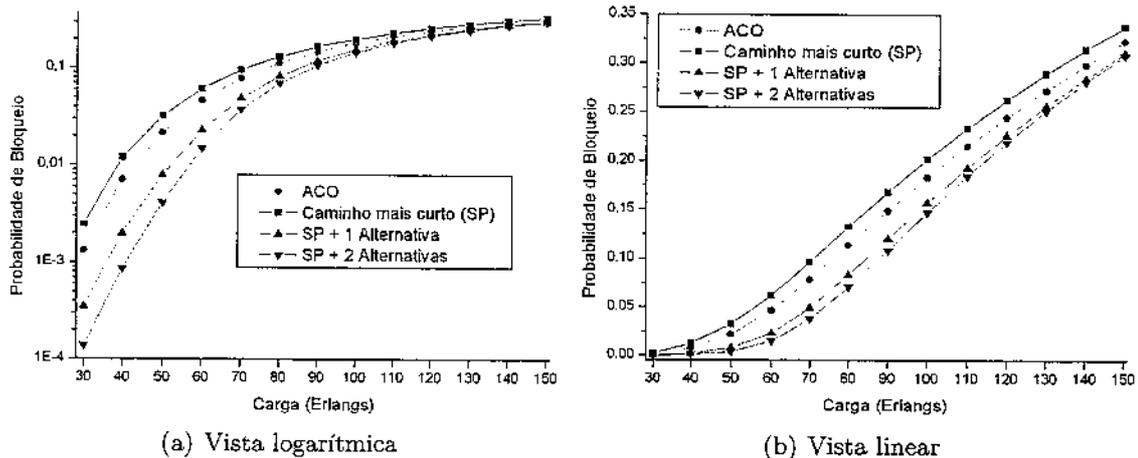


Figura 3.2: Roteamento fixo-alternado versus roteamento baseado em ACO para a rede NSFNet.

As mesmas simulações foram feitas para a rede NTTNet, como pode ser visto na Figura 3.3.

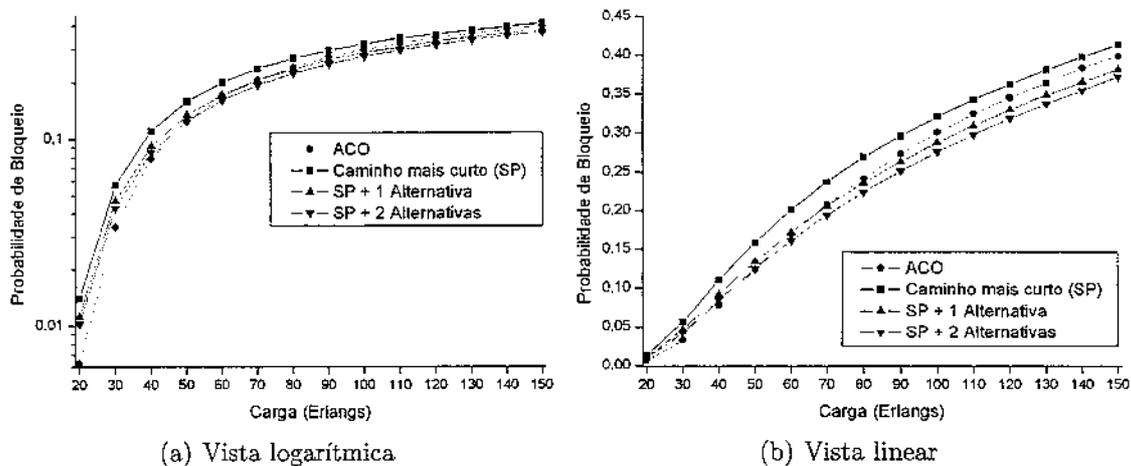


Figura 3.3: Roteamento fixo-alternado versus roteamento baseado em ACO para a rede NTTNet.

Na Figura 3.2, foi obtida alguma melhoria em termos de probabilidade de bloqueio sobre o roteamento por caminho mais curto, mas não sobre o roteamento fixo-alternado. Por outro lado, foram obtidos melhores resultados quando se utiliza a rede NTTNet, principalmente com cargas mais baixas. A rede NTTNet não é bem-balanceada como a NSFNet e tem mais opções de rotas alternativas com mesmo número de *hops*, de forma que há mais chances de se obter uma rota menos congestionada para se estabelecer um caminho óptico.

Na Figura 3.4, é possível observar os *hot-spots*, que são os nós que mais contribuem para o bloqueio de requisições. O roteamento baseado em ACO consegue reduzir o bloqueio da maior parte dos nós afetados, quando se compara com os esquemas de roteamento fixo e fixo-alternado, aumentando a imparcialidade do bloqueio em relação aos nós.

Finalmente, podemos ver na Figura 3.5 outro efeito resultante do balanceamento de carga do algoritmo baseado em ACO. Ele é capaz de re-rotear requisições que seriam bloqueadas no roteamento fixo para rotas mais longas, mas menos congestionadas. Em outras palavras, o algoritmo é capaz de minimizar a probabilidade de bloqueio das conexões através de desvios estabelecidos dinamicamente do tráfego “excedente” para rotas mais longas. A figura mostra a percentagem das conexões estabelecidas em relação ao número total de requisições da simulação. Note que a soma dessas percentagens sobre todos os comprimentos de rota é igual ao comple-

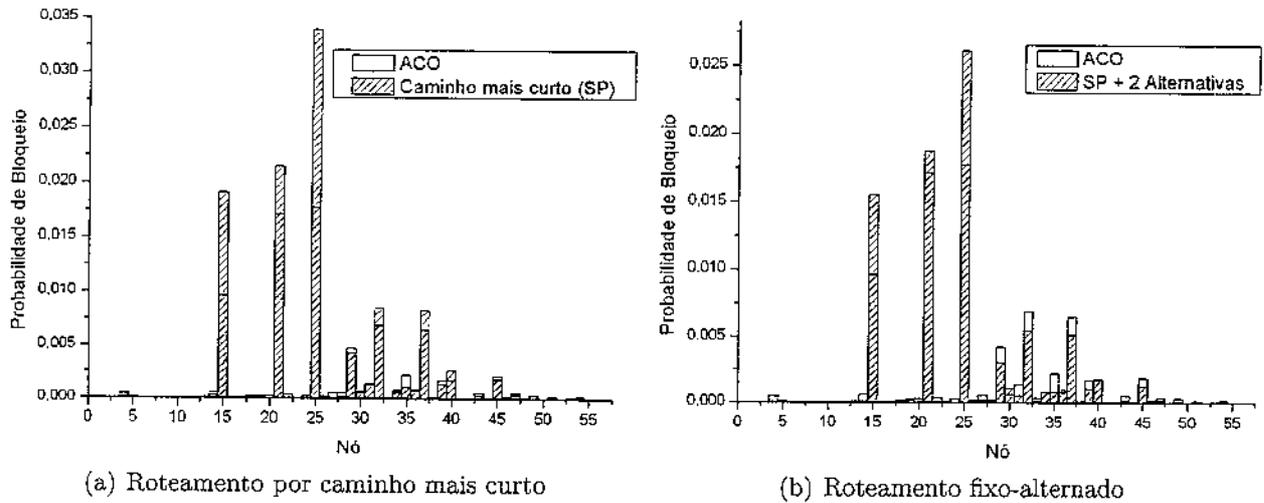


Figura 3.4: Hot-spots para a rede NTTNet com carga = 40 erlangs

mento da probabilidade de bloqueio mostrada na Figura 3.3 para uma carga de 20 erlangs.

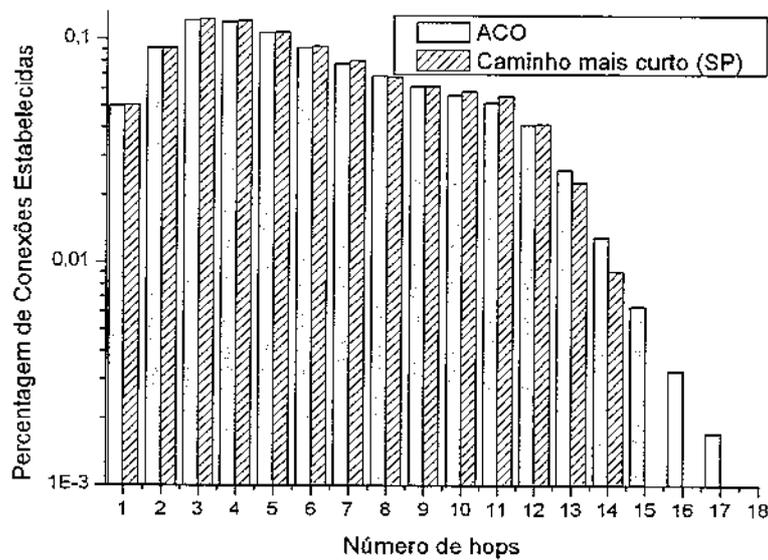


Figura 3.5: Percentagem das conexões estabelecidas por comprimento da rota na rede NTTNet com uma carga = 20 erlangs.

3.5.1 Restauração

Primeiramente, verificou-se a influência dos parâmetros que mais influem no roteamento por formigas: o parâmetro α , por ser responsável pelo balanço entre os níveis de ferormônio e a métrica da heurística local, e os parâmetros c_1 e c_2 , que são responsáveis pelo reforço adaptativo na atualização dos níveis de ferormônio.

A capacidade de restauração R é o aspecto mais importante para se comparar o desempenho entre diferentes algoritmos e é expresso pela seguinte fórmula:

$$R = \frac{C_{restored}^{failed}}{C_{failed}^{failed}}, \quad (3.2)$$

onde C_{failed}^{failed} é o número de conexões afetadas pela(s) falha(s) e $C_{restored}^{failed}$ é o número de conexões afetadas e que foram recuperadas pelo processo de restauração.

A Figura 3.6 mostra o desempenho do algoritmo baseado em ACO para restauração em caso de falha simples de nó para a rede NSFNet.

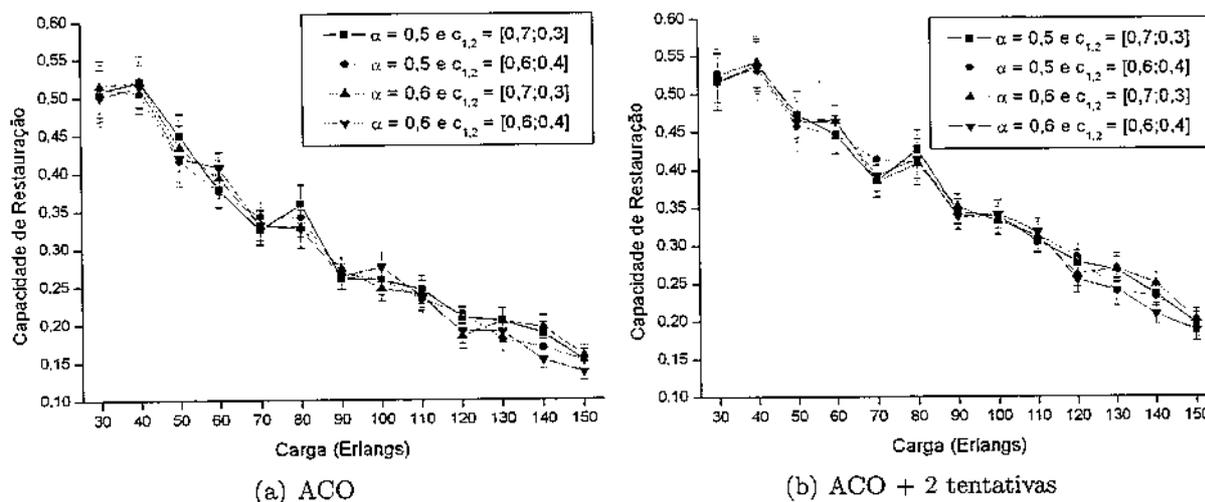


Figura 3.6: Influência dos parâmetros na capacidade de restauração da rede NSFNet em falha simples de nó para o algoritmo baseado em ACO.

E a Figura 3.7 mostra o desempenho da restauração por formigas no caso de falha simples de enlace na rede NSFNet.

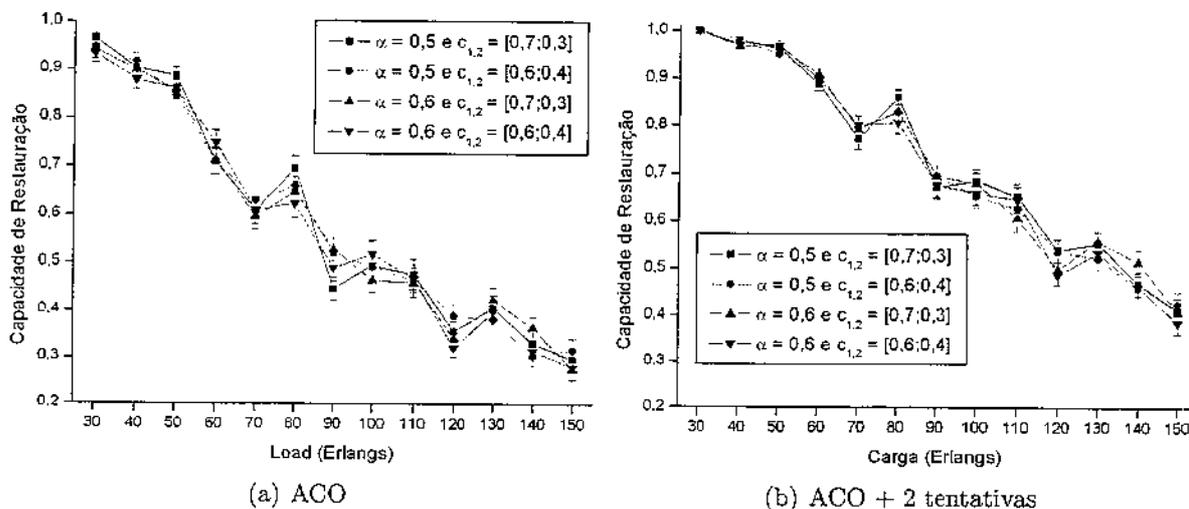


Figura 3.7: Influência dos parâmetros na capacidade de restauração da rede NSFNet em falha simples de enlace para o algoritmo baseado em ACO.

Tanto na falha simples de nó como na falha simples de enlace, a capacidade de restauração é bastante robusta aos parâmetros usados. Entretanto, de forma mais ou menos habitual, os valores de $c_1 = 0,7$ e $c_2 = 0,3$ têm um desempenho um pouco superior em relação aos valores de $c_1 = 0,6$ e $c_2 = 0,4$. Por esse motivo, as figuras seguintes considerarão $c_1 = 0,7$ e $c_2 = 0,3$.

Em ambas as Figuras 3.6 e 3.7, é possível notar que as duas tentativas de restauração após um curto intervalo de tempo aumentam significativamente a capacidade de restauração do algoritmo. Supondo que o processo de sinalização dure por volta de 50 ms, a restauração com até duas tentativas é feita em até 500 ms ($2 \times [I_{holdoff} + 50ms]$), o que é ainda muito abaixo do limiar de 2 segundos para afetar as conexões em TCP/IP ou provocar reações na camada IP/MPLS [19].

A Figura 3.8 mostra a capacidade de restauração de diferentes estratégias no caso de falha simples de nó na rede NSFNet em função da carga da rede.

A Figura 3.9 mostra a capacidade de restauração de diferentes estratégias no caso de falha simples de enlace na rede NSFNet em função da carga da rede.

Assim, podemos constatar que a restauração que usa somente a memória das tabelas de

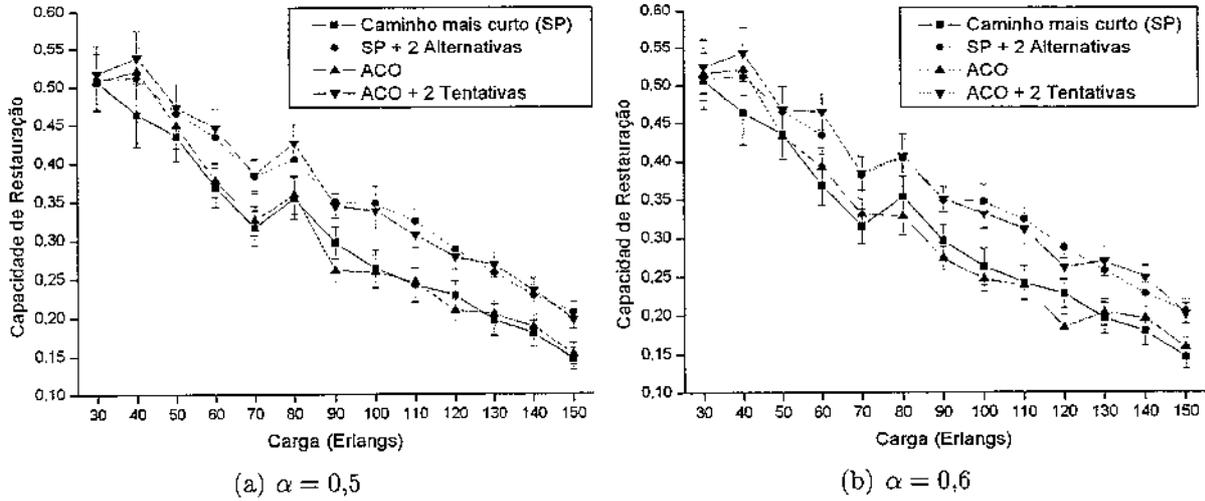


Figura 3.8: Capacidade de restauração no caso de falha simples de nó para a rede NSFNet.

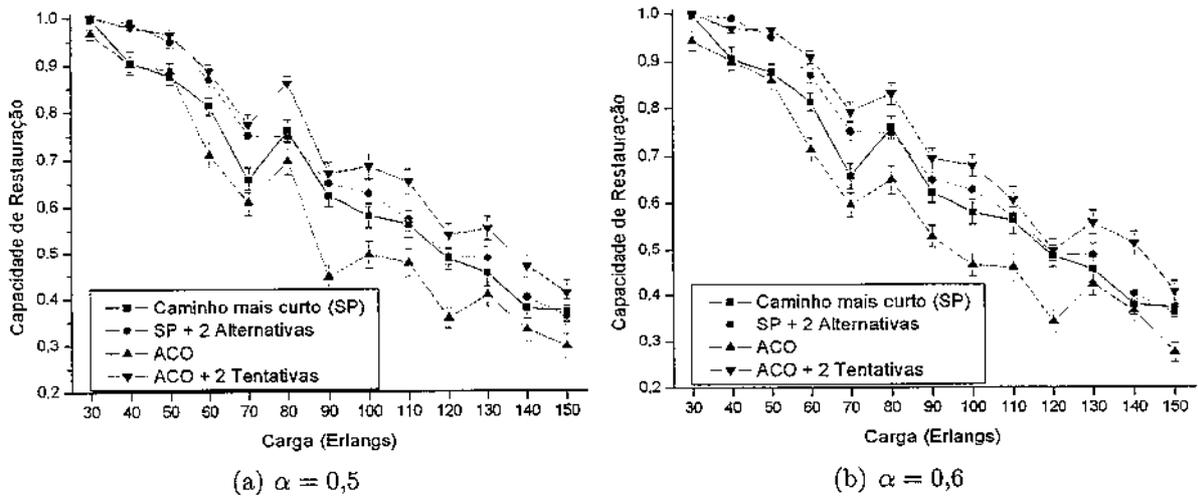


Figura 3.9: Capacidade de restauração no caso de falha simples de enlace para a rede NSFNet.

ferormônio é ineficiente, pois esse mecanismo tem apenas efeito local. Permitir que as formigas explorem rotas em uma situação de emergência, que é o caso de falhas na rede, possibilita alcançar uma capacidade de restauração superior àquela alcançada no roteamento fixo e fixo-alternado na maioria dos casos observados.

As mesmas simulações são repetidas para a rede NTTNet. Na Figura 3.10, é mostrada a capacidade de restauração das diferentes estratégias no caso de falha simples de nó em função da carga da rede.

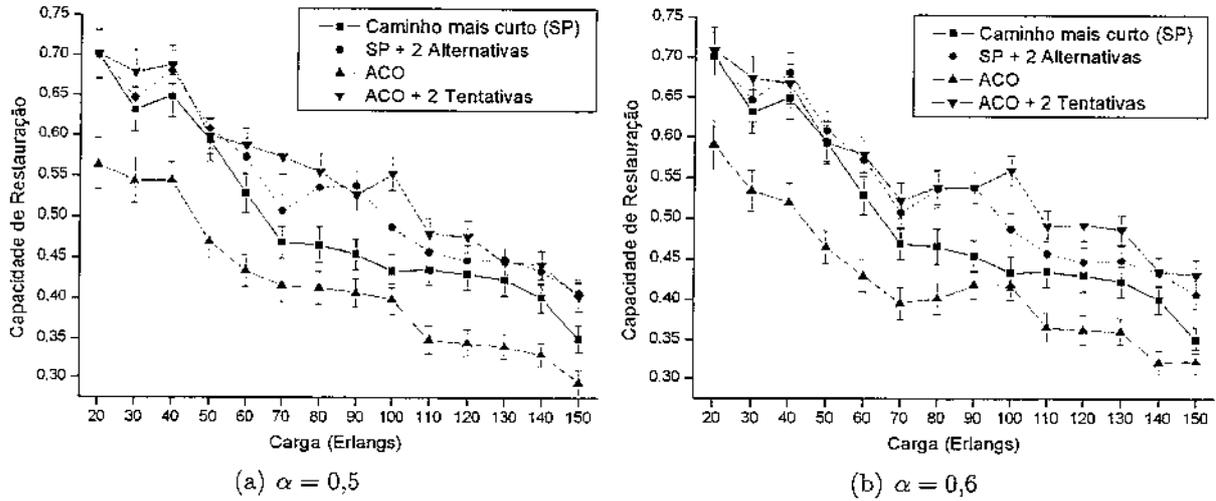


Figura 3.10: Capacidade de restauração no caso de falha simples de nó para a rede NTTNet.

E, finalmente, na Figura 3.11 é mostrada a capacidade de restauração das diferentes estratégias no caso de falha simples de enlace em função da carga da rede.

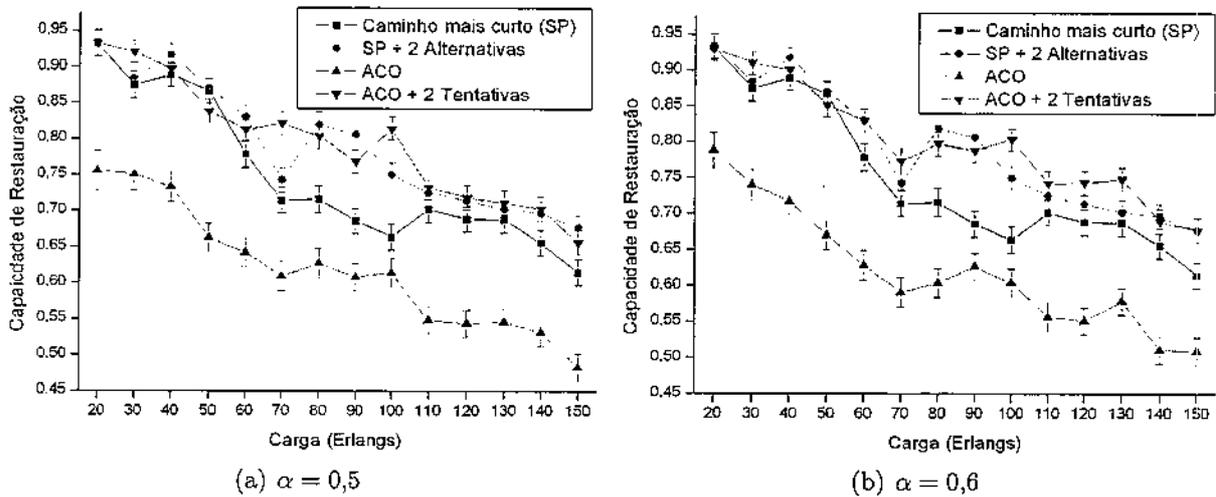


Figura 3.11: Capacidade de restauração no caso de falha simples de enlace para a rede NTTNet.

Como já discutido para a rede NSFNet, permitir que o algoritmo baseado em ACO faça mais tentativas no sentido de recuperar as conexões afetadas por falha, permite alcançar maiores níveis de capacidade de restauração, de forma que esse tipo de restauração seja, na média, mais eficiente que a restauração por k -rotas.

Em ambas as redes, é difícil decidir o valor mais apropriado de α . Aparentemente, $\alpha = 0,6$ leva uma ligeira vantagem sobre $\alpha = 0,5$. Contudo, o mais importante é considerar a robustez do algoritmo em relação aos valores dos parâmetros.

3.5.2 Considerações finais

O algoritmo proposto tende a ser mais eficiente quando a rede tem várias rotas possíveis com o mesmo número de *hops*. As formigas não necessitam de carregar outra informação além de uma identificação dos nós visitados.

Cada formiga que vaga pela rede tende a introduzir pequenas melhorias nas tabelas de roteamento. Quando se faz uma requisição de conexão, a informação contida nas tabelas de roteamento é próxima do conhecimento completo e global do estado da rede. Assim, o tempo necessário para o estabelecimento de conexões é pequeno e o roteamento está otimizado para se reduzir a taxa de bloqueio.

Há outras vantagens de se usar algoritmos de roteamento baseados em formigas. O primeiro aspecto está relacionado ao fato que este tipo de algoritmo tende a ser mais escalável, uma vez que as tabelas de roteamento crescem linearmente com o número de nós. No caso do roteamento fixo-alternado, por outro lado, as tabelas de roteamento crescem de forma quadrática com o número de nós, pois são necessárias entradas para cada par origem-destino da rede.

Se as formigas são usadas na descoberta de novos nós na rede, não se faz necessário o uso do protocolo OSPF-TE no plano de controle, pois se obtém um bom desempenho em termos de probabilidade de bloqueio com o emprego de ACO, a qual utiliza informação de congestionamento nos enlaces para o roteamento local das formigas.

Capítulo 4

Degenerações na Camada Física

Em uma rede transparente não há regeneração do sinal, de forma que ruído e distorção do sinal acumulam ao longo do caminho óptico. Assim, em redes de longa distância, algumas rotas não são passíveis de serem estabelecidas, pois não possuem um desempenho de transmissão aceitável. De fato, alguns trabalhos [32, 33, 34, 35, 36, 37, 38] indicaram a necessidade de se considerar as degenerações na camada física no provisionamento de caminhos ópticos.

Neste Capítulo, o foco é no ruído de emissão espontânea amplificada – Amplified Spontaneous Emission (ASE), pois esse ruído costuma ser o fator limitante para sistemas de longa distância [39] devido ao cascadeamento de amplificadores à fibra dopada de érbio – Erbium Doped Fiber Amplifier (EDFA) e ao fato da ASE ser facilmente modelada analiticamente.

Assim, a máxima taxa de erros de bits – Bit Error Rate (BER) aceitável, que leva em conta o ruído gerado pelos amplificadores, determina a mínima potência que um sinal pode ter ao longo da rede. Essa variante do problema do RWA é conhecida como RWA-P, ou seja, RWA com restrições de potência. Essa variante foi introduzida em [34, 40], com um valor fixo de potência mínima, e estendida em [41, 38], os quais propõem um método analítico que relaciona a BER e a potência mínima em cada componente da rede.

As restrições de potência podem ser definidas da seguinte forma. Um caminho óptico entre

dois nós da rede, o qual atravessa vários enlaces de fibra e encontra diversos dispositivos ópticos, deve manter níveis aceitáveis de potência ao longo de toda a sua rota. Além disso, o nível de potência de um sinal individual deve ser mantido de tal maneira que a potência agregada (isto é, a soma dos níveis de potência de todos os sinais) não pode exceder um determinado valor. O que torna esse modelo complicado é o ganho do amplificador óptico, que é dependente da potência agregada que passa por ele.

Outras degenerações na camada física, como os efeitos não-lineares da fibra, não são considerados, pois tais degenerações criam restrições que consideram rotas em conjunto, conduzindo a complexas interdependências, tais como a ordem com que se atravessa fibras específicas [35]. Além disso, o tratamento completo dessas restrições não-lineares requer um conhecimento total da infraestrutura física da rede, incluindo valores de dispersão para cada enlace de fibra, área do núcleo da fibra e sua composição, além de detalhes de subsistemas como a tecnologia de controle de dispersão. Essas informações combinadas com o estado de tráfego da rede podem determinar o nível de degeneração causado pelas não-linearidades. Claramente, isso é muito mais complicado do que o modelo proposto, que só considera a ASE como degeneração da rede. Entretanto, como a potência máxima usada em cada componente da rede é controlada, esses efeitos não lineares são indiretamente gerenciados devido a sua extrema dependência da potência do sinal óptico.

Além disso, este Capítulo apresenta como o RWA-P pode ser usado em uma rede com uma plano de controle GMPLS integrado. Em [42], também foi proposta uma arquitetura distribuída para o roteamento que leva em consideração as restrições de camada física, mas essa arquitetura introduziu modificações aos protocolos de sinalização e gerenciamento de enlace do padrão GMPLS. Neste trabalho, pelo contrário, o algoritmo proposto não necessita de modificação alguma nos protocolos do GMPLS.

Na Seção a seguir, o RWA-P é explicado em mais detalhes.

4.1 RWA com Restrições de Potência (RWA-P)

Para um cenário dinâmico, o problema RWA-P pode ser definido da seguinte forma: dado uma topologia de rede e um conjunto de requisições que chegam dinamicamente na rede, o objetivo é minimizar a probabilidade de bloqueio das conexões através do roteamento, alocação de comprimento de onda, a manutenção de um nível aceitável de potência óptica e uma relação sinal-ruído – Signal-Noise Ratio (SNR) adequada em toda a rede. Assim, devem-se obedecer às seguintes restrições:

1. Uma conexão é roteada usando um mesmo comprimento de onda em todos os enlaces da rede;
2. Duas conexões não podem compartilhar um mesmo comprimento de onda em um mesmo enlace;
3. A potência do sinal é mantida acima de um limiar (P_{sen}) da origem até o destino.
4. A potência total de todos os sinais em um enlace deve ser menor ou igual a P_{max} , onde P_{max} é a potência agregada máxima permitida em um enlace.

A restrição de potência mínima, ou nível de sensibilidade, assegura que o sinal óptico pode ser detectado em todos os componentes ópticos. A restrição de potência mínima garante a minimização das degenerações na camada física do tipo não-linear, pois limita a potência agregada do enlace a um valor máximo.

Uma das dificuldades desse problema é que o ganho do amplificador óptico é uma grandeza não-determinística dependente do tráfego [33]. Além disso, o ruído ASE acumulado satura ainda mais os amplificadores.

4.1.1 Ruído ASE

O ruído ASE é o ruído dominante gerado pelo EDFA. Essa emissão dá origem a um largo espectro de fótons que são amplificados junto com o sinal óptico.

O processo de emissão espontânea não contribui para o ganho de primeira ordem do amplificador. Embora os fótons emitidos tenham a mesma energia que os fótons incidentes do sinal óptico, eles são emitidos em direção, polarização e fase aleatórios. Esse processo é diferente do processo de emissão estimulada, onde os fótons emitidos têm a mesma direção, polarização e fase dos fótons incidentes, e que é responsável pela amplificação do sinal óptico.

A emissão espontânea prejudica o sistema óptico. O amplificador não consegue distinguir o sinal óptico das emissões espontâneas, amplificando ambas. Assim, em redes ópticas de grande distâncias, que se valem do cascadeamento de amplificadores ópticos para compensar as perdas do sistema, a ASE se torna o fator mais crítico que deve ser gerenciado. Há duas razões para isso [43]: a ASE vai se acumulando pelos amplificadores e degrada a relação sinal-ruído quando o número de amplificadores aumenta. A segunda razão é que, conforme o nível de ASE aumenta, ela contribui para saturar os amplificadores ópticos, reduzindo, dessa forma, o ganho dos amplificadores que estão localizados mais a frente no enlace. O resultado final é que o sinal tende a diminuir e a ASE tende a aumentar em sistemas de longa distância, quando se vai aumentando o número de amplificadores ópticos, até que, em um caso limite, a BER fique inaceitável.

4.1.2 Nível de sensibilidade

A potência mínima ou nível de sensibilidade de cada componente da rede é calculada baseando-se no ruído ASE e pode ser determinado pela seguinte expressão [41, 38]:

$$P_{sen} = 4\gamma^2 N_{sp} h f_c B_e \left\{ 1 + \sqrt{1 + \frac{\frac{B_o}{B_e} - \frac{1}{2}}{4\gamma^2}} \right\} \quad (4.1)$$

onde N_{sp} é o fator de emissão espontânea, f_c é a frequência da portadora óptica, h é a

constante de Planck ($\therefore hf_c$ é a energia do fóton), G é o ganho do amplificador, B_o é a faixa óptica (igual à faixa do filtro óptico em sistemas Wavelength Division Multiplexing (WDM)), B_e é a faixa elétrica do filtro passa-baixo do foto-detector. Seja $\gamma = Q^{-1}(BER)$ e a função Q pode ser numericamente calculada [39]:

$$Q(t) = \int_t^{\infty} \frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}} dx \quad (4.2)$$

O fator Q é comumente usado para especificar o desempenho do receptor, pois ele está relacionado a uma razão sinal-ruído que é necessária para se alcançar determinada taxa de erro de bits. Por exemplo, para uma BER de 10^{-12} , temos, aproximadamente, $\gamma = 7$. Atualmente, se considera uma BER de 10^{-12} para sistemas WDM novos.

4.1.3 Cascata de Amplificadores: O Pré-Amplificador Equivalente

Supondo agora que o receptor receba o sinal de uma linha equipada com uma cascata de amplificadores de linha, numerados de 1, 2, ... a partir do receptor, como é mostrado na Figura 4.1. O pré-amplificador pode ser considerado como o amplificador número 0 da cascata. Seja G_i o ganho do amplificador i e $N_{sp\ i}$ o seu fator de emissão espontânea. O trecho de linha entre o i -ésimo e o $(i - 1)$ -ésimo amplificador tem atenuação L_i . Seja ainda P_i a potência da marca na entrada do i -ésimo amplificador.

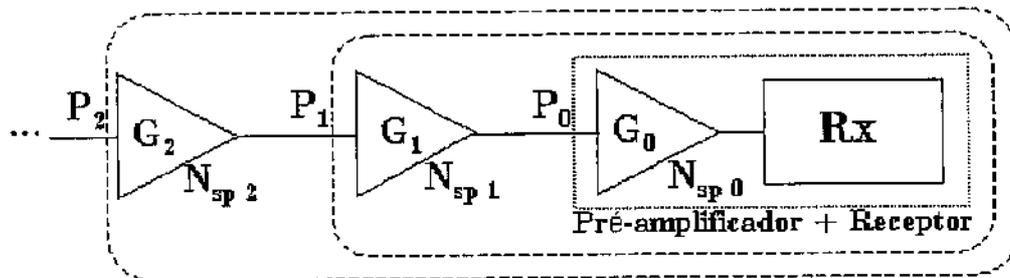


Figura 4.1: Cascata de amplificadores.

O valor mínimo de P_i pode ser obtido dobrando o valor da equação 4.1, isto é, $P_i = 2P_{sen}$. Ele é representado pela cascata, da entrada do amplificador i até a saída do pré-amplificador, como um amplificador equivalente de ganho G_i^{eq} e fator de emissão espontânea equivalente $N_{sp i}^{eq}$, que é calculado através da seguinte expressão [41, 38]:

$$N_{sp 1}^{eq} = \frac{N_{sp 1}(G_1 - 1)L_1G_0 + N_{sp 0}(G_0 - 1)}{G_1L_1G_0 - 1} \quad (4.3)$$

Finalmente, o valor de $N_{sp 1}^{eq}$ substitui N_{sp} na Equação 4.1 para calcular o valor mínimo de P_1 . Calculando-se recursivamente $N_{sp i}^{eq}$, é possível encontrar os níveis de sensibilidade em todos os componentes que são usados em uma conexão.

4.2 Arquitetura do Nó Óptico

Uma rede óptica consiste na interconexão de nós capazes de rotear comprimentos de onda – Wavelength Routing Node (WRN). Um par de fibras unidirecionais conectam WRNs na rede. Um WRN contém componentes tais como conectores, amplificadores, multiplexadores, etc. Além disso, as estações locais e o *cross-connect* são considerados parte do WRN. Ao longo deste Capítulo, o WRN também é referido como nó óptico ou simplesmente nó.

A Figura 4.2 mostra um WRN típico. Este WRN 4×4 tem uma única estação local representada pelos receptores e transmissores. O WRN também contém uma chave óptica – Optical Cross-Connect (OXC). O OXC utiliza multiplexadores e demultiplexadores e comutadores de comprimento de onda – Wavelength Routing Switch (WRS) para realizar todos os tipos de permutação de tráfego.

Neste exemplo, há 3 WRS no nó e cada WRS é dedicado a um comprimento de onda, ou seja, esse nó somente suporta 3 comprimentos de onda. O sinal entrando no WRN encontra vários componentes que contribuem ou para o ganho ou para a perda do sinal. A perda causada

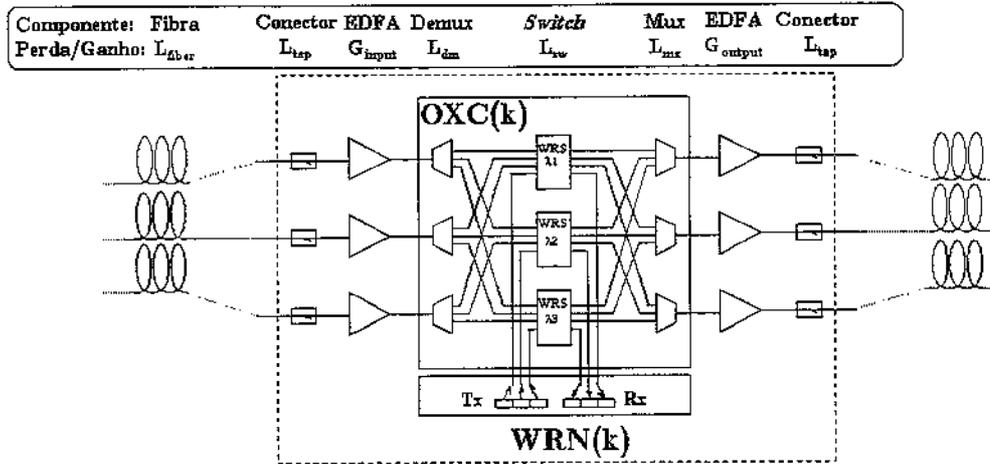


Figura 4.2: Arquitetura do WRN.

por um WRS $D_i \times D_i$ é dada pela seguinte fórmula [44]:

$$L_{sw} = 2\lceil \log_2(D_i) \rceil L_s + 4L_w, \tag{4.4}$$

onde D_i é o grau do nó, ou seja, o número de enlaces e estações que são ligadas a esse nó; L_s é a perda devida à inserção e L_w é a perda devida ao acoplamento. Tanto o OXC como o WRS são referidos simplesmente como *switch* neste trabalho.

4.2.1 Ganho do amplificador

O ganho no EDFA depende da potência total dos sinais ópticos de entrada. Para altas potências de entrada, o EDFA satura e seu ganho diminui. Dessa forma, os sistemas que usam o EDFA devem levar em conta a sua saturação.

Esse comportamento pode ser aproximado pela seguinte equação [45]:

$$G = 1 + \frac{P_{sat}}{P_{in}} \ln \frac{G_{max}}{G}, \tag{4.5}$$

onde G_{max} é o ganho não-saturado ou ganho de sinal pequeno, P^{sat} é a potência interna de saturação do amplificador e P_{in} é a potência total de entrada. A potência interna de saturação é uma função da potência de bombeio e de outros parâmetros do amplificador e é da ordem de 10 a 100 mW.

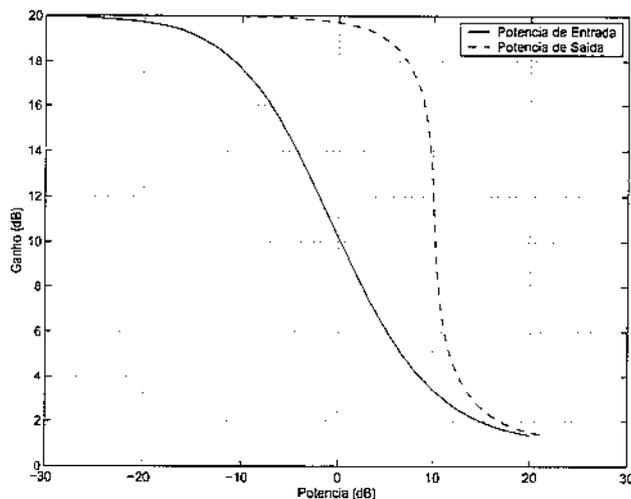


Figura 4.3: Curvas de saturação para um amplificador óptico de $G_{max} = 20$ dB e $P^{sat} = 13,7$ mW.

Para baixos valores de potência de entrada, o ganho tende a ser o ganho de sinal pequeno. Para altos valores de potência de entrada, o ganho tende a ficar igual a 1 e a potência de saída igual à potência de entrada (vide Figura 4.3).

4.3 Variações do Algoritmo Proposto

A métrica heurística local leva em conta o nível de potência de cada enlace conectado ao nó. Assim, ela é calculada da seguinte forma:

$$h_{nd} = \frac{P_{max} - P_n}{\sum_{n \in \mathcal{N}_i} (P_{max} - P_n)}, \quad (4.6)$$

onde P_{max} é o valor máximo da potência total no enlace (em mW) e P_n é o valor da potência que entra no enlace vizinho n (em mW).

4.4 Simulação

Consideramos um tráfego Poissoniano homogêneo com um perfil espacial uniforme. A duração de cada caminho óptico segue uma distribuição exponencial com o valor médio de 100 segundos. Há $W = 8$ comprimentos de onda disponíveis por enlace na rede, a qual está mostrada na Figura 4.4.

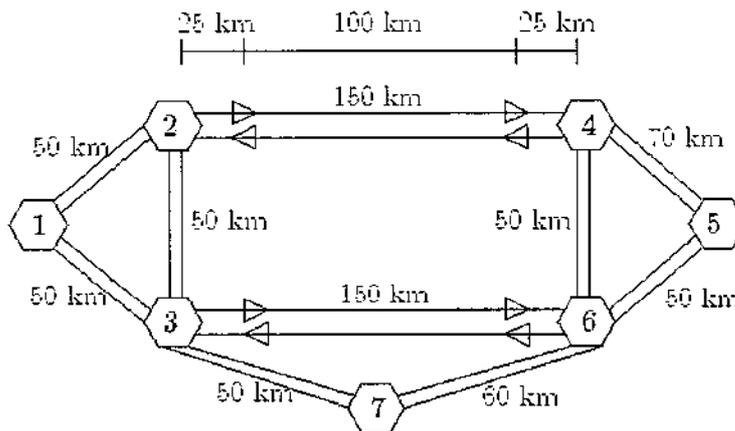


Figura 4.4: Rede de teste para os algoritmos de RWA-P.

Como o nível de ruído ASE vai acumulando ao longo dos enlaces, neste Capítulo, a distância usada em todos os algoritmos não é o número de *hops* e sim o comprimento do enlace em quilômetros.

As simulações foram executadas com 10^6 requisições para o roteamento fixo-alternado e 10^5 requisições para o roteamento baseado em ACO por ponto, respectivamente. Em todos os casos, os valores de probabilidade de bloqueio já haviam convergido.

Para o sub-problema de alocação de comprimento de onda, usamos uma estratégia *first-fit* para ambos os algoritmos, pois estamos comparando a eficiência do roteamento.

O mecanismo de estabelecimento de caminhos ópticos com o auxílio de um plano de controle GMPLS está detalhado no Apêndice B. Além disso, usa-se um sistema para controle de admissão das conexões – Call (or Connection) Admission Control (CAC) no nó destino. Após a definição da rota e do comprimento de onda do caminho óptico através da mensagem *Path*, o nó destino verifica se aquele caminho óptico não viola as restrições de potência (mínima e máxima) na rede. Os valores usados nesse cálculo estão apresentados na Tabela 4.1 e, uma vez que eles raramente mudam, podem ser configurados por gerenciamento ou podem usar algum tipo de OSPF-TE modificado para transportar informação de componentes da rede [46].

Um método iterativo é usado para se descobrir a potência de transmissão da nova conexão. Começa-se com -25 dBm de potência de transmissão e a cada iteração a potência é aumentada em 1 dB. As interações terminam quando todos os componentes estão acima do nível de sensibilidade e abaixo do nível máximo de potência agregada ou quando o laser atinge sua potência máxima de transmissão (P_{max}^{emit}).

Se a potência está acima do nível de sensibilidade em todos os componentes e as conexões já estabelecidas não violam as restrições de potência mínima e máxima, a conexão é aceita e uma mensagem *Resv* é enviada para o nó de ingresso (origem) para estabelecer a conexão. Caso contrário, a conexão é rejeitada e uma mensagem *PathErr* é gerada para indicar o bloqueio da requisição de conexão.

No caso do roteamento fixo-alternado, uma nova tentativa de conexão é feita para o caso de bloqueio lógico (violação da restrição de continuidade de comprimento de onda) ou rejeição no CAC, até que o número máximo de rotas alternativas é alcançado.

Na Tabela 4.1, são apresentados os parâmetros relativos ao problema RWA-P.

Os parâmetros usados nas simulações dos algoritmos baseados em ACO são mostrados na Tabela 4.2, os quais foram escolhidos após um grande número de testes.

Parâmetro	Símbolo	Valor
Máxima potência agregada no enlace	P_{max}	1 mW (0 dBm)
Potência máxima de transmissão	P_{max}^{pmit}	1 mW (0 dBm)
Ganho de pequeno sinal do amplificador de linha	G_{inline}	20 dB
Ganho de pequeno sinal do pré-amplificador	G_{input}	12 dB
Ganho de pequeno sinal do amplificador de potência	G_{output}	12 dB
Atenuação da fibra	L_{fiber}	0,2 db/km
Perda devida ao conector	L_{tap}	1 dB
Perda devida ao multiplexador	L_{mx}	4 dB
Perda devida ao demultiplexador	L_{dm}	4 dB
Perda devida à inserção do <i>switch</i>	L_s	1 dB
Perda devida ao acoplamento do <i>switch</i>	L_w	1 dB
Frequência da portadora óptica	f_c	193 THz (1,55 μ m)
Constante de Planck	h	6,63 $\times 10^{-34}$ J/Hz
Faixa óptica	B_o	100 GHz
Taxa de bits	B	10 Gbps
Faixa elétrica	B_e	8 GHz
Fator de emissão espontânea do amplificador de linha	N_{sp}^{inline}	2
Fator de emissão espontânea do pré-amplificador	N_{sp}^{input}	2
Fator de emissão espontânea do amplificador de potência	N_{sp}^{output}	2
Potência de saturação do amplificador de linha	P_{sat}^{inline}	13,7 mW
Potência de saturação do pré-amplificador de linha	P_{sat}^{input}	13,7 mW
Potência de saturação do amplificador de potência	P_{sat}^{output}	13,7 mW
Parâmetro da função de Q (probabilidade de erro)	γ	7

Tabela 4.1: Parâmetros e seus respectivos valores usados no RWA-P.

Tabela 4.2: Parâmetros relativos aos algoritmos baseados em ACO para o problema RWA-P.

Parâmetro	Símbolo	Valor
Taxa global de lançamento de formigas	R_{ants}	100 formigas/s
Intervalo para chegada das requisições	I_{warmup}	5000 s
Correção para o roteamento das formigas	α	0,6
Peso do modelo exponencial para a janela	η	0,005
Redutor para a janela do modelo paramétrico	c	0,3
Primeiro peso para o reforço adaptativo	c_1	0,6
Segundo peso para o reforço adaptativo	c_2	0,4
Nível de confiança para o reforço	γ	0,65
Amplificador da função de "amassamento"	a	5

4.5 Resultados

Comparamos os resultados do algoritmo proposto com esquemas de roteamento fixo e roteamento fixo-alternado, como pode ser visto na Figura 4.5.

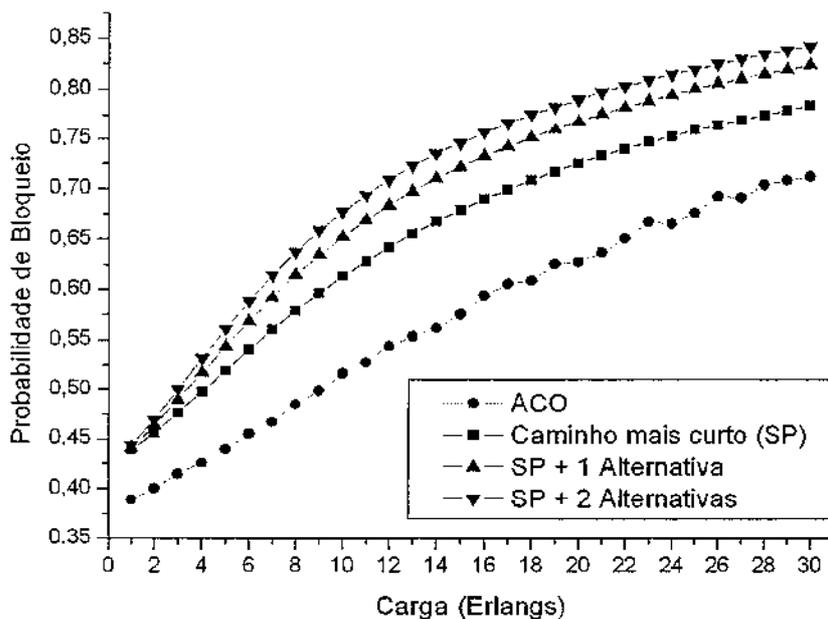


Figura 4.5: Roteamento fixo e fixo-alternado versus roteamento baseado em ACO.

A estratégia de se escolher rotas mais longas sem levar em consideração as degenerações na camada física não é tão boa como se poderia esperar, pois rotas mais longas significam mais ruído ASE acumulado, o que dificulta o estabelecimento de futuras conexões.

A natureza adaptativa do algoritmo baseado em ACO permite alcançar níveis mais baixos de probabilidade de bloqueio, uma vez que ele é capaz de selecionar as rotas baseando-se na potência óptica total que entra em cada enlace, isto é, ele é capaz de atingir um maior conhecimento da rede sem usar mais informação disseminada pelos protocolos de roteamento que o esquema fixo-alternado.

A Figura 4.6 mostra a percentagem de conexões estabelecidas sobre o total de requisições da rede. Note que a soma dessas percentagens sobre todos os comprimentos de rota é igual ao complemento da probabilidade de bloqueio mostrada na Figura 4.5 para uma carga de 10

erlangs.

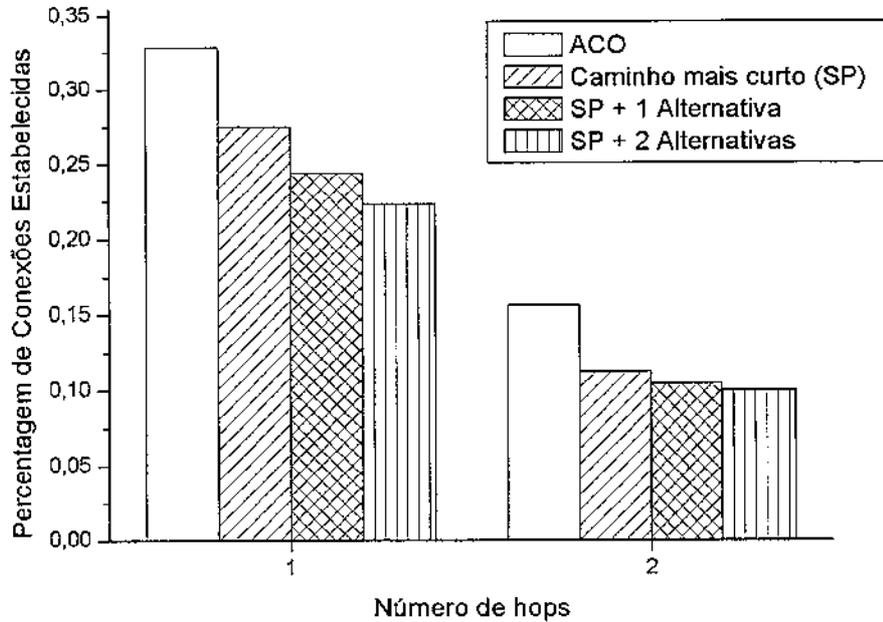


Figura 4.6: Percentagem das conexões estabelecidas por comprimento da rota com uma carga igual a 10 erlangs.

A rota mais longa que é passível de ser estabelecida possui apenas 2 *hops*. Essa limitação é devido ao fato de que as conexões que já foram estabelecidas não têm a sua potência de transmissão modificada quando novas requisições de conexão são feitas na rede. Em [41, 38], um algoritmo genético é usado para maximizar o número de conexões estabelecidas, através da otimização conjunta das potências de transmissão das conexões, atingindo, assim, níveis mais baixos de bloqueio. Além disso, um planejamento cuidadoso da rede é um pré-requisito muito importante para se alcançar níveis baixos de bloqueio.

Capítulo 5

Gerenciamento de Recursos em Grids

Grid pode ser definido como um sistema de processamento que utiliza recursos computacionais distribuídos, permitindo o acesso a processamento de forma confiável e consistente e economicamente atrativa.

O termo grid, que foi introduzido na metade dos anos 90 por Foster e Kesselman [47], foi inspirado em outros tipos de rede, como as redes de distribuição de energia elétrica, linhas de trem de ferro ou as redes telefônicas, onde uma certa infra-estrutura ou serviço é oferecido ao usuário após ser distribuído através de uma malha capilar.

O gerenciamento de recursos em grids é uma tarefa extremamente complicada, pois os recursos que devem ser compartilhados são muito heterogêneos: tempo de processamento, espaço de armazenamento / dados e banda de rede. Os sistemas atuais de grid focam no gerenciamento dos dois primeiros recursos, enquanto a rede é considerada como um recurso estático e pré-configurado. Mas as novas aplicações que começam a surgir no campo da *e-Science*, como as intensivas em dados, são enormemente beneficiadas por redes que podem ser configuradas dinamicamente para transferência de grandes quantidades de dados.

O advento das redes ópticas de longa distância foi fator decisivo para o desenvolvimento de grids computacionais em escalas geográficas intercontinentais, capacitando-os a transferir uma

quantidade massiva de dados entre usuários geograficamente dispersos, mas para se aproveitar todo o seu potencial é necessário que elas (i.e., as redes ópticas) sejam convenientemente tratadas como mais um recurso passível de ser gerenciado pelos sistemas de gerenciamento dos Grids, da mesma forma como hoje já ocorre com os recursos de processamento e de armazenamento.

Embora a rede óptica seja um recurso difícil de ser gerenciado e monitorado, as redes ópticas dos novos sistemas de grid estão sendo concebidas com um plano de controle GMPLS integrado, o qual provê uma maneira padronizada de gerenciar os recursos da rede e é uma grande fonte de informações que, contudo, foi comumente ignorada em estudos prévios ou arquiteturas de grid.

Neste Capítulo, empregaremos um sistema baseado em ACO como escalonador de sistemas de grid¹ construídos usando uma rede óptica com um plano de controle GMPLS, onde as formigas artificiais agem como representantes do usuário para a tomada de decisões de escalonamento.

Consideramos o poder de processamento e o provisionamento de caminhos ópticos, sem levar em conta o espaço de armazenamento e sem permitir a reserva de recursos, pois estamos mais interessados na avaliação do comportamento em regime do sistema. Uma vez que é necessário o provisionamento de caminhos ópticos sob demanda, o algoritmo de RWA deve ser integrado em uma plataforma com capacidade de descoberta de recursos.

Pela perspectiva do RWA, o ambiente de grid difere de uma rede tradicional roteada por comprimento de onda, pois não tem como objetivo a conexão entre um par de nós origem-destino e sim entre uma origem e um recurso presente em outro nó do grid. Assim, a disponibilidade do recurso deve ser verificada e a conexão via caminho óptico estabelecida. Os algoritmos baseados em formigas combinam naturalmente os processos de descoberta e roteamento: quando um recurso é descoberto, os níveis de ferormônio produzidos na atividade de busca são uma métrica apropriada para um bom roteamento.

¹Escalonador (ou broker) do grid é a entidade responsável para a alocação de tarefas a recursos entre diferentes domínios administrativos [48].

5.1 Variações do Algoritmo Proposto

1. Vetor de disponibilidade A^i : É um vetor que contém uma métrica de disponibilidade para cada destino d da rede. Neste trabalho, assumiu-se que todos os nós têm o mesmo número de processadores, de forma que empregamos a seguinte expressão para calcular a disponibilidade: $A_d = P_d^{idle} / P_{proc}$, onde P_d^{idle} é o número dos processadores que não estão em uso no destino d e P_{proc} é o número total de processadores instalados em cada nó. A idéia por trás desta estrutura é permitir o balanceamento de carga das fazendas de servidores.

Um grid é formado por nós, que podem ser usuários e/ou fornecedores de recursos. O algoritmo tentará estabelecer um caminho óptico para conectar um nó usuário que procura por um nó fornecedor em algum lugar da rede, de forma a reduzir a probabilidade de bloqueio da rede óptica enquanto se balanceia a carga entre as fazendas de servidores. Assim, os nós usuários serão os nós origem e os nós fornecedores os nós destino do algoritmo baseado em ACO.

O vetor de disponibilidade de cada nó é inicializado com todos os valores iguais a 1, assumindo que todos os processadores nas fazendas de servidores estão sem uso.

A métrica heurística local leva em conta o complemento do congestionamento de cada enlace vizinho. Assim, ela é calculada da seguinte forma:

$$h_{nd} = \frac{l_n^a}{W}, \quad (5.1)$$

onde l_n^a é o número de comprimentos de onda disponíveis no vizinho n e W é o número total de comprimentos de onda implementados no enlace.

5.2 Arquitetura

O escalonamento em grids é composto de três partes [48]: descoberta de recurso, seleção do sistema e execução da tarefa.

A descoberta de recursos é feita pelas formigas artificiais que circulam na rede. Elas podem utilizar a base de dados enlace-estado do protocolo OSPF-TE para a descoberta de todos os nós da rede. Se esse banco de dados não está disponível, a descoberta dos nós é feita pelas próprias formigas. A informação de disponibilidade é coletada pelas formigas quando chegam ao nó de destino, que posteriormente atualizam o vetor de disponibilidade do seu respectivo nó origem.

Quando uma tarefa deve ser executada, o nó usuário s inspeciona o vetor de disponibilidade A^s e procura pelo nó fornecedor com o maior valor de disponibilidade para ser o destino da requisição do caminho óptico. Se a disponibilidade de todos os nós fornecedores é igual a zero, então a tarefa é bloqueada, pois não é possível fazer o agendamento de recursos.

Quando o caminho óptico é estabelecido, os dados relacionados à tarefa a ser executada são transferidos ao nó fornecedor para serem processados. Entretanto, se o nó de destino estiver completamente ocupado ou se a requisição de conexão for bloqueada por insuficiência de recursos na rede, a submissão da tarefa é bloqueada.

Finalmente, se os dados relativos à tarefa forem transferidos com sucesso, o LSP é finalizado e a tarefa é executada. Consideramos que não há retorno dos resultados ao nó usuário pela rede óptica, pois o conteúdo do resultado é muito menor que os dados processados, ou que a transferência do resultados ao nó usuário é feita sem bloqueio através de algum esquema de reserva especial.

Neste trabalho, assumiu-se apenas um nó usuário no grid. Contudo, a adição de mais usuários que compartilhem o grid pode ser facilmente alcançada, pois cada usuário pode possuir um diferente tipo de colônia de formigas, de tal maneira que cada colônia possui apenas uma vista parcial dos recursos no grid.

Além disso, o algoritmo baseado em ACO é escalável, pois pode ser implementado, por

exemplo, de uma forma hierárquica. Nesse esquema multi-camadas, o algoritmo é usado de forma independente em cada nível. A Figura 5.1 ilustra um exemplo de um sistema de grid de 2 camadas, onde o primeiro nível pode ser uma rede nacional e o segundo nível uma rede regional. A camada superior consegue enxergar os recursos disponíveis na camada inferior, mas o escalonamento da camada inferior é feita por ela mesmo.

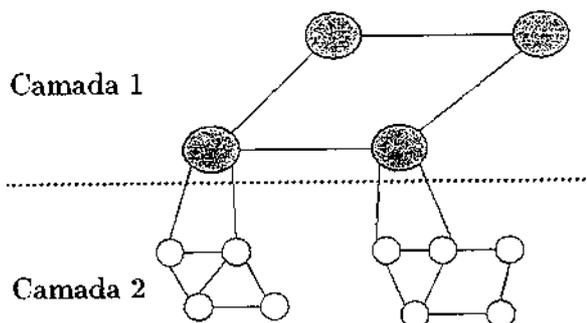


Figura 5.1: Um sistema de grid gerenciado em duas camadas.

5.3 Simulação

Consideramos um tráfego Poissoniano homogêneo com um perfil espacial uniforme. Somente a rede NSFNet foi simulada neste Capítulo, com 10^6 requisições para cada simulação. O nó “1” foi escolhido como o gerador de todas as requisições de caminho ópticos, ou seja, ele é o único nó usuário da rede. Os outros nós são os destinos e cada um está equipado com P_{proc} processadores, ou seja, eles são os nós fornecedores do grid.

Consideramos que a duração do caminho óptico tem um valor fixo de 14,4 s, o que é igual a uma transferência de um arquivo de 1,8 GB a uma taxa de 10 Gbps, sendo que cada tarefa requer 12 horas de processamento.

A rede possui W comprimentos de onda disponíveis por enlace e foram avaliados os seguintes valores de W : 2, 4 e 8. Para o sub-problema de alocação de comprimento de onda, usamos uma estratégia *first-fit*.

O mecanismo de estabelecimento de caminhos ópticos com o auxílio de um plano de controle GMPLS está detalhado no Apêndice B, com duas ressalvas: os LSPs são estabelecidos de forma bidirecional e há um novo código de erro caso a fazenda de servidores esteja completamente ocupada quando a mensagem *Path* é recebida pelo nó fornecedor.

Os parâmetros usados nas simulações, que são parcialmente baseados na Tabela 3.1, estão mostrados na Tabela 5.1:

Tabela 5.1: Parâmetros usados na simulação do gerenciamento de recursos no grid.

Parâmetro	Símbolo	Valor
Taxa de transmissão de dados	b	10 Gbps
Número total de nós por nó	P_{proc}	1000
Duração do caminho óptico	t_λ	14,4 s
Taxa global de lançamento de formigas	R_{ants}	26 formigas/s
Intervalo para chegada das requisições	I_{warmup}	1000 s
Correção para o roteamento das formigas	α	0.6
Peso do modelo exponencial para a janela	η	0.005
Redutor para a janela do modelo paramétrico	c	0.3
Primeiro peso para o reforço adaptativo	c_1	0.6
Segundo peso para o reforço adaptativo	c_2	0.4
Nível de confiança para o reforço	γ	0.65
Amplificador da função de “amassamento”	a	5

5.4 Resultados

Primeiramente, avaliamos o algoritmo baseado em ACO sob diferentes cargas para verificar sua eficiência. Os resultados para diferentes números de comprimento de onda estão mostrados na Figura 5.2. O bloqueio pode ser causado por insuficiência de recursos da rede ou por falta de processadores sem uso. Nesta figura, podemos observar que quando temos $W = 2$, a insuficiência de comprimentos de onda é responsável pelo bloqueio. Por outro lado, para $W = 4$ ou 8, o bloqueio é dominado pela falta de processadores disponíveis nas cargas mais altas (≥ 5 erlangs), enquanto nas cargas mais baixas, o bloqueio está mais ligado a falta de recursos de rede.

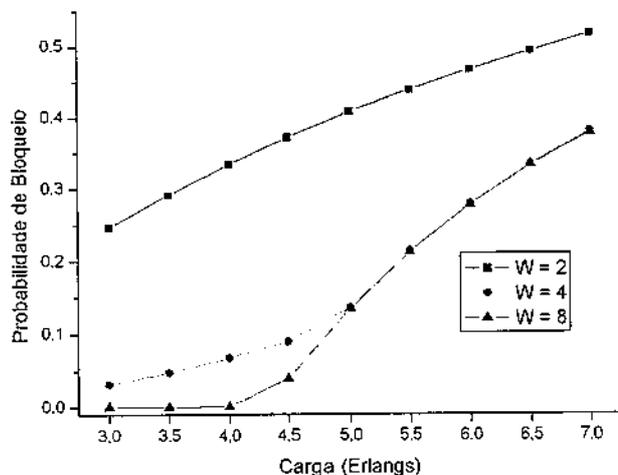


Figura 5.2: Probabilidade de bloqueio no grid para diferentes cargas.

Olhando a Figura 5.3, podemos notar que as requisições para submeter uma tarefa são bloqueadas em todas as cargas mostradas devido somente à falta de recursos na rede óptica.

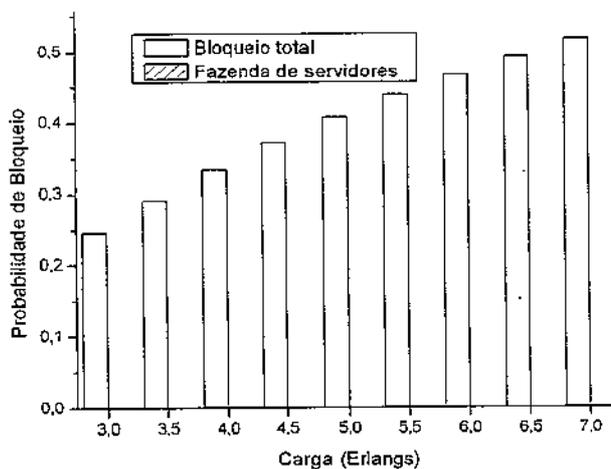


Figura 5.3: Probabilidade de bloqueio total e relativa somente à fazenda de servidores para $W = 2$ comprimentos de onda.

A Figura 5.4, por outro lado, mostra que o fator limitante são os recursos da rede óptica para cargas mais baixas. Contudo, para cargas mais altas, o número de processadores também é um gargalo do sistema.

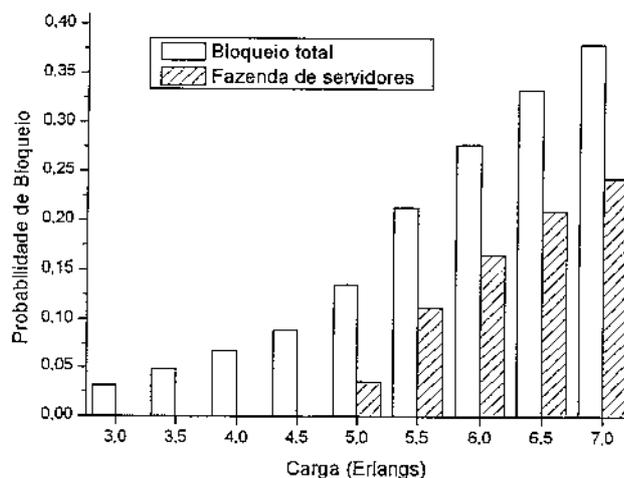


Figura 5.4: Probabilidade de bloqueio total e relativa somente à fazenda de servidores para $W = 4$ comprimentos de onda.

O caso oposto que foi observado na Figura 5.3 é mostrado na Figura 5.5. Neste caso, o gargalo é principalmente dado pelo poder de processamento dos nós fornecedores. Somente uma pequena parcela do bloqueio é devido à falta de recursos de rede.

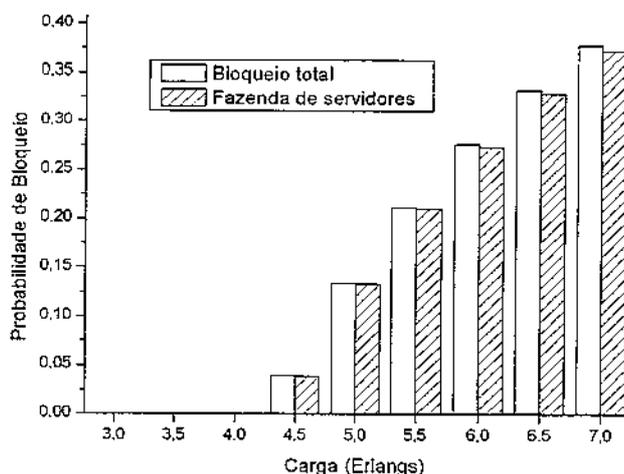


Figura 5.5: Probabilidade de bloqueio total e relativa somente à fazenda de servidores para $W = 8$ comprimentos de onda.

A Figura 5.6 ilustra a carga de trabalho no grid em função do tempo para uma carga fixa oferecida de 4 erlangs na rede óptica. Como já notado nas figuras anteriores, pode-se verificar

que com $W = 2$ ou 4, o grid não alcança a sua máxima utilização por falta de recursos da rede óptica.

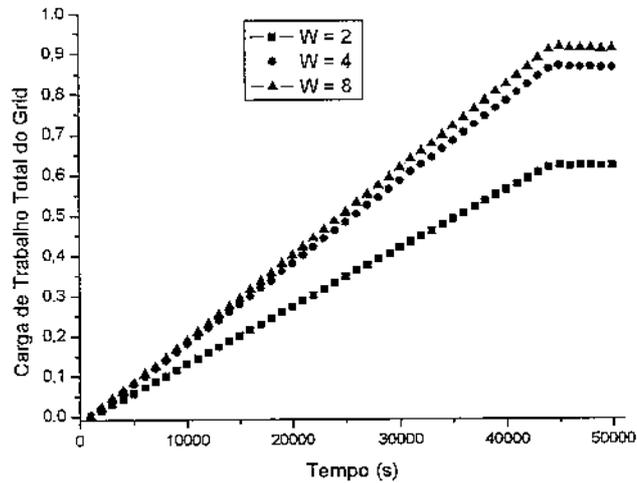


Figura 5.6: Carga de trabalho total do grid para uma carga de 4 erlangs na rede óptica.

Também foi avaliada a carga de trabalho total do grid sob diferentes cargas na rede óptica para $W = 8$, sendo que os resultados estão mostrados na Figura 5.7. A partir de uma carga de 4,5 erlangs, as fazendas de servidores saturam e não conseguem mais processar todas as submissões de tarefas devido à falta de processadores disponíveis.

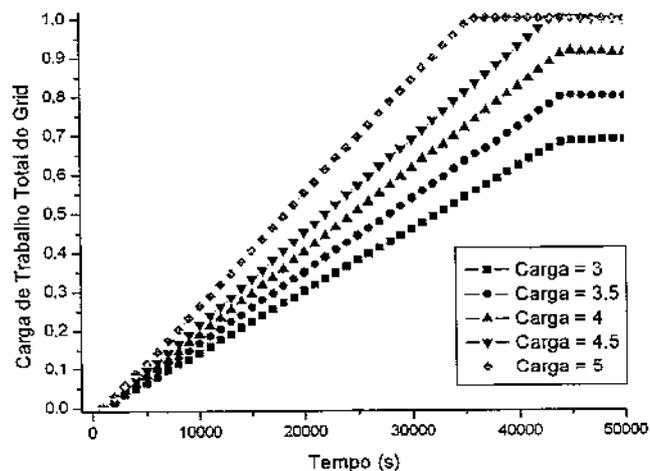


Figura 5.7: Carga de trabalho total do grid para $W = 8$ comprimentos de onda.

Além disso, foi verificada a influência do número de processadores na probabilidade de bloqueio, cujos resultados estão mostrados na Figura 5.8. Pode-se perceber os pontos onde a rede óptica começa a limitar o desempenho em termos de probabilidade de bloqueio. Para $W = 2$, esse limite é $P_{proc} > 600$, para $W = 4$, o limite é $P_{proc} > 800$ e, para $W = 8$, ele é $P_{proc} > 900$, para uma carga oferecida na rede óptica de 4 erlangs. Isto pode ser confirmado quando se olha a Figura 5.6. Nessa figura, a carga de trabalho do grid alcança o valor médio de $> 60\%$, $> 80\%$ e $> 90\%$ para $W = 2$, 4 e 8, respectivamente, para essa carga de 4 erlangs e $P_{proc} = 1000$.

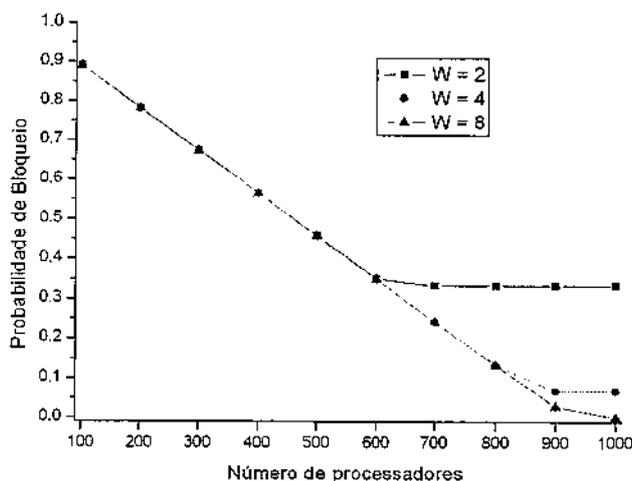


Figura 5.8: Probabilidade de bloqueio em função do número de processadores com um carga de 4 erlangs na rede óptica.

O desempenho mais detalhado para $W = 2$, quando o número de processadores implementados é variado, é mostrado na Figura 5.9. Embora o bloqueio seja causado pela insuficiência de recursos de rede, a probabilidade de bloqueio resultante não é afetada somente por esse fato quando o número de processadores é insuficiente, como podemos ver na Figura 5.8. Como discutido anteriormente, o bloqueio estabiliza quando o número de processadores disponíveis por nó é maior que 600, de forma que o bloqueio é dominado pela falta de recursos de rede.

Observando a Figura 5.10, pode-se perceber que o bloqueio devido à insuficiência de recur-

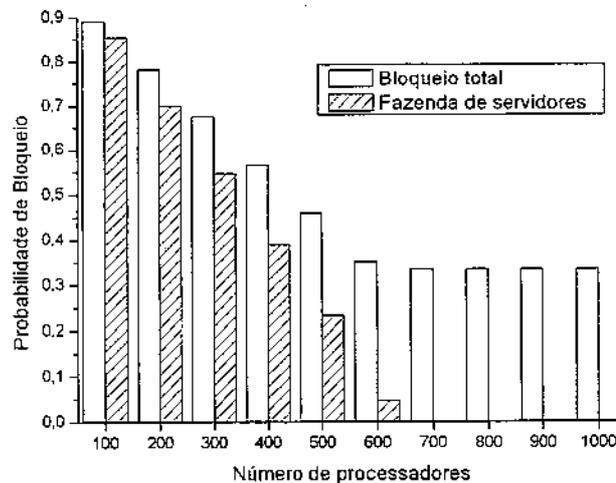


Figura 5.9: Probabilidade de bloqueio total e relativa somente à fazenda de servidores em função do número de processadores para $W = 2$ e carga na rede óptica igual a 4 erlangs.

os da rede óptica não afeta o bloqueio total do sistema, quando o número de processadores é pequeno, como mostrado na Figura 5.9. Neste caso, todavia, o desempenho somente é limitado pela rede quando P_{proc} é maior que 800.

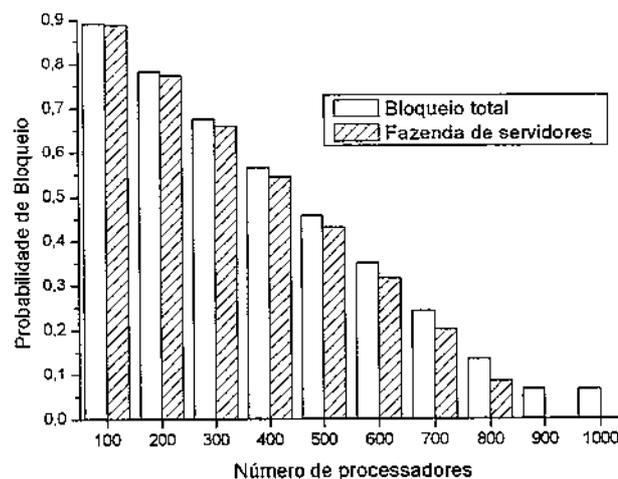


Figura 5.10: Probabilidade de bloqueio total e relativa somente à fazenda de servidores em função do número de processadores para $W = 4$ e carga na rede óptica igual a 4 erlangs.

Por outro lado, quando se usa $W = 8$, o desempenho é principalmente limitado pela

falta de processadores nas fazendas de servidores. O bloqueio é praticamente nulo quando 1000 processadores estão disponíveis por nó, como pode ser visto na Figura 5.11. A eficiência do roteamento do algoritmo baseado em ACO pode otimizar o uso dos recursos da rede quando há poder de processamento suficiente nas fazendas de servidores do grid.

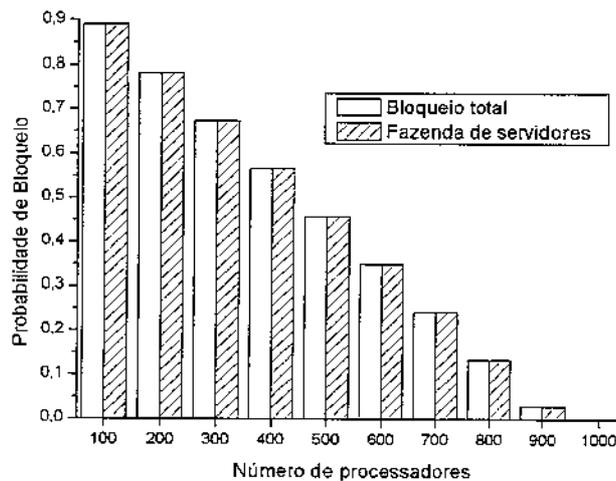


Figura 5.11: Probabilidade de bloqueio total e relativa somente à fazenda de servidores em função do número de processadores para $W = 8$ e carga na rede óptica igual a 4 erlangs.

Outro aspecto importante a ser avaliado é a influência da taxa global de lançamento de formigas na probabilidade de bloqueio para uma carga na rede óptica de 4 erlangs. Foram usados múltiplos de 13, que é o número de nós fornecedores do grid. É um parâmetro importante do algoritmo e está ligado à duração do caminho óptico. Um valor muito pequeno para essa taxa pode comprometer a eficiência do algoritmo.

Os resultados para os diferentes comprimentos de onda estão mostrados na Figura 5.12. O aumento da taxa de lançamento de formigas melhora o desempenho do sistema através da redução da probabilidade de bloqueio. Essa melhoria tende a estabilizar após um certo valor, o qual deve ser mantido dentro de uma taxa razoável a fim de evitar o congestionamento dos canais de controle GMPLS.

O *overhead* causado pelas formigas é extremamente dependente da implementação das

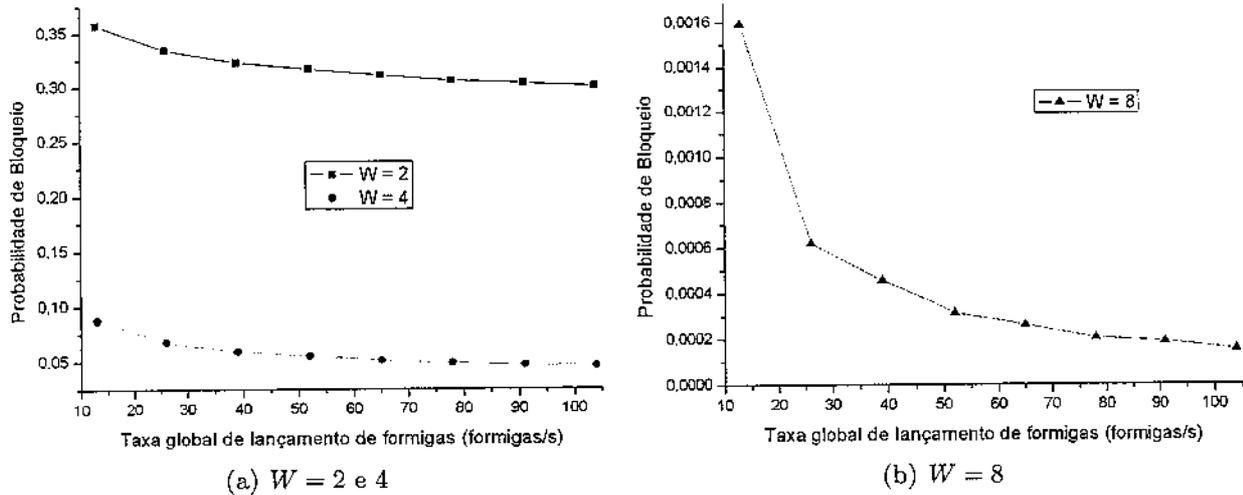


Figura 5.12: Probabilidade de bloqueio em função da taxa global de lançamento de formigas para uma carga na rede óptica de 4 erlangs.

formigas artificiais. Suponha que a formiga é implementada como uma datagrama em uma rede que utiliza Internet Protocol version 6 (IPv6) [49]. No caso mais simples, temos um cabeçalho com 40 bytes e cada *hop* contribui com 16 bytes para o tamanho da formiga. Além disso, a formiga que faz o caminho de volta para o nó origem, carrega um campo extra de 8 bytes que é a informação de disponibilidade coletada.

A Figura 5.13 mostra o *overhead* médio causado pelas formigas nos canais de controle da rede em função da taxa global de lançamento, onde as barras indicam o erro padrão da média. De fato, o *overhead* introduzido pelas formigas é muito pequeno quando comparado com as taxas de transmissão dos canais de controle reais, os quais possuem vários Mbit/s de banda disponível. Esse *overhead* escala linearmente com a taxa de lançamento de formigas.

Como cada formiga introduz pequenas melhorias nas tabelas de roteamento, a taxa global de lançamento de formigas deve considerar a dinâmica do sistema. Em nosso caso, esse parâmetro deve levar em conta a duração dos caminhos ópticos e a duração do processamento das tarefas. É muito difícil definir uma expressão fechada para calcular esse parâmetro, de forma que usamos um método de tentativa e erro para descobrir um valor apropriado. Entretanto, com uma

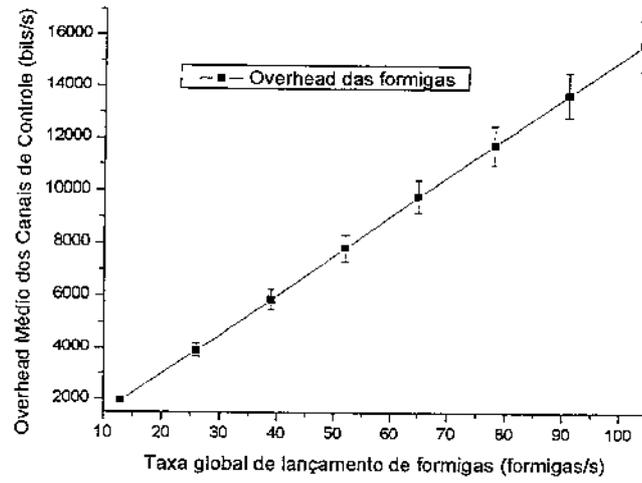


Figura 5.13: Overhead das formigas artificiais $W = 8$ comprimentos de onda e carga na rede óptica de 4 erlangs.

implementação cuidadosa das formigas, é pouco provável que os canais de controle possam ficar congestionados quando uma taxa maior de lançamento de formigas for necessária em um cenário mais dinâmico.

Capítulo 6

Redes de Pacotes Ópticos

As técnicas de comutação por pacotes ópticos – Optical Packet Switching (OPS) prometem aumentar o desempenho em relação as redes tradicionais comutadas por circuitos [50]. Entretanto, devido à falta de uma memória óptica real em redes do tipo OPS, é necessário uma forma de gerenciamento que evite uma grande ocorrência de perda de pacotes.

Este Capítulo trata de redes assíncronas e com tamanho de pacote variável, com capacidade completa de conversão de comprimento de onda em cada nó [51]. Cada nó está composto de um número arbitrário de fibras de entrada e de saída, onde cada fibra é capaz de transportar W comprimentos de onda cada uma.

Cada nó da rede é equipado com *buffers* ópticos construídos usando linhas de atraso – Fiber Delay Line (FDL), que são rolos de fibras ópticas usados para atrasar os pacotes. Como a velocidade da luz é em torno de 200.000 km/s em fibras, um quilômetro de fibra introduz um atraso de $5 \mu s$.

A principal função dos *buffers* ópticos é gerenciar o congestionamento no domínio do tempo. Consideramos o caso degenerado, onde o buffer é confeccionado usando B FDLs, sendo que cada FDL é um múltiplo consecutivo da unidade de atraso D . A primeira FDL introduz um atraso de 0, a segunda um atraso de D e assim por diante, até a última B -ésima FDL, a qual introduz

um atraso de $(B - 1)D$.

Pacotes de diferentes comprimentos de onda podem compartilhar a mesma FDL e todos os comprimentos de onda são logicamente iguais. Assim, cada comprimento de onda pode ser visto como uma fila para agendar os pacotes.

Na Figura 6.1, é mostrado um exemplo de um nó OPS com duas fibras de entrada e duas de saída, onde cada fibra suporta 4 diferentes comprimentos de onda, uma matriz de comutação não-bloqueante e um *buffer* confeccionado com 4 FDLs. As unidades de entrada são compostas por um de-multiplexador, conversores plenos de comprimentos de onda e um processador de cabeçalho. As unidades de saída são compostas por um multiplexador e conversores plenos de comprimento de onda.

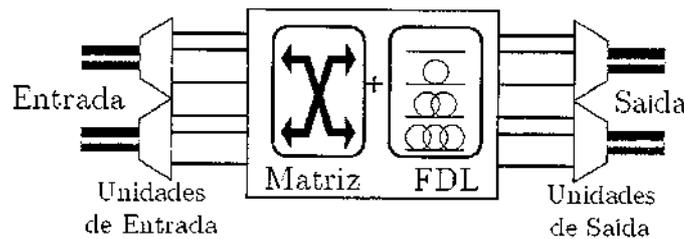


Figura 6.1: Exemplo de um nó OPS com $W = 4$ e $B = 4$.

6.1 Resolução de Contenção em Nós OPS

O problema da escolha de um comprimento de onda para transmitir um pacote óptico e, caso o comprimento de onda escolhido esteja ocupado, também do atraso que o pacote vai sofrer no *buffer*, é chamado de problema de seleção do comprimento de onda e do atraso – Wavelength and Delay Selection (WDS).

Algumas heurísticas foram propostas para tratar esse problema, como a “Menor lacuna” (MinGAP) e a “Menor tamanho” (MinLEN) [51]. A estratégia MinGAP tenta achar a menor lacuna para inserir o pacote. Se há empate entre diferentes comprimentos de onda, então o es-

quema MinLEN é usado nesses comprimentos de onda. No caso da estratégia MinLEN, tenta-se descobrir o comprimento de onda com menor atraso. Se há empate, então o esquema MinGAP é usado nos comprimentos de onda selecionados pelo heurística de MinLEN. Se todos os comprimentos de onda estão ocupados e não há alguma FDL disponível, então o pacote é descartado. No exemplo mostrado na Figura 6.2, se MinGAP é usado, então w_3 é escolhido e se MinLEN é usado, então w_4 é escolhido.

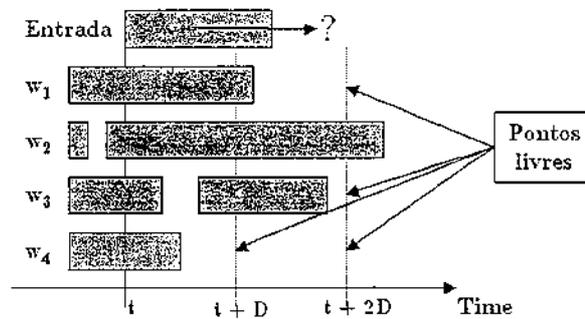


Figura 6.2: Algoritmo de WDS.

Ambas heurísticas foram simuladas usando tráfego poissoniano e auto-similar, conforme descrito nas Sub-seções 6.3.1 e 6.3.2, respectivamente. Os resultados para o primeiro tipo de tráfego pode ser visto na Figura 6.3.

Observando-se a Figura 6.3(a), pode-se verificar que o melhor algoritmo de WDS é o MinGAP e o melhor valor para a unidade de atraso D é cerca de $3,2 \mu s$, que é aproximadamente 80% do tamanho de pacote usado.

Usando-se $D = 3,2 \mu s$, simulou-se o *buffer* sobre diferentes cargas para se verificar o comportamento da sua ocupação média e a probabilidade média de perda de pacotes. Examinando-se os resultados mostrados na Figura 6.4, pode-se notar que a ocupação média do *buffer* varia não-linearmente com a probabilidade média de perda.

O mesmo processo é repetido para a Figura 6.3(b). Também nesse caso, a estratégia MinGAP é a vencedora, mas o melhor valor de unidade de atraso está em torno de 40% do

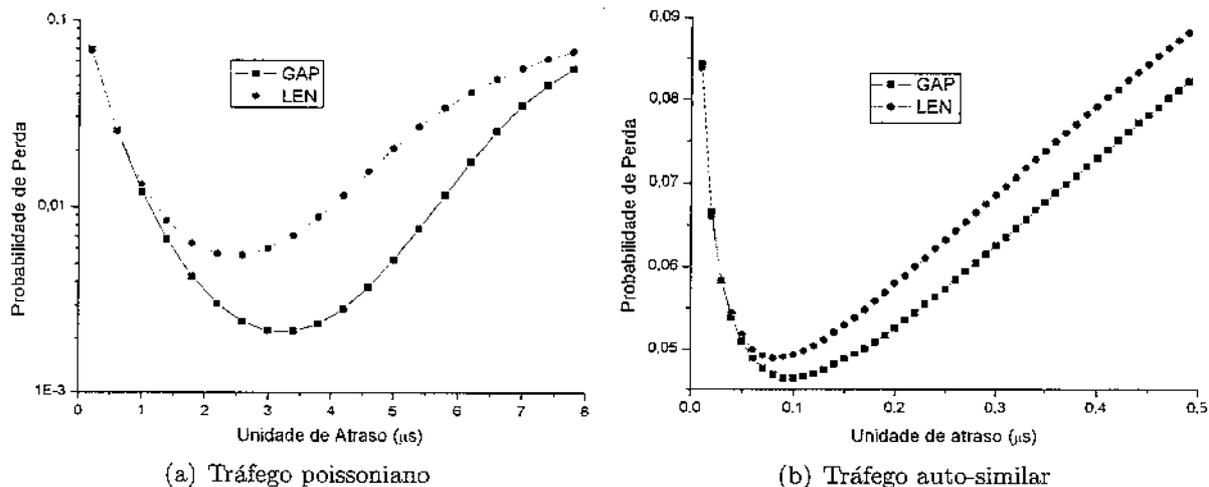


Figura 6.3: Probabilidade de perda versus unidade de atraso para carga = 6.4 erlangs.

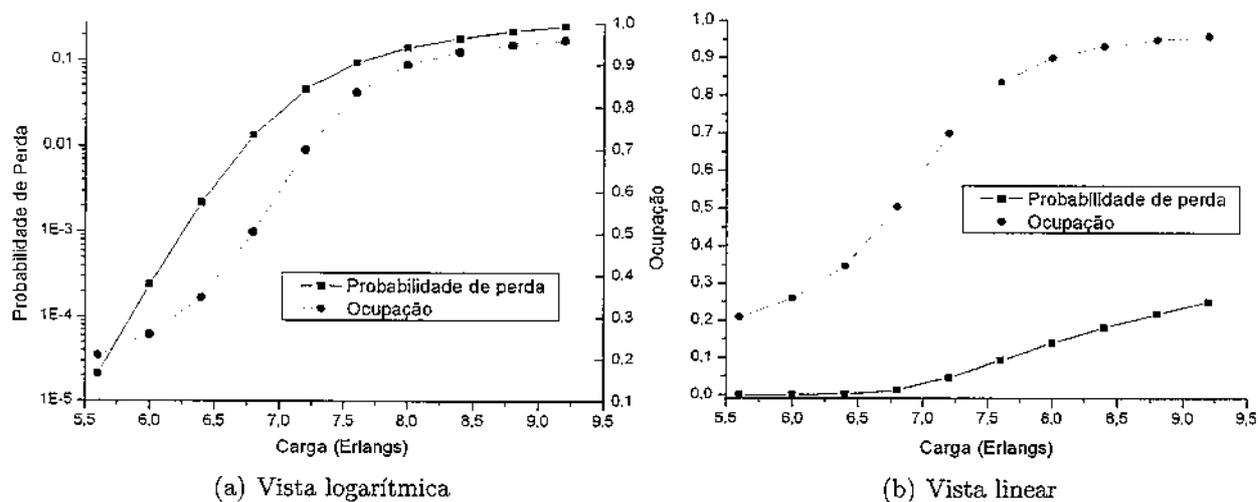


Figura 6.4: Influência da ocupação média do buffer sobre a probabilidade de perda para tráfego poissoniano.

tamanho médio do pacote, o que é equivalente a uma unidade de atraso $D = 0,1 \mu s$. Assim, para esse valor de D , a Figura 6.5 mostra o comportamento da ocupação média do *buffer* em relação a probabilidade média de perda de pacotes, quando se varia a carga. Assim como para o tráfego poissoniano, a ocupação média do *buffer* varia não-linearmente com a probabilidade média de perda.

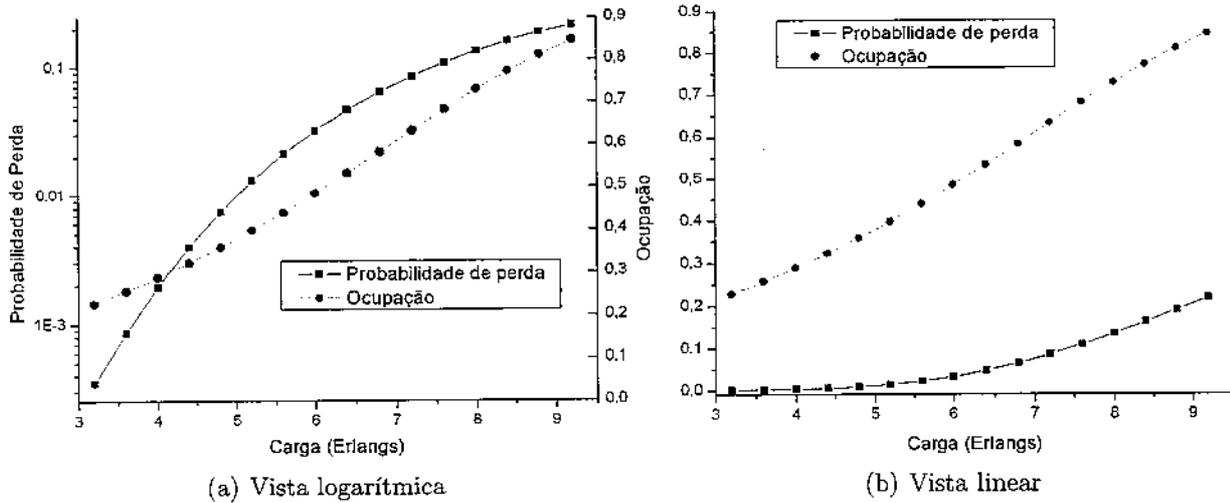


Figura 6.5: Influência da ocupação média do buffer sobre a probabilidade de perda para tráfego auto-similar.

6.2 Variações do Algoritmo Proposto

Para o roteamento em redes do tipo OPS é necessária uma estrutura de dados extra para cada nó:

1. Tabela de roteamento de dados \mathcal{R}^i : É uma tabela contendo uma linha para cada destino d da rede e uma coluna para cada nó vizinho com o propósito de rotear os pacotes de dados na rede. Possui a mesma estrutura e propriedades de T^i .

A métrica heurística local leva em conta o estado de ocupação do *buffer* óptico. Para se calcular a ocupação do *buffer* óptico foi usada a seguinte fórmula:

$$\Omega_{nd} = \frac{\sum_{w \in W} (B - N_{free})}{W B}, \quad (6.1)$$

onde N_{free} é o número de “pontos livres” como mostrado na Figura 6.2.

Uma vez que a probabilidade de perda de pacotes varia não-linearmente com o valor da ocupação do *buffer*, como pode ser visto nas Figuras 6.4 e 6.5, empregamos a seguinte função de

decaimento exponencial para computar a métrica heurística local do roteamento:

$$h_{nd} = e^{-\lambda \Omega_{nd}} \quad (6.2)$$

O efeito do parâmetro λ é ilustrado na Figura 6.6:

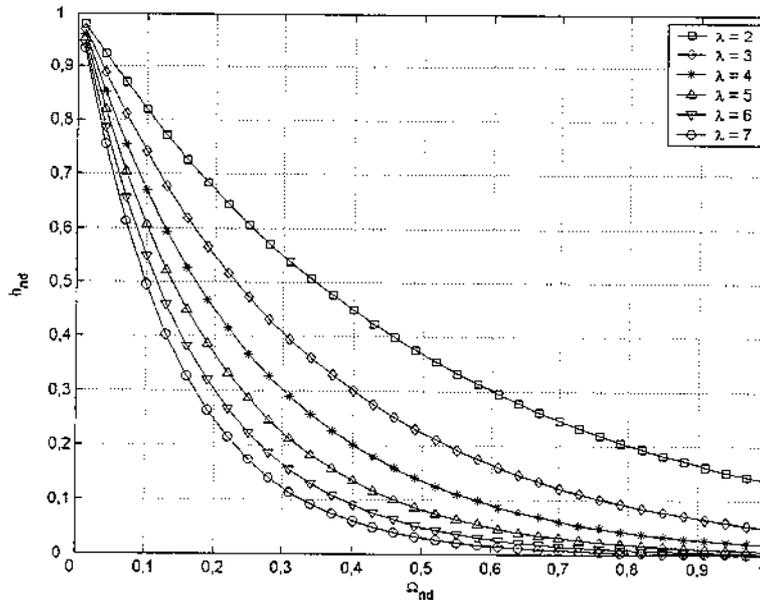


Figura 6.6: A transformação da função de decaimento exponencial.

Dessa forma, os pacotes de dados são roteados *hop a hop* de forma probabilística, de acordo com os valores de \mathcal{R}^i , dado o destino d do pacote. Para se evitar que os pacotes cheguem no destino fora de ordem, pacotes com o mesmo nós origem e destino ou com outra tipo de informação, tais como o tipo do protocolo da carga útil ou a cor do DiffServ, devem ser agregados e seguir a mesma rota.

Quando as formigas fazem a atualização referente à linha k da tabela de roteamento por ferormônio \mathcal{T}^i , a respectiva linha da tabela de roteamento de dados \mathcal{R}^i também é atualizada, segundo a seguinte expressão:

$$\mathcal{R}_{nk}^i = (\tau_{nk})^\varepsilon, \quad (6.3)$$

onde ε é o fator da transformação exponencial.

Essa transformação é seguida da normalização dos valores, de tal modo que a soma de todos os vizinhos seja igual a 1. Essa transformação favorece as rotas que tem um maior nível de ferormônio associado.

Por ocasião de uma falha, após a atualização da tabela de roteamento por ferormônio T^i , a tabela de roteamento de dados \mathcal{R}^i também é atualizada. Como em T^i , a linha relativa ao vizinho m também é removida.

Após essa remoção, se aplica a transformação exponencial em \mathcal{R}^i usando-se a Equação 6.3, seguido da normalização das linhas restantes, como descrito anteriormente.

6.3 Simulação

Ao contrário dos Capítulos precedentes, as redes do tipo OPS foram simuladas com dois tipos de tráfego: tráfego Poissoniano com duração dos pacotes seguindo uma distribuição exponencial e tráfego auto-similar. Assim, os parâmetros para cada tipo diferente de tráfego estão apresentados nas respectivas Sub-Seções.

6.3.1 Tráfego Poissoniano

Para a rede NSFNet, os enlaces têm todos os mesmos comprimentos e foram simulados com 4 valores diferentes: 0, 100, 200 e 400 km. A rede NTTNet também foi simulada com diferentes tamanhos de enlace, com valores iguais a 0 e 80 km.

Foi usado um tráfego poissoniano homogêneo e o destino do pacote é escolhido seguindo-se uma distribuição uniforme. O comprimento de cada pacote segue uma distribuição exponencial com o valor médio igual a $\overline{L_p}$ e um valor mínimo igual a L_{min} . Escolhemos o valor de L_{min} igual a 40 bytes, que é o mínimo tamanho de pacote em redes IPv6 [49].

Cada ponto é a média de 10 diferentes simulações. Cada simulação é executada com P_{run}

pacotes e uma semente aleatória diferente.

Para a simulação da parte relativa a restauração, se considerou 4 diferentes tempos para cada evento de falha (falha simples de nó ou enlace). Neste caso, cada simulação foi executada com 10^6 pacotes. Na rede NSFNet foram usados enlaces de 200 km e na rede NTTNet enlaces de 80 km, para se verificar a eficiência da disseminação da informação de falha entre os nós.

Após uma série de testes, os parâmetros usados nas simulações com tráfego poissoniano estão mostrados na Tabela 6.1:

Tabela 6.1: Parâmetros usados na simulação de redes OPS com tráfego poissoniano.

Parâmetro	Símbolo	Valor
Taxa de transmissão de dados	b	10 Gbps
Número de comprimentos de onda por enlace	W	8
Número de pacotes para cada ponto	P_{run}	10^8
Taxa global de lançamento de formigas ^a	R_{ants}	10^5 formigas/s
Intervalo para chegada dos pacotes ^a	I_{warmup}	100 ms
Tempo para detectar e localizar uma falha	I_{detect}	10 ms
Tamanho médio de pacote	\bar{L}_p	5000 bytes
Tamanho mínimo de pacote	L_{min}	40 bytes
Número de bytes adicionados a cada hop	b_H	6 bytes
Número de FDLs no <i>buffer</i>	B	8
Unidade de atraso do <i>buffer</i>	D	3,2 μ s
Correção para o roteamento das formigas	α	0,5
Correção para a métrica de congestionamento do <i>buffer</i>	λ	7
Peso do modelo exponencial para a janela	η	0,005
Redutor para a janela do modelo paramétrico	c	0,3
Fator exponencial para a tabela de roteamento de dados	ϵ	1,4
Primeiro peso para o reforço adaptativo	c_1	0,7
Segundo peso para o reforço adaptativo	c_2	0,3
Nível de confiança para o reforço	γ	0,65
Amplificador da função de "amassamento"	a	5

^a Para NTTNet, este valor é dobrado.

Foi estudada a influência de alguns parâmetros mostrados na Tabela 6.1. De fato, não existe muita diferença entre os resultados quando se mantém $\alpha \in [0,5; 0,7]$, $c_1 \in [0,6; 0,7]$, $c_2 \in [0,3; 0,4]$ ou $a \in [5; 7]$. Entretanto, manteve-se os valores que apresentaram os melhores resultados.

O aumento da taxa global de lançamento de formigas para um valor maior que o mostrado na Tabela 6.1 praticamente não melhora o desempenho geral do sistema. Um aumento muito grande nessa taxa, por outro lado, pode aumentar a probabilidade de perda de pacotes, uma vez que as formigas compartilham os *buffers* ópticos com os pacotes de dados e, por isso, podem contribuir para o aumento da contenção nos nós OPS.

A rede NTTNet necessita de uma taxa maior de geração de formigas e um período de exploração pelas formigas (I_{warmup}) maior, uma vez que ela possui três vezes mais nós que a rede NSFNet. Por outro lado, os outros parâmetros permanecem os mesmos para ambas as redes. Isso reflete o fato que os valores dos parâmetros escolhidos são robustos para diferentes topologias ou cargas.

6.3.2 Tráfego Auto-similar

As rede NSFNet e NTTNet foram simuladas com enlaces de mesmo comprimento, com valores iguais a 100 e 80 km, respectivamente.

Foi usado um tráfego auto-similar, conforme descrito no Apêndice A. O parâmetro de forma da distribuição Bounded Pareto (BP) é α_{Pareto} e os valores mínimo e máximo de tamanho de pacote são L_{min} e L_{max} , respectivamente.

Após uma série de testes, os parâmetros usados nas simulações com tráfego auto-similar estão mostrados na Tabela 6.2:

Tabela 6.2: Parâmetros usados na simulação de redes OPS com tráfego auto-similar.

Parâmetro	Símbolo	Valor
Tamanho médio de pacote	L_p	328,3 bytes
Tamanho mínimo de pacote	L_{min}	64 bytes
Tamanho máximo de pacote	L_{max}	10^6 bytes
Parâmetro de forma da distribuição de Pareto	α_{Pareto}	1,2
Unidade de atraso do <i>buffer</i>	D	0,1 μs

Os demais parâmetros permanecem iguais ao da Tabela 6.1. Isso é muito importante, pois somente os valores dos parâmetros relacionados diretamente ao tamanho do pacote foram mudados, enquanto os valores dos demais parâmetros continuam sendo ótimos para o problema.

6.4 Resultados

Como foram simulados dois tipos de tráfego diferentes, cada tipo de simulação recebeu uma Sub-seção própria, cujos resultados estão apresentados a seguir.

6.4.1 Tráfego Poissoniano

Primeiramente, usamos a rede NSFNet para avaliar o desempenho do algoritmo baseado em ACO quando comparado a uma estratégia de caminho mais curto. Os resultados são mostrados na Figura 6.7.

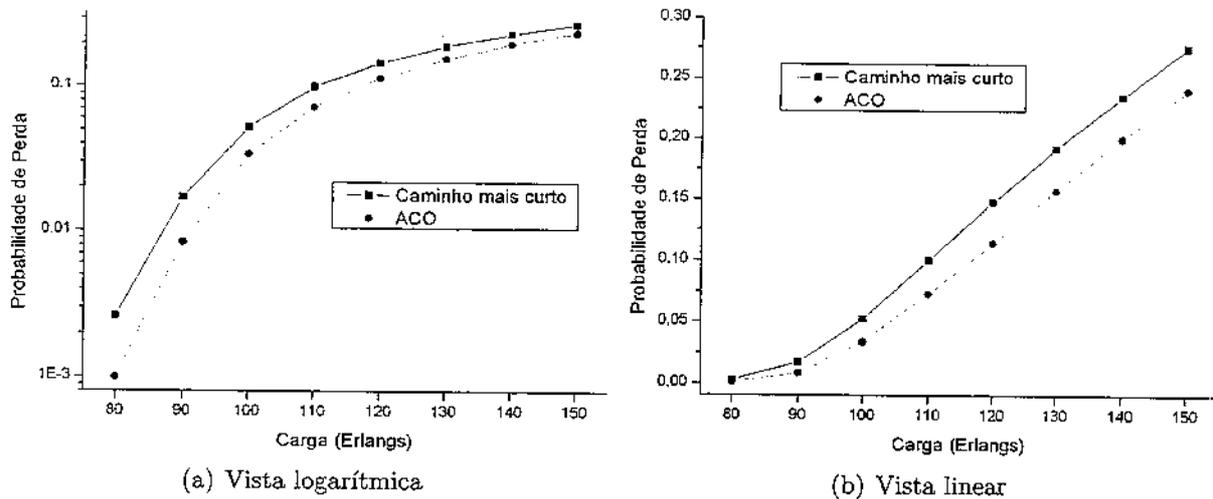


Figura 6.7: Roteamento por caminho mais curto versus roteamento baseado em ACO para a rede NSFNet com tráfego poissoniano.

As mesmas simulações foram feitas para a rede NTTNet, como pode ser visto na Figura 6.8.

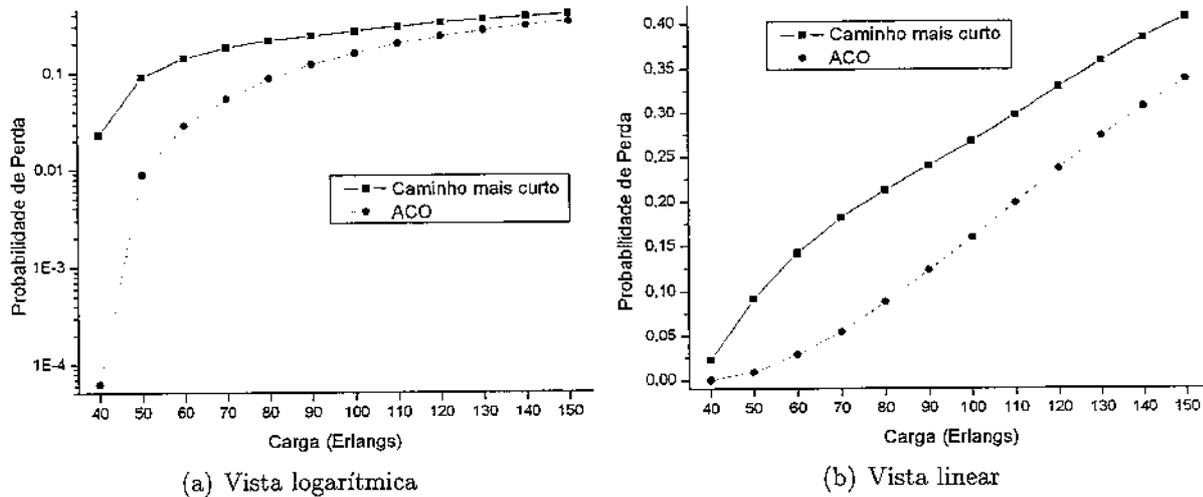


Figura 6.8: Roteamento por caminho mais curto versus roteamento baseado em ACO para a rede NTTNet com tráfego poissoniano.

Na Figura 6.7, obtemos uma melhora sobre o algoritmo de roteamento por caminho mais curto, mas os resultados são mais significativos na Figura 6.8, porque a rede NTTNet não é bem-balanceada como a rede NSFNet, de forma que há mais espaço para melhorias do roteamento na rede NTTNet, uma vez que o algoritmo baseado em ACO é capaz de balancear a carga na rede. Este fato já tinha sido percebido em [9].

Em ambas figuras, o desvio padrão é muito pequeno para ser visto e nenhuma diferença perceptível na probabilidade de perda de pacotes foi notada para os diferentes valores de tamanho de enlace, que não foram mostrados para facilitar o entendimento dos gráficos. Essa é uma interessante característica do algoritmo, pois ele não é prejudicado pelo atraso de propagação. De fato, protocolos do tipo enlace-estado costumam ter problemas com informações desatualizadas devido a atrasos de propagação ou pela baixa frequência de disseminação da informação, o que não ocorre no algoritmo proposto.

A redução da probabilidade de perda na rede devido ao balanceamento da carga no roteamento baseado em ACO pode ser observado nas Figuras 6.9 e 6.10.

Os enlaces com menor utilização no roteamento por caminho mais curto estão, em geral,

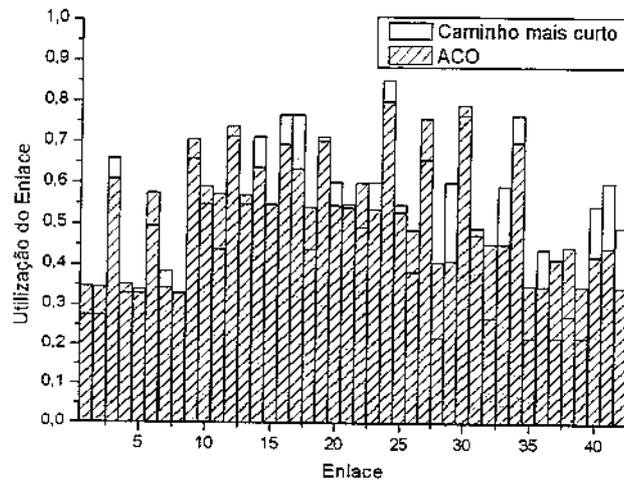
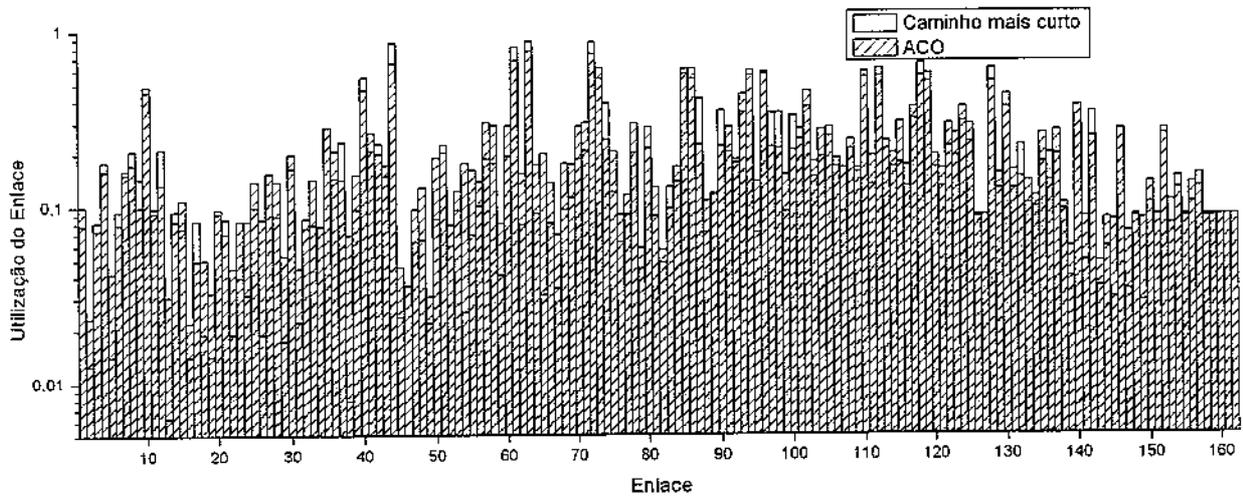


Figura 6.9: Utilização média de enlace na rede NSFNet com carga igual a 80 erlangs para tráfego poissoniano.

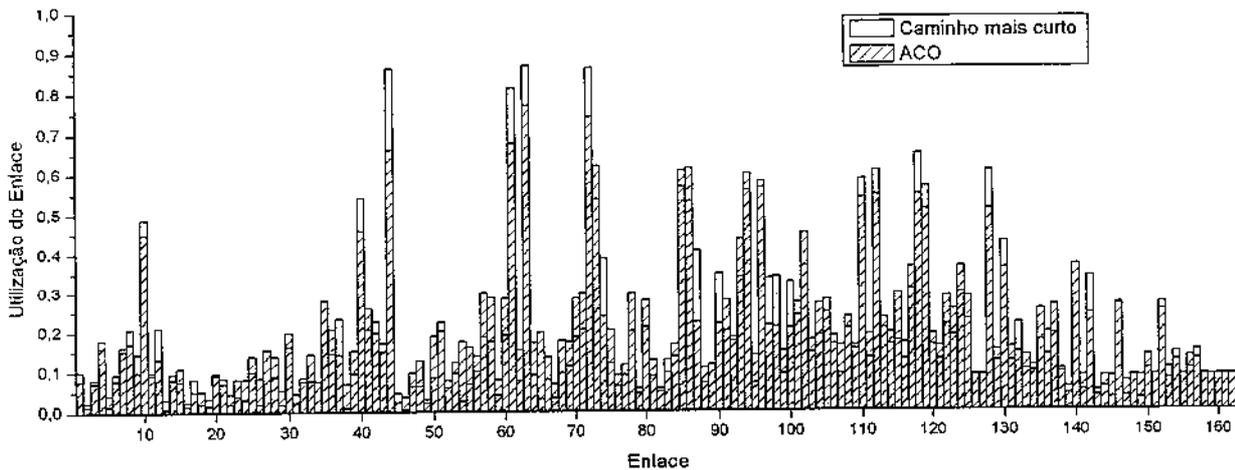
mais carregados no roteamento baseado em ACO, como pode ser constatado nas Figuras 6.9 e 6.10(a). Por outro lado, os enlaces com maior utilização no roteamento por caminho mais curto são menos usados no roteamento por formigas, como pode ser visto nas Figuras 6.9 e 6.10(b). Isto é um excelente efeito, porque a perda cresce muito rapidamente com a ocupação do *buffer*.

Na Figura 6.11, pode-se observar os *hot-spots*, que são os nós que mais contribuem para a perda de pacotes. Usando-se o roteamento por caminho mais curto, o nó 8 na rede NSFNet e os nós 4, 15, 21, 25, 40 e 45 na rede NTTNet são considerados *hot-spots*. Contudo, quando se usa o roteamento por formigas, pode-se observar a redução da perda em todos esses nós, com exceção do nó 21 na rede NTTNet. Assim, o algoritmo baseado em ACO aumentou a imparcialidade em termos de probabilidade de perda dos nós da rede.

Finalmente, podemos notar na Figura 6.12 o efeito de se permitir que pacotes que poderiam ter sido descartados devido à contenção na estratégia de caminho mais curto, sejam roteados em caminhos mais longos, mas menos congestionados. Em outras palavras, o algoritmo baseado em ACO minimiza a perda de pacotes através de desvios estabelecidos dinamicamente do tráfego “excedente” para rotas mais longas. A figura mostra a percentagem de chegadas de pacotes ao



(a) Vista logarítmica



(b) Vista linear

Figura 6.10: Utilização média de enlace na rede NTTNet com carga igual 40 erlangs para tráfego poissoniano.

destino em relação ao número total de pacotes enviados na simulação. Note que a soma dessas percentagens sobre todos os comprimentos de rota é igual ao complemento da probabilidade de perda mostrada na Figura 6.8 para uma carga de 80 erlangs.

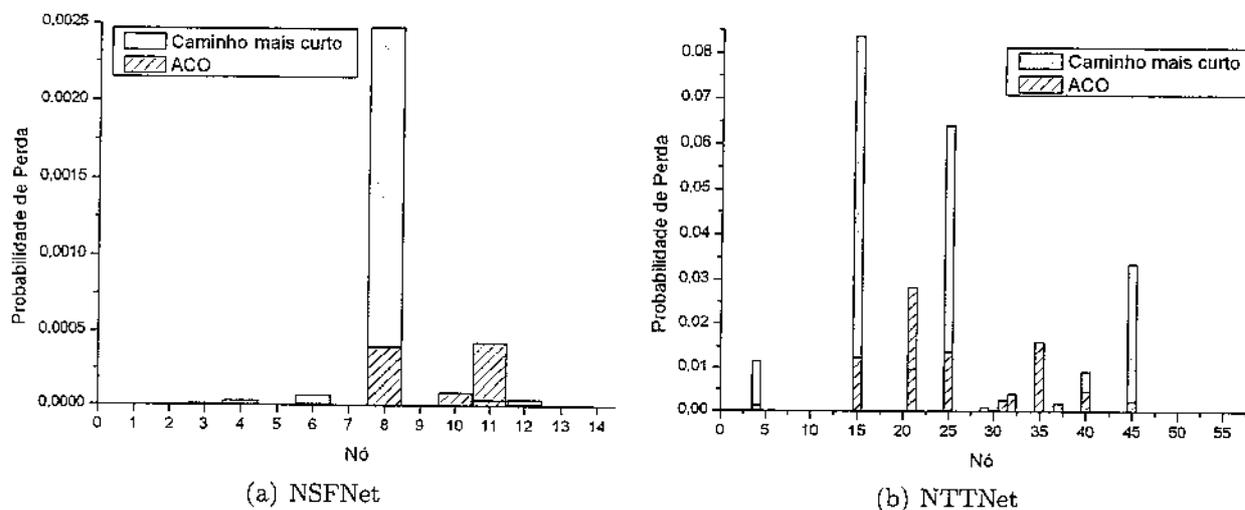


Figura 6.11: Hot-spots com carga = 80 erlangs para tráfego poissoniano.

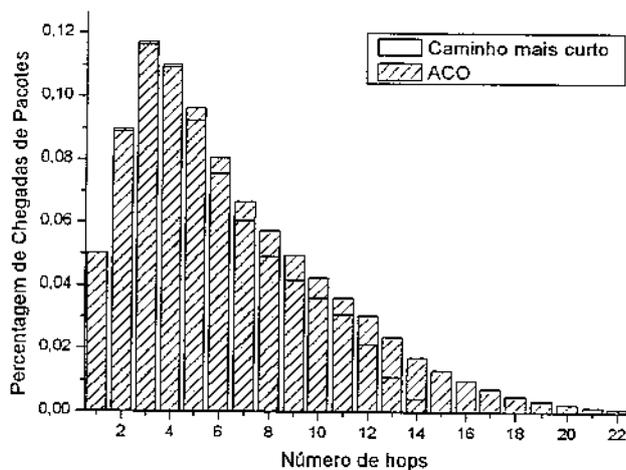


Figura 6.12: Percentagem de chegadas de pacote por comprimento da rota na rede NTTNet com carga = 80 erlangs para tráfego poissoniano.

6.4.1.1 Restauração

Foram simuladas falhas simples de nó e de enlace para todos os nós e enlaces de ambas as redes. Todas as simulações foram executadas com uma carga de 90 erlangs. Obteve-se a média dos resultados que estão mostrados nas Figuras 6.13, 6.14 e 6.15.

Na Figura 6.13, pode-se perceber que a perda acumulada média na rede NSFNet para o

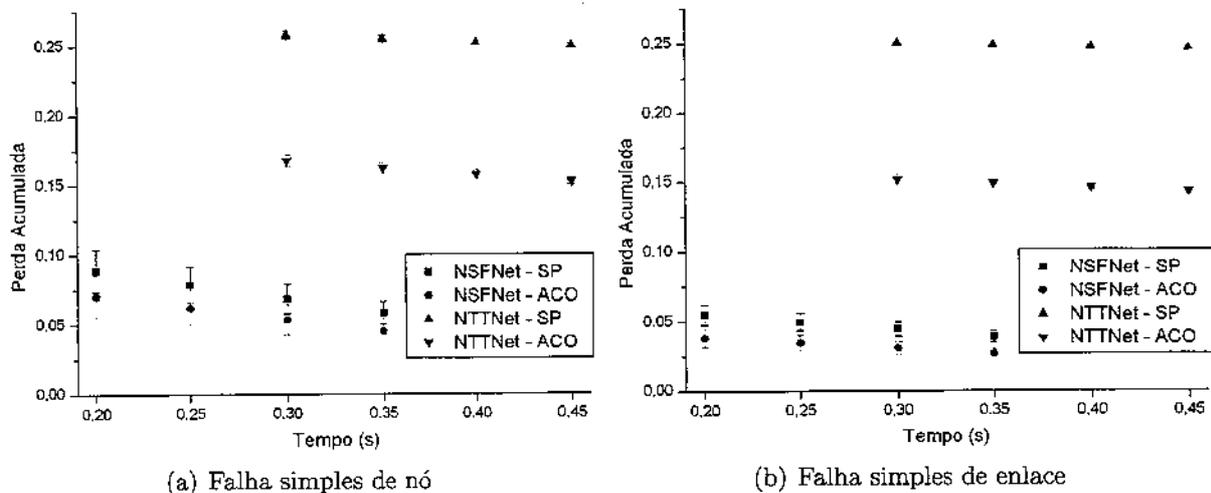


Figura 6.13: Perda acumulada para restauração por caminho mais curto (SP) versus ACO.

algoritmo proposto é menor que a restauração por caminho mais curto. Para a rede NTTNet, mesmo em caso de falha, o algoritmo baseado em ACO tem um desempenho em termos de probabilidade de perda muito melhor que o caso de caminho mais curto.

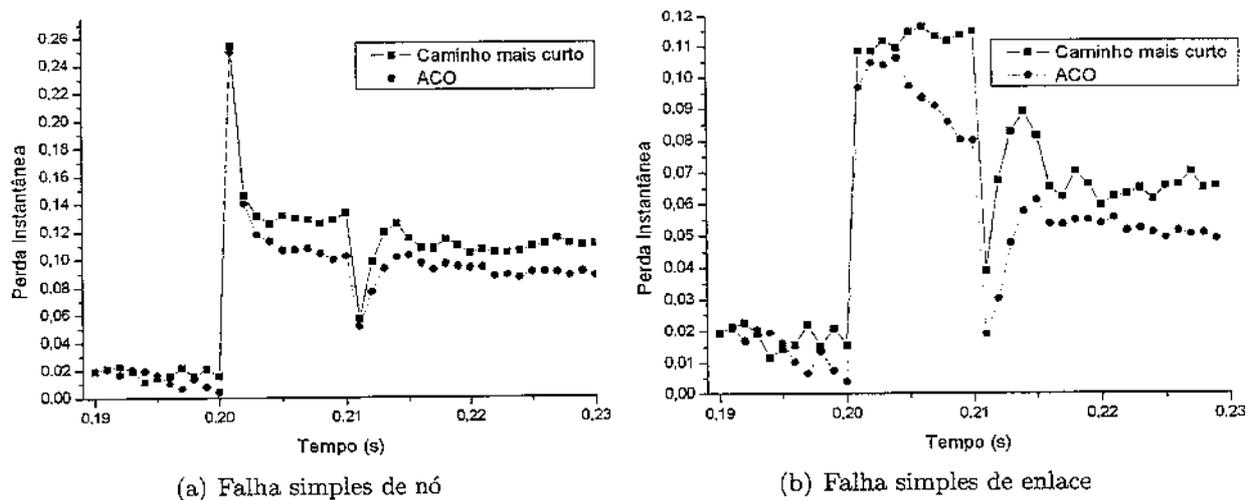


Figura 6.14: Perda instantânea na rede NSFNet para restauração por caminho mais curto versus restauração baseada em ACO.

Olhando a Figura 6.14, é possível perceber que, antes da falha no instante $t = 0,2$ s, a perda instantânea é similar para ambos os algoritmos. Esse fato também pode ser verificado com a

ajuda da Figura 6.7. Quando a falha ocorre, as formigas que circulam na rede começam a evitar o enlace ou nó defeituoso antes mesmo que a rede detecte e isole a falha no instante $t = 0,21$ s. Nesse instante, o processo de notificação inicia e se percebe uma grande diminuição na perda de pacotes em ambos os algoritmos. Quando o sistema estabiliza, é fácil notar que o algoritmo baseado em ACO atinge um nível de perda de pacotes instantâneo menor que no esquema de caminho mais curto, sendo mais eficiente para re-rotear o tráfego para rotas alternativas.

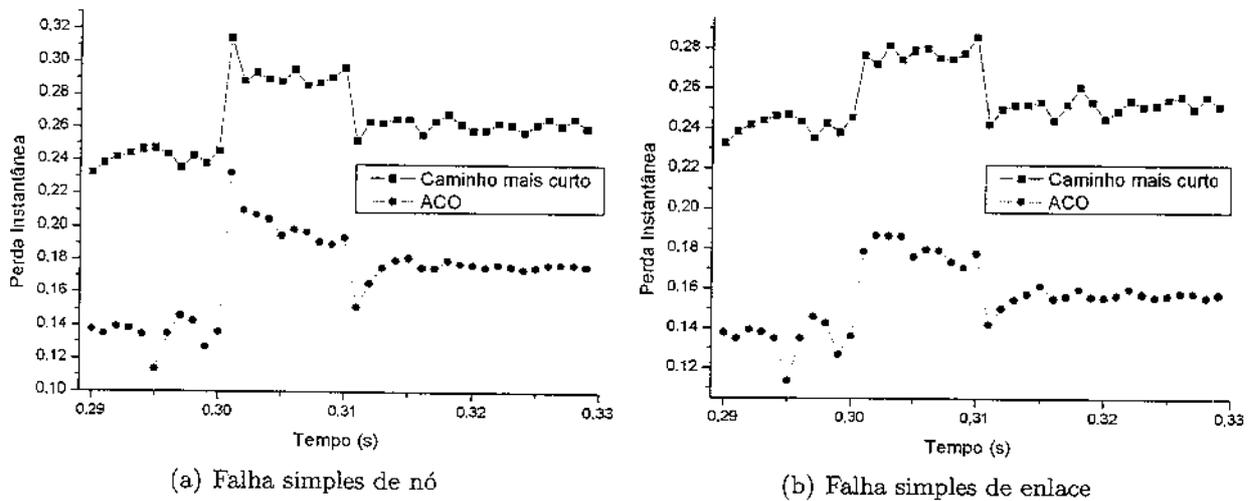


Figura 6.15: Perda instantânea na rede NTTNet para restauração por caminho mais curto versus restauração baseada em ACO.

Além disso, é possível notar na Figura 6.15 essa mesma propriedade do algoritmo baseado em ACO de reagir à falha antes mesmo que ela seja detectada e localizada. De fato, uma rota afetada por uma falha deixa de receber o reforço do ferormônio associado à ela, de forma que rotas alternativas acabam por substituí-la no transporte dos pacotes.

6.4.2 Tráfego Auto-Similar

Como já explicado no Apêndice A, a simulação de uma distribuição de Pareto para representar o tamanho de pacotes de um tráfego auto-similar não tem aplicação prática, pois normalmente a rede possui um tamanho máximo de pacote. Assim, empregamos a distribuição

Bounded Pareto (BP) para simular esse tipo de tráfego, com diversos valores de tamanho máximo de pacote mostrados na Tabela 6.3:

Tabela 6.3: Valores de média e variância para $k = 64$ e diferentes valores de p na distribuição BP.

p	Média	Variância
10^5	295,83	$2,20 \times 10^6$
10^6	328,34	$1,39 \times 10^7$
10^7	348,88	$8,78 \times 10^7$
10^8	361,84	$5,54 \times 10^8$
10^9	370,02	$3,50 \times 10^9$
10^{10}	375,18	$2,21 \times 10^{10}$
∞	384,00	Não existe

Os valores de variância para a distribuição BP são muito grandes. Basta lembrar que a variância da distribuição de Poisson é igual a própria média da distribuição.

Para se avaliar o impacto do tráfego auto-similar nas redes tipo OPS, foram feitas algumas simulações para um buffer com $W = 8$ e $B = 8$. Na Figura 6.16 são mostrados os resultados obtidos para o tráfego poissoniano, enquanto na Figura 6.17 são mostrados os resultados simulados para o tráfego auto-similar.

É possível perceber, comparando-se as Figuras 6.16 e 6.17, que um perfil de tráfego auto-similar aumenta enormemente a probabilidade de perda nos *buffers* ópticos.

Como a rede NSFNet tem poucas opções de rotas alternativas, o algoritmo baseado em ACO tem um desempenho muito semelhante em termos de probabilidade de perda, como podemos observar na Figura 6.18.

Entretanto, na rede NTTNet, o desempenho do algoritmo baseado em ACO é claramente superior em termos de probabilidade de bloqueio, como podemos ver na Figura 6.19:

A Figura 6.20 mostra a utilização média de cada enlace na rede NSFNet com uma carga de 80 erlangs. Embora a redução total de perda de pacotes seja muito pequena quando se usa algoritmo baseado em ACO, este algoritmo melhora o balanceamento da carga da rede. De modo

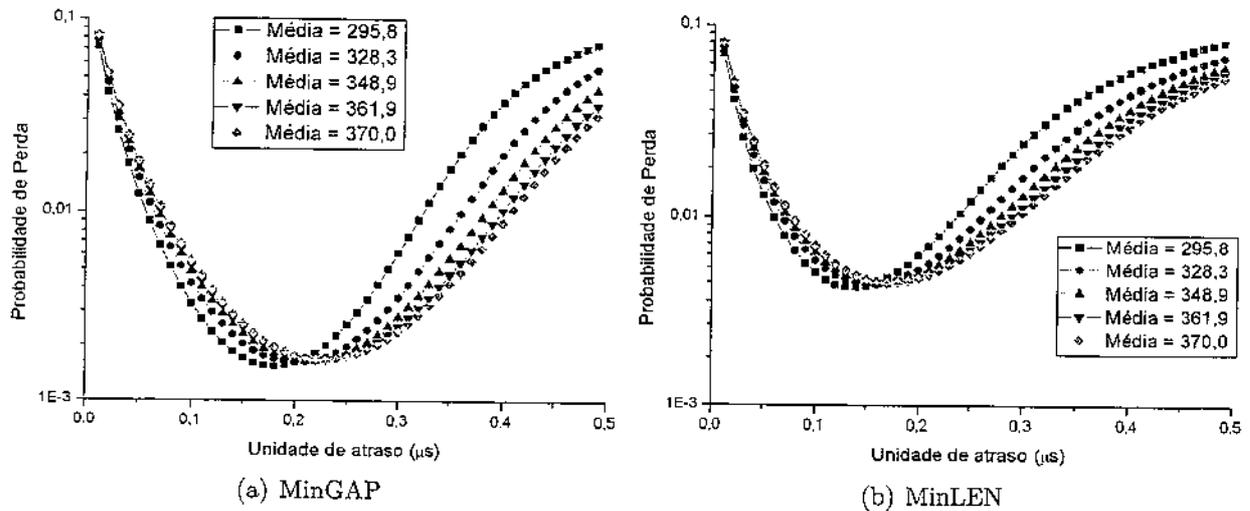


Figura 6.16: Probabilidade de perda em um buffer com carga = 6.4 erlangs e tráfego poissoniano.

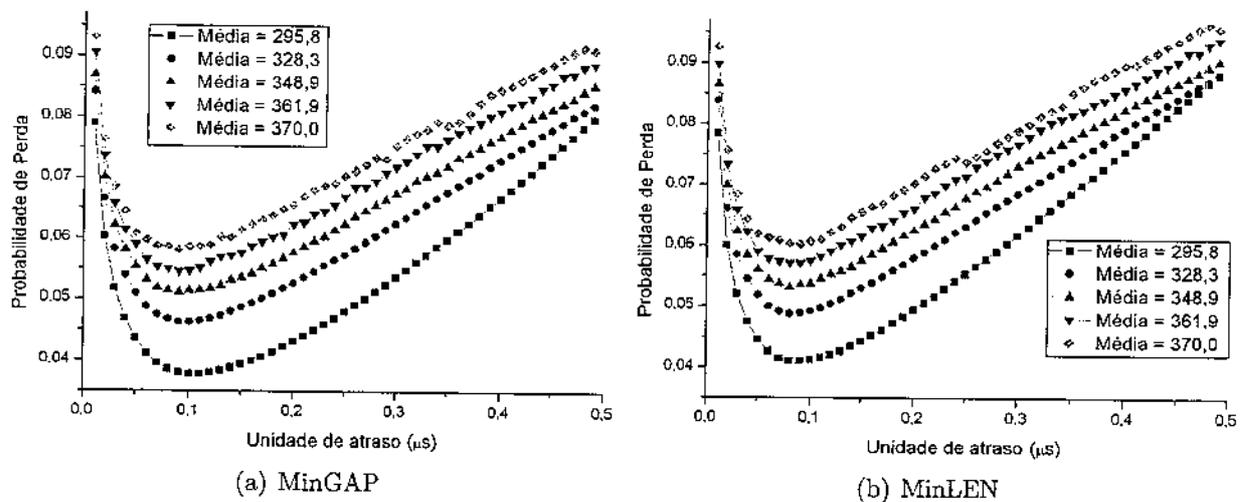


Figura 6.17: Probabilidade de perda em um buffer com carga = 6.4 erlangs e tráfego auto-similar.

geral, os enlaces mais carregados no roteamento por caminho mais curto são menos utilizados no roteamento baseado em ACO, enquanto os enlaces menos usados no roteamento por caminho mais curto recebem uma carga maior no roteamento baseado em ACO.

Podemos observar na Figura 6.21, a utilização média de cada enlace da rede NTTNet para uma carga de 40 erlangs. O mesmo efeito de balanceamento de carga notado na rede NSFNet

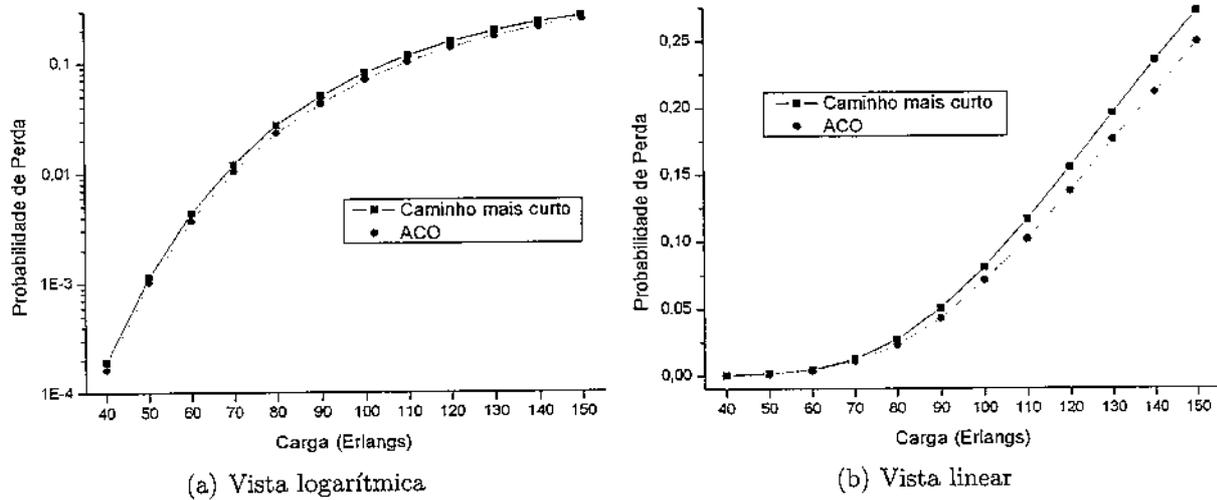


Figura 6.18: Roteamento por caminho mais curto versus roteamento baseado em ACO para a rede NSFNet com tráfego auto-similar.

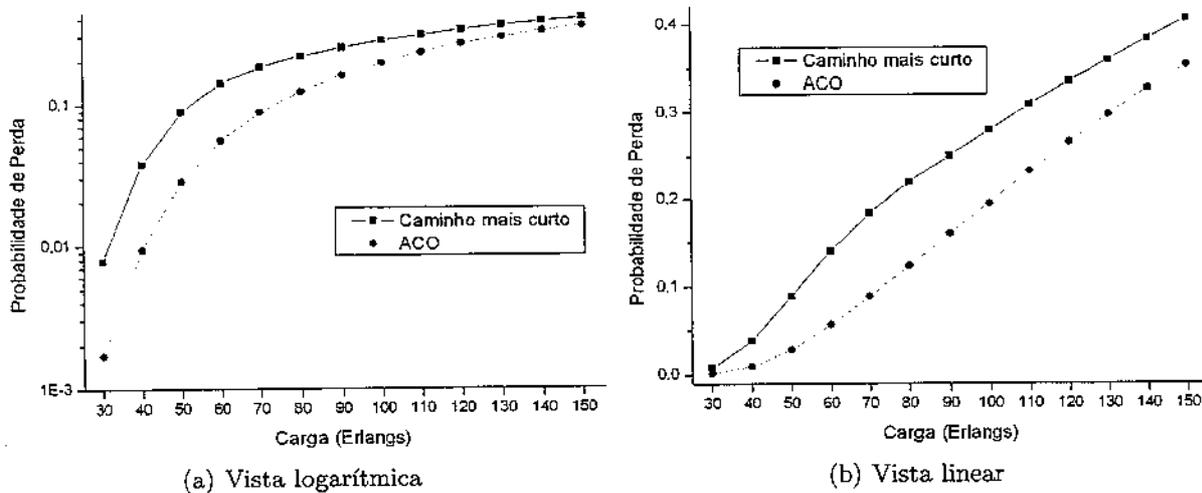


Figura 6.19: Roteamento por caminho mais curto versus roteamento baseado em ACO para a rede NTTNet com tráfego auto-similar.

também pode ser constatado nesta figura para a rede NTTNet.

O efeito de balanceamento de carga também aumenta a imparcialidade na perda de pacotes por nó. O algoritmo baseado em ACO reduz o número de “hot-spots” e distribui de forma mais homogênea a perda de pacotes entre os nós da rede, conforme podemos constatar na Figura 6.22:

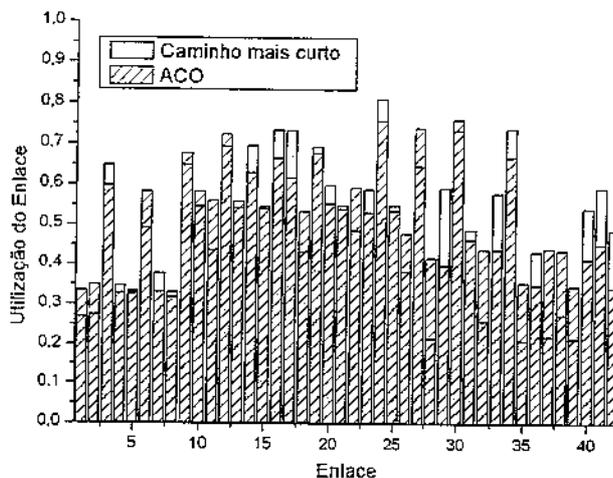


Figura 6.20: Utilização média de enlace na rede NSFNet com carga = 80 erlangs para tráfego auto-similar.

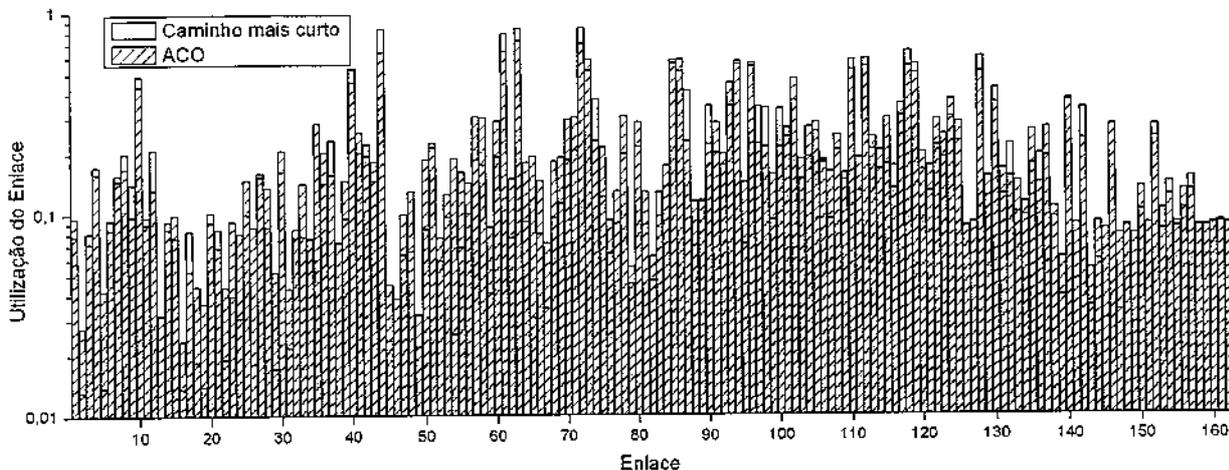
Finalmente, é possível observar, na Figura 6.23, também o efeito de desvio do tráfego que poderia congestionar uma rota para rotas mais longas, mas menos congestionadas. Note que a soma dessas percentagens sobre todos os comprimentos de rota é igual ao complemento da probabilidade de perda mostrada na Figura 6.19 para uma carga de 80 erlangs.

6.4.3 Considerações finais

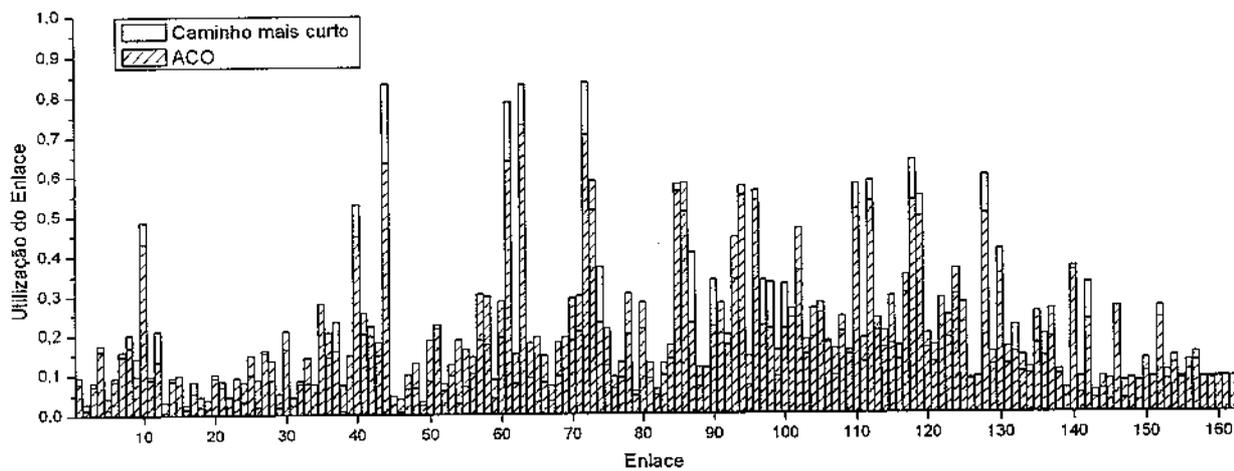
A capacidade de balanceamento de carga do algoritmo baseado em ACO é o recurso mais importante para se reduzir a probabilidade de perda de pacotes na rede.

O algoritmo proposto também é capaz de gerenciar falhas na rede. De fato, este tipo de algoritmo é distribuído por natureza e pode tratar falhas localmente com simples cálculos nos níveis de feromônio. O processo de recuperação do tráfego começa antes mesmo que a falha seja detectada e localizada, pois as formigas que selecionam um enlace ou nó estragado morrem, enquanto outras formigas reforçam caminhos alternativos, que evitam a falha.

Finalmente, ACO tende a ser mais escalável que algoritmos de inundação [16], uma vez



(a) Vista logarítmica



(b) Vista linear

Figura 6.21: Utilização média de enlace na rede NTTNet com carga = 40 erlangs para tráfego auto-similar.

que a memória de cada formiga tipicamente cresce de com o tamanho médio da rota na rede, enquanto a informação inundada tende a crescer de forma quadrática com o número de nós na rede. Como as formigas viajam junto com os pacotes de dados na rede, a rede tem uma visão mais atualizada do congestionamento nos nós do que quando se utiliza inundação, que pode publicar informações desatualizadas devido a atrasos de propagação e processamento.

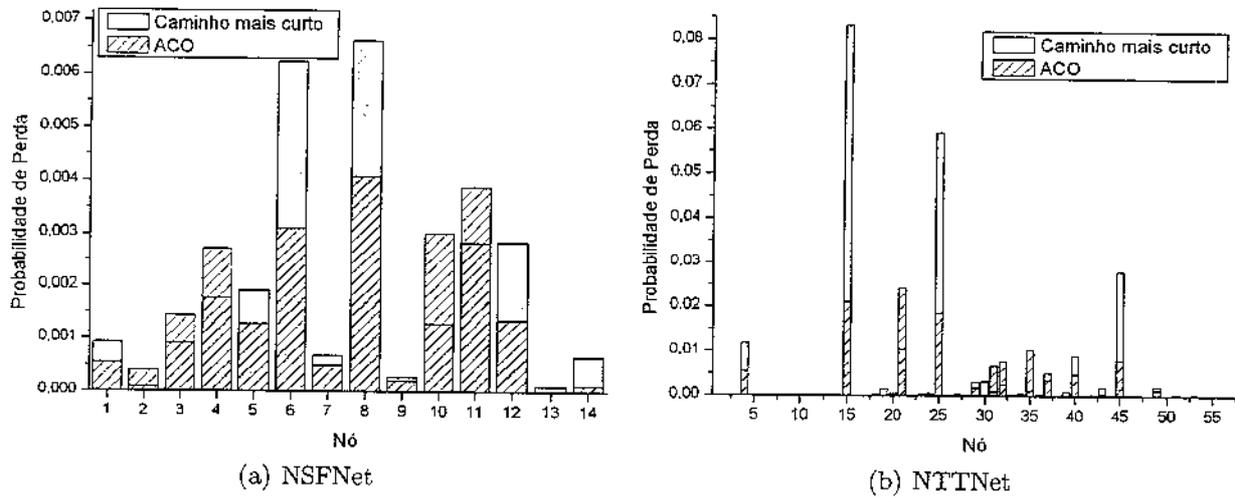


Figura 6.22: Hot-spots com carga = 80 erlangs para tráfego auto-similar.

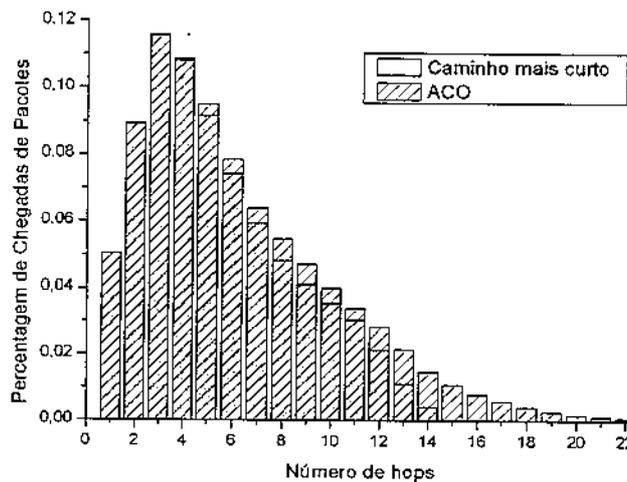


Figura 6.23: Percentagem de chegadas de pacote por comprimento da rota na rede NSFNet com carga = 80 erlangs para tráfego auto-similar.

Capítulo 7

Conclusões

A otimização por colônia de formigas demonstrou ser extremamente versátil em redes ópticas. Entre as aplicações dessa técnica podemos citar:

1. Aprovisionamento e restauração de caminhos ópticos em redes dinâmicas;
2. Roteamento que leva em consideração degenerações na camada física;
3. Gerenciamento integrado de caminhos ópticos e de recursos de processamento em arquiteturas de grid; e
4. Roteamento adaptativo e restauração em redes de pacotes ópticos.

De fato, conforme apresentado neste trabalho, o emprego de otimização por colônia de formigas em redes ópticas apresenta um série de vantagens:

Escalabilidade A população de formigas pode ser adaptada de acordo com a dinâmica e o tamanho da rede. Além disso, as formigas interagem localmente com o nó, criando um sistema de roteamento distribuído. Como as formigas são agentes muito simples, o seu uso introduz um overhead muito pequeno no controle do roteamento.

Os algoritmos baseados em ACO tendem a ser mais escaláveis que os algoritmos que usam “inundação” para a disseminação de informação na rede [16], pois a memória de

cada formiga cresce linearmente com o tamanho do caminho médio da rede, enquanto a disseminação da informação desses outros algoritmos tendem a crescer de forma quadrática com o número de nós da rede. A escalabilidade do algoritmo AntNet original foi provada em [52].

Tolerância a falhas Não há controle centralizado ou ponto único de falha. Dessa forma, mesmo com a perda de alguns nós ou enlaces, a rede pode continuar operando. Além disso, as formigas artificiais possuem um grande capacidade de auto-organização [53], de forma que a rede seja resiliente a falhas.

No caso das redes OPS, as formigas são capazes de começar a recuperação do tráfego antes mesmo da detecção e localização da falha.

Adaptação As formigas são capazes de se adaptar rapidamente a mudanças no tráfego, na topologia ou no congestionamento da rede, provendo naturalmente um balanceamento da carga [54]. Por isso, são apropriadas para a implementação de funções de engenharia de tráfego em redes.

Velocidade Mudanças na topologia ou congestionamento podem ser rapidamente notificadas, pois as formigas são geradas numa taxa muito maior do que as normalmente encontradas no sistemas de roteamento tradicionais. Assim, o roteamento baseado em ACO tende a disseminar uma informação mais atual do que os protocolos de roteamento baseados em “inundação”.

Modularidade As formigas podem agir independentemente de outros sistemas ou camadas da rede óptica.

Autonomia Pouca ou nenhuma supervisão é necessária. As formigas podem agir como representantes do usuário, alocando, por exemplo, recursos de processamento e criando caminhos ópticos em um grid.

Paralelismo A operação das formigas artificiais é inerentemente paralela.

Os valores dos parâmetros relacionados ao roteamento por formigas em todas as aplicações desta Tese são muito semelhantes. Isto reflete a robustez dos parâmetros para diferentes tipos de tráfego, topologia e aplicação. Além disso, esses valores são muito próximos aos usados no algoritmo AntNet original [9, 10].

O algoritmo baseado em ACO proposto nesta Tese pode ser usado em outras aplicações com mínimas adaptações. De modo geral, basta definir qual será a métrica heurística local e se, por ventura, são necessárias outras estruturas de dados específicas ao problema em cada nó da rede. Por exemplo, uma área onde esse tipo de algoritmo tem apresentado bons resultados é no roteamento em redes *ad-hoc* [55].

Além disso, os algoritmos propostos neste trabalho podem ser facilmente integrados a um plano de controle GMPLS, sem necessidade alguma de modificação.

Entre as desvantagens do roteamento por formigas podemos citar duas principais. Seu desempenho está ligado à taxa de lançamento global de formigas, que é um parâmetro que varia de acordo com o problema a ser tratado e que não possui um método direto ou analítico para determiná-lo. O desempenho do algoritmo também depende de uma métrica heurística local apropriada, que normalmente é estabelecida através de um método de tentativa e erro, demandando conhecimento específico do problema.

Finalmente, o roteamento baseado em otimização em colônia de formigas constitui uma alternativa mais eficiente ao roteamento baseado em topologia, como o roteamento por caminho mais curto, sobretudo em topologias com um maior número de rotas alternativas entre os diferentes pares origem-destino. É imparcial a comparação entre esses algoritmos, pois as formigas artificiais transportam somente informações relacionadas ao caminho percorrido por elas na rede.

Apêndice A

Simulação de Tráfego

A.1 Tráfego Poissoniano com Duração Exponencialmente Distribuída

A forma mais clássica de se modelar tráfego em redes de telecomunicação é através de um tráfego Poissoniano. Tal modelo é baseado na modelagem das redes de telefonia tradicionais, onde se assume a independência das chamadas e a duração das chamadas segue um modelo exponencialmente distribuído.

Assim, para modelarmos esse tipo de tráfego na rede, usamos uma fila do tipo M/M/n.

Para se gerar o tempo de duração de pacote ou conexão x e o tempo entre chegadas y que obedecem a esse tipo de tráfego, basta usar as seguintes expressões:

$$x = -\frac{\log(1-u)}{\mu} \quad (\text{A.1})$$

$$y = -\frac{\log(1-u^*)}{\lambda} \quad (\text{A.2})$$

, onde μ é a taxa de serviço de conexões ou pacotes, que é igual ao inverso do tamanho médio do pacote ou da duração média das conexões, λ é a taxa entre chegadas de conexões ou pacotes e

$u, u^* = U(0, 1)$, isto é, u e u^* são obtidos de uma distribuição Uniforme entre $[0, 1]$. Esses números são gerados aleatoriamente, sendo que em nosso caso, usamos um gerador pseudo-aleatório muito eficiente, chamado Mersenne Twister [56].

Finalmente, a carga ρ na rede é dada pela seguinte expressão:

$$\rho = \frac{\lambda}{\mu} \quad (\text{A.3})$$

A.2 Tráfego Auto-Similar

As chegadas de pacotes (ou de requisição de conexão) são normalmente geradas por entidades independentes. Assim, o processo de chegada é normalmente Poissoniano, de forma que os tempos entre chegadas são exponencialmente distribuídos.

Entretanto, alguns estudos demonstraram [57, 58] que a duração de um pacote (ou conexão) está associada a distribuições de cauda pesada. Dessa forma, a auto-similaridade está ligada ao tempo de duração dos pacotes (ou conexões).

Em geral, uma distribuição de cauda pesada tem $Pr\{X > x\} \sim x^{-\alpha}$, onde $0 < \alpha < 2$ e α é o parâmetro de forma da distribuição. A distribuição de cauda pesada mais simples é a de Pareto, que têm como função de probabilidade:

$$f(x) = \alpha k^\alpha x^{-\alpha-1}, \quad \alpha, k > 0, \quad \wedge x \geq k \quad (\text{A.4})$$

onde k é o mínimo valor da distribuição. Se $\alpha < 1$, a média e a variância não existem e por isso não é de interesse prático nessa discussão. Se $1 \leq \alpha \leq 2$, somente a variância não existe. Para $\alpha > 2$, a distribuição de Pareto não exhibe auto-similaridade.

A duração de um evento em uma distribuição de cauda pesada tem as seguintes propriedades:

1. Taxa de término decrescente: Em particular, quanto mais o pacote (ou conexão) está

ativo, mais tempo é esperado para continuar ativo;

2. Variância infinita (se $\alpha < 1$, a média também é infinita);
3. A propriedade que uma pequena fração ($< 1\%$) dos maiores pacotes (ou conexões mais duradouras) são responsáveis por grande parte da carga na rede. Essa propriedade é conhecida como propriedade da cauda pesada.

O tráfego auto-similar tem como propriedade a preservação da distribuição de um processo aleatório em diferentes escalas de tempo. O parâmetro de Hurst H representa o grau de auto-similaridade do tráfego. Quando $0,5 < H \leq 1$, o tráfego é dito auto-similar, sendo que um valor de H próximo de 1 indica que o tráfego contém um alto teor de auto-similaridade.

$$H = \frac{3 - \alpha}{2} \quad (\text{A.5})$$

Como vários processos reais tem cauda pesada, mas valores finitos, é mais interessante [59] usar uma função de Pareto limitada – Bounded Pareto (BP). Assim, para modelarmos o tráfego auto-similar em uma rede, usamos uma fila do tipo M/BP/n.

A função densidade de probabilidade da distribuição de Pareto limitada é dada pela seguinte expressão:

$$f(x) = \alpha x^{-\alpha-1} \frac{k^\alpha}{1 - \left(\frac{k}{p}\right)^\alpha}, \quad k \leq x \leq p, \quad (\text{A.6})$$

onde k e p são a mínima e máxima possíveis observações, respectivamente.

Como a distribuição BP tem todos os momentos finitos, ela não é uma distribuição de cauda pesada no sentido estrito. Entretanto, essa distribuição exibe uma alta variabilidade se $k \ll p$. Além disso, o segundo momento cresce exponencialmente quando α diminui e a distribuição BP ainda exibe a propriedade de cauda pesada.

A média (primeiro momento) da distribuição BP é dada pela seguinte expressão:

$$E_x(x) = \frac{k^\alpha}{\left[1 - \left(\frac{k}{p}\right)^\alpha\right]} \left(\frac{\alpha}{\alpha - 1}\right) \left(\frac{1}{k^{\alpha-1}} - \frac{1}{p^{\alpha-1}}\right) \quad (\text{A.7})$$

E o segundo momento é dado pela seguinte fórmula:

$$E_x(x^2) = \frac{k^\alpha}{\left[1 - \left(\frac{k}{p}\right)^\alpha\right]} \left(\frac{\alpha}{\alpha - 2}\right) \left(\frac{1}{k^{\alpha-2}} - \frac{1}{p^{\alpha-2}}\right) \quad (\text{A.8})$$

Finalmente, para se gerar um número x da distribuição BP:

$$x = \frac{k}{\left\{1 + u \left[\left(\frac{k}{p}\right)^\alpha - 1\right]\right\}^{\frac{1}{\alpha}}} \quad (\text{A.9})$$

, onde $u = U(0, 1)$, isto é, u é obtido de uma distribuição Uniforme entre $[0,1]$.

Apêndice B

O Padrão GMPLS

A rede óptica deve implementar mecanismos de gerenciamento e controle eficientes para estabelecer caminhos ópticos. Esses mecanismos podem ser tanto centralizados como distribuídos. No caso do controle centralizado, todas as requisições de conexão passam por um controlador central, que é responsável por todos os estabelecimentos de conexão. Tal controlador deve manter informações sobre o estado global da rede e as requisições de conexão são processadas de forma seqüencial. Por outro lado, sob controle distribuído, todas as conexões são processadas de forma concorrente pelos diferentes nós. Cada nó toma suas decisões baseadas nas informações do estado da rede que ele mantém.

É nesse contexto distribuído que surgiu o padrão Generalized Multi-Protocol Label Switching (GMPLS) [60], que é definido pelo Internet Engineering Task Force (IETF) e que provê um plano de controle comum para equipamentos que operam em qualquer dos seguintes domínios: pacote, *time slot*, comprimento de onda e fibra. O padrão GMPLS simplifica a operação da rede e seu gerenciamento por automatizar o provisionamento fim-a-fim de conexões, gerenciar os recursos da rede e prover um nível de qualidade de serviço (QoS) necessário para aplicações sofisticadas.

Numa rede GMPLS, o plano de controle e o plano de dados (transporte) são logicamente separados.

O plano de controle é constituído por 3 tipos de protocolo:

Sinalização É o processo de troca de mensagens para configurar, manter, modificar e finalizar conexões no plano de dados. Essas conexões são chamadas de caminhos comutados por rótulo – Label Switched Path (LSP) no GMPLS. O protocolo responsável pela sinalização é o Resource Reservation Protocol with Traffic Engineering (RSVP-TE) [61], o qual possibilita o estabelecimento de LSPs de duas maneiras: a cada *hop* ou de forma explícita, ou seja, no nó de origem. Neste trabalho, como o cálculo de rota é distribuído, usamos o roteamento feito a cada *hop*.

Roteamento É o processo de disseminação de informação que será usado como base para o cálculo de rotas. É diferente do roteamento IP, que determina o próximo salto na rede de um pacote que vai em direção ao nó de destino. O protocolo normalmente usado para o roteamento é o Open Shortest Path First with Traffic Engineering (OSPF-TE) [21, 22].

As extensões TE desses protocolos usam informações sub-Type-Length-Value (TLV) para especificar e distribuir informação de enlaces TE¹. São definidas características de comutação para pacotes (PSC), camada 2 (L2SC), multiplexação por divisão de tempo (TDM-SC), comprimento de onda (LSC) e fibra (FSC). A unidade básica para disseminação de informações nesse protocolo é o Link-State Advertisement (LSA).

A Figura B.1 representa uma base de dados de estado-enlace de cada nó da rede. Por questões de escalabilidade, com exceção do Capítulo 4, onde são publicadas os comprimentos de onda usados pelos caminhos ópticos, somente a topologia física é divulgada, ou seja, somente há enlaces TE do tipo Lambda Switch Capable (LSC).

Gerenciamento de Enlace É o processo onde é possível que os nós da rede sejam configurados, se descubra o estado operacional dos enlaces e possibilita a detecção de falhas.

¹Um enlace TE é um enlace entre dois LSRs adjacentes que possui um conjunto de características de engenharia de tráfego.

O protocolo responsável pelo gerenciamento do enlace é o Link Management Protocol (LMP) [62] e suas extensões [63].

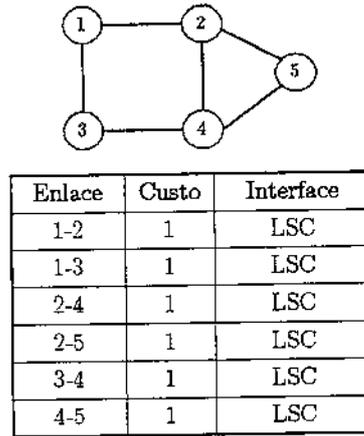


Figura B.1: Base de dados estado-enlace de cada nó óptico da rede exemplificada.

Na arquitetura apresentada neste trabalho, há uma relação lógica um-para-um entre um nó óptico e um roteador de comutação por rótulos – Label Switching Router (LSR). Cada nó óptico é capaz de comutar somente comprimentos de onda, com canais de controle para a transmissão de mensagens e pacotes no plano de controle. Esses canais podem ser implementados usando sinalização fora-de-banda.

O cálculo do caminho e a escolha do comprimento de onda é específico do equipamento ou da rede. Esse assunto é detalhado na próxima Seção.

B.1 Cálculo de Rota e Alocação de Comprimento de Onda

No sub-problema da alocação de comprimento de onda deve-se alocar um comprimento de onda para cada caminho óptico de tal forma que dois caminhos ópticos não compartilham um mesmo comprimento de onda em uma mesma fibra. Como se considerou que as redes utilizadas nesta Tese não estão equipadas com conversores de comprimento de onda, então um caminho

óptico deve ter sempre o mesmo comprimento de onda em todos os segmentos de fibra pelos quais passa. Essa propriedade é chamada de restrição de continuidade de comprimento de onda.

Para este trabalho, utilizou-se a estratégia *first-fit* para a alocação de comprimento de onda. Neste esquema, todos os comprimentos de onda são numerados. Na escolha do comprimento de onda, aquele que possuir um índice menor tem maior prioridade. Assim, o primeiro comprimento de onda disponível é selecionado. Essa estratégia é usada devido à sua simplicidade e baixo custo computacional, além de ter bom desempenho em termo de probabilidade de bloqueio e imparcialidade [24].

B.1.1 Roteamento fixo

É a maneira mais simples de se rotear, pois sempre se escolhe uma mesma rota para um dado par origem-destino. Um exemplo de tal estratégia é o roteamento pela rota mais curta. A rota mais curta para cada par origem-destino é calculada usando algoritmos de caminho mais curto, como o algoritmo de Dijkstra [8] ou de Bellman-Ford, e qualquer conexão entre os nós é estabelecida usando uma rota pré-determinada.

B.1.2 Roteamento fixo-alternado

Esta estratégia considera múltiplas rotas entre pares origem-destino. Dessa forma, deve-se manter uma lista ordenada de rotas para um par-origem destino. Por exemplo, essas rotas podem incluir a rota mais curta, a segunda rota mais curta, a terceira rota mais curta, etc. Em geral, a rota mais curta é a rota primária e as demais são consideradas como alternativas. Essas rotas podem ser calculadas, por exemplo, com o algoritmo de Yen [64], o qual é capaz de achar as k rotas mais curtas e que não contém *loops*, embora essas rotas não são necessariamente disjuntas.

Para se decidir o próximo *hop* nesta estratégia, a requisição deve fornecer além da origem e do destino, qual é a k -ésima rota a ser usada.

Nesta Tese, se restringiu o valor de k a 3, ou seja, temos uma rota mais curta e duas rotas alternativas.

B.1.3 Roteamento baseado em ACO

É um tipo de roteamento adaptativo², que se baseia nos valores da tabela de ferormônio para se decidir localmente o próximo nó da rota.

Assim, a rota é calculada *hop a hop*, selecionando o vizinho com maior nível de ferormônio na tabela de roteamento por ferormônio dado um destino. Um exemplo desse roteamento pode ser visto na Figura B.2.

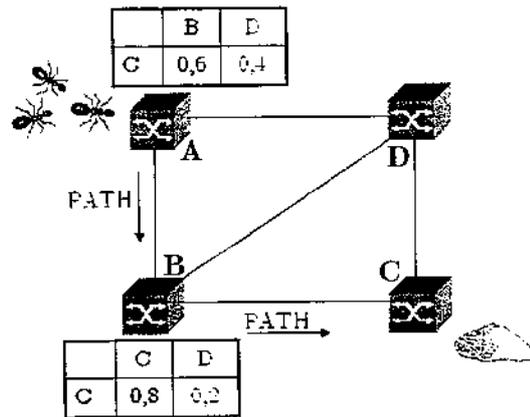


Figura B.2: Cálculo da rota de um LSP do nó A ao nó C.

B.2 Sinalização

Este trabalho usou a sinalização baseando-se no protocolo RSVP-TE, emprestando algumas idéias das extensões de *crankback* [65], que é uma técnica onde a informação de estabelecimento de uma conexão é retornada do ponto de bloqueio ou falha para permitir uma nova tentativa de

²No roteamento adaptativo, a rota de um nó origem a um nó destino é escolhida dinamicamente, dependendo do estado da rede.

estabelecimento de conexão que evita os recursos bloqueados ou indisponíveis.

Quando uma requisição para um LSP chega no nó de ingresso (origem), uma mensagem *Path* é gerada e enviada em direção do nó de egresso (destino). A mensagem é roteada *hop a hop* até chegar no nó de egresso. Como não há informação de uso de comprimento de onda nas bases de dados enlace-estado, a mensagem *Path* coleta os comprimentos de onda livres, incluindo-os na lista inclusiva de rótulos disponíveis do objeto “Label Set”.

Os rótulos devem satisfazer a restrição de continuidade de comprimento de onda, de forma que a cada nó, os rótulos que não podem ser usados são apagados do “Label Set”. O próximo é calculado conforme especificado na Seção B.1. Se o objeto “Label Set” resultante estiver vazio, então uma mensagem *PathErr* é enviada ao nó de ingresso indicando o código de erro “Routing problem/Label set” e a mensagem *Path* é descartada. Caso contrário, a mensagem *Path* é encaminhada para o próximo nó. A Figura B.3 ilustra um exemplo de LSP bloqueado devido à violação da restrição de continuidade de comprimento de onda.

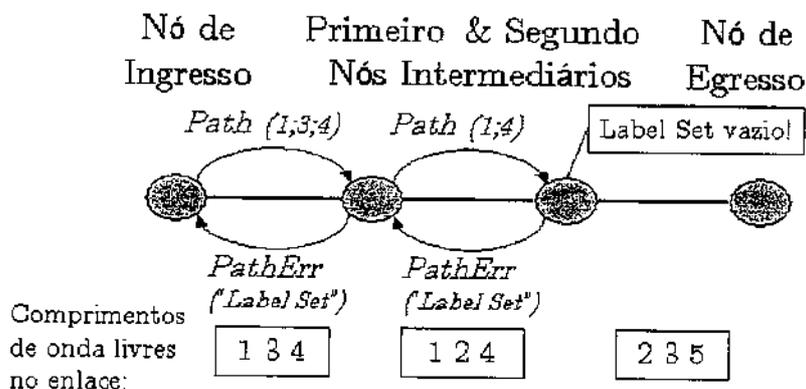


Figura B.3: Exemplo de falha de estabelecimento de um LSP devido à violação da restrição de continuidade de comprimento de onda.

Quando a mensagem *PathErr* chega no nó de ingresso indicando o erro “Routing problem/Label set”, se o roteamento fixo-alternado estiver sendo usado, então uma nova mensagem *Path* é gerada, a qual indica o uso de uma rota alternativa à anterior, até que o limite de rotas alternativas seja alcançado. Quando esse limite é ultrapassado, então a requisição é bloqueada.

Pode ocorrer que, quando se calcula a rota usando os níveis de ferormônio, a mensagem *Path* entre em um laço. Se isto vier a acontecer, a mensagem *Path* deve ser descartada e uma mensagem *PathErr* deve ser enviada para o nó de ingresso, indicando o bloqueio da requisição.

Se a mensagem *Path* consegue chegar no nó de egresso, então o primeiro rótulo disponível no objeto “Label Set” é alocado. Então, uma mensagem *Resv* é gerada e enviada ao nó de ingresso, usando o mesmo caminho da mensagem *Path*, mas no sentido oposto. A cada nó intermediário, a mensagem *Resv* aloca os recursos necessários, até o nó de ingresso, onde termina de estabelecer o LSP. A Figura B.4 ilustra um exemplo de estabelecimento de um LSP.

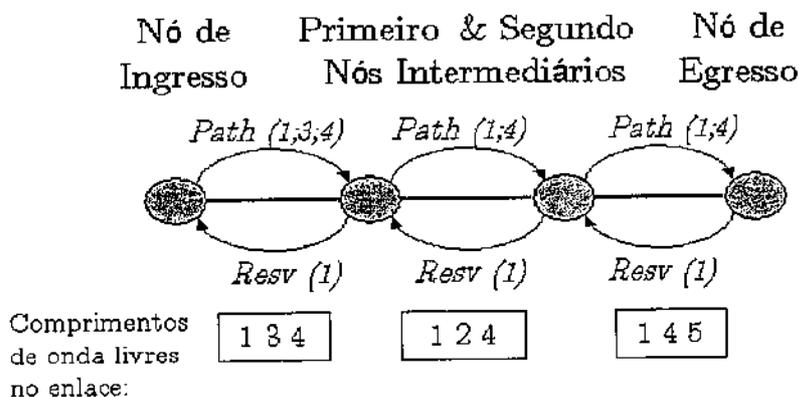


Figura B.4: Exemplo de estabelecimento de um LSP. O nó de egresso seleciona o primeiro rótulo disponível para alocação do comprimento de onda.

A reserva de recursos pode falhar durante o estabelecimento do LSP, durante o processamento da mensagem *Resv*. O caso mais comum é que o comprimento de onda indicado pelo algoritmo de *first-fit* foi ocupado por outro LSP estabelecido concorrentemente no mesmo enlace, violando a restrição de continuidade de comprimento de onda.

Neste caso, a mensagem *ResvErr* é enviada em direção ao nó de egresso indicando o erro “Admission Control Failure” e, a cada nó intermediário, os recursos alocados para esse LSP devem ser liberados. Quando a mensagem *ResvErr* chega no nó de egresso, este gera uma mensagem *PathErr* contendo o mesmo código de erro, enviando-a em direção ao nó de ingresso. Por ocasião da chegada da mensagem de *PathErr*, o nó de ingresso irá tentar estabelecer o LSP

na mesma rota anterior, enviando uma nova mensagem *Path*. Isso reduz a contenção que ocorre devido a condições de corrido na estabelecimento de LSP, resultando numa menor probabilidade de bloqueio.

Um LSP é removido da rede através de uma mensagem *PathTear*, que segue o mesmo caminho do LSP, liberando os recursos alocados até o nó de egresso.

B.2.1 Recuperação de falhas

Para a recuperação no caso de falhas, o nó mais próximo da montante da falha deve notificar o(s) nó(s) de ingresso do(s) LSP(s) afetado(s) pela falha, enviando-o(s) uma mensagem *PathErr*, a qual contém uma indicação da falha e uma indicação para liberar os recursos do LSP afetado.

Já o nó mais próximo da jusante da falha deve enviar uma mensagem *PathTear* em direção ao(s) nó(s) de egresso do(s) LSP(s) afetado(s), liberando os recursos alocados por esse(s) LSP(s).

A Figura B.5 ilustra o processo de sinalização da falha para posterior recuperação do LSP.

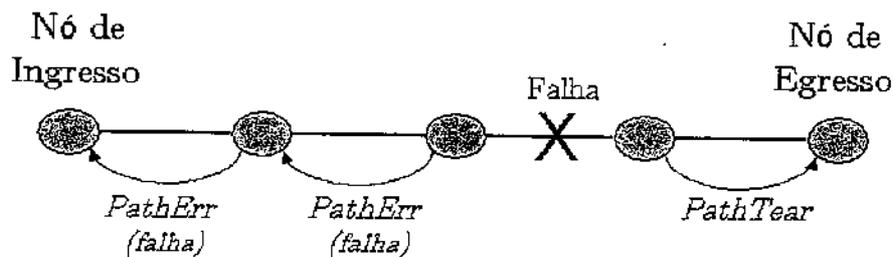


Figura B.5: Exemplo de sinalização de um LSP afetado por falha.

O re-roteamento é feito pelo nó de ingresso como se fosse uma nova requisição de LSP, após a chegada da mensagem *PathErr*.

No caso do roteamento fixo e fixo-alternado, os nós mais próximos da montante e da jusante da falha, notificam a rede da mudança de topologia, de forma que quando o nó de ingresso deve fazer o roteamento do LSP afetado pela falha, o nó já possui a nova topologia da rede.

Apêndice C

Lista de Trabalhos Publicados

C.1 Trabalhos Ligados Diretamente a Tese

1. G. S. Pavani, H. Waldman, "Traffic Engineering and Restoration in Optical Packet Switching Networks by means of Ant Colony Optimization". In *Third International Conference on Broadband Communications, Networks, and Systems* (BroadNets 2006), San Jose, CA, USA. October 2006.
2. G. S. Pavani, H. Waldman, "Grid Resource Management by means of Ant Colony Optimization". In *Third International Conference on Broadband Communications, Networks, and Systems* (BroadNets 2006), San Jose, CA, USA. October 2006.
3. G. S. Pavani, H. Waldman, "Adaptive Routing and Wavelength Assignment with Power Constraints using Ant Colony Optimization". In *International Telecommunications Symposium* (ITS 2006), Fortaleza, Brazil. September 2006.
4. G. S. Pavani, H. Waldman, "Evaluation of an Ant-based Architecture for All-Optical Networks". In *10th Conference on Optical Network Design and Modelling* (ONDM 2006), Copenhagen, Denmark. May 2006.
5. G. S. Pavani, H. Waldman, F. Callegati, A. Campi, W. Cerroni, "Adaptive Routing in Optical Packet Switching Networks using Ant Colony Optimization". In *13th International Conference on Telecommunications* (ICT 2006). Funchal, Madeira Island, Portugal. May 2006.
6. G. S. Pavani, H. Waldman, "Dynamic Routing and Wavelength Assignment with Power Constraints". In *Anais do XXI Simpósio Brasileiro de Telecomunicações* (SBT 2004), Belém, Brazil. September 2004.

C.2 Demais Trabalhos

1. F. Callegati, W. Cerroni, L. H. Bonani, F. R. Barbosa, E. Moschim, G. S. Pavani, "Congestion Resolution in Optical Burst/Packet Switching with Limited Wavelength Conversion". In *IEEE Global Communications Conference (GlobeCom 2006)*, San Francisco, USA. November/December 2006. (Aceito para publicação)
2. L. G. Zuliani, M. S. Savasini, G. S. Pavani, R. Pasquini, F. L. Verdi, M. Magalhães, "An Implementation of an OSPF-TE to Support GMPLS-controlled All-Optical WDM Networks". In *International Telecommunications Symposium (ITS 2006)*, Fortaleza, Brazil. September 2006.
3. G. S. Pavani, H. Waldman, "A Hybrid Emulation System for KyaTera Testbed". In *Proceedings of the Second e-Photon/ONE Summer School*, Cesenatico/Pisa, Italy; pp. 9-12. August/September 2005.
4. D. A. A. Mello, J. U. Pelegri, M. S. Savasini, G. S. Pavani, H. Waldman, "Inter-arrival Planning for Sub-graph Routing Protection in WDM Networks". In *Proceedings of the 11th International Conference on Telecommunications (ICT 2004)*, LNCS 3124, pp. 328-335, Springer. August 2004.
5. G. S. Pavani, H. Waldman, "Using Genetic Algorithms in Constrained Routing and Wavelength Assignment". In *Proceedings of the 8th IFIP Working Conference on Optical Network Design and Modelling (ONDM 2004)*, Ghent, Belgium; v. 1, pp. 565-584. February 2004.

Apêndice D

Lista de Acrônimos

ACO	Ant Colony Optimization
AS	Autonomous System
ASE	Amplified Spontaneous Emission
BER	Bit Error Rate
BP	Bounded Pareto
CAC	Call (or Connection) Admission Control
EDFA	Erbium Doped Fiber Amplifier
FDL	Fiber Delay Line
FSC	Fiber Switch Capable
GMPLS	Generalized Multi-Protocol Label Switching
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPv6	Internet Protocol version 6
LMP	Link Management Protocol
LOL	Loss Of Light
LSA	Link-State Advertisement
LSC	Lambda Switch Capable
LSR	Label Switching Router

LSP	Label Switched Path
OPS	Optical Packet Switching
OSPF-TE	Open Shortest Path First with Traffic Engineering
OXC	Optical Cross-Connect
PSC	Packet Switch Capable
QoS	Quality of Service
RSVP-TE	ReSource reserVation Protocol with Traffic Engineering
RWA	Routing and Wavelength Assignment
SI	Swarm Intelligence
SNR	Signal-Noise Ratio
SP	Shortest Path
SRLG	Shared Risk Link Group
TCP	Transmission Control Protocol
TDM	Time Division Multiplexing
TE	Traffic Engineering
TLV	Type-Length-Value
WDM	Wavelength Division Multiplexing
WDS	Wavelength and Delay Selection
WRN	Wavelength Routing Node
WRON	Wavelength Routed Optical Network
WRS	Wavelength Routing Switch

Apêndice E

Índice Remissivo de Autores

- Agobua, J. 1
Agrawal, G. 44
Ali, M. 41
Andriolli, N. 42
Arantes, E. A. J. 41
Ash, G. R. 103
Awduche, D. 1
- Barán, B. 21
Barr, R. S. 29
Bastos-Filho, C. J. A. 41
Beni, G. 2
Berdusco, M. 41
Berger, L. 100
Bestravos, A. 96
Bettstetter, C. 92
Binetti, S. 41
Bruten, J. 92
- Callegati, F. 69, 70
Camelo, R. 26, 27
Campelo, D. 26, 27
Carrillo, L. 92
Castoldi, P. 42
Cerroni, W. 69, 70
Chen, J. 6
Chiu, A. 41, 42
Cincotti, G. 8
Clouqueur, M. 4
Coelho, L. D. 41
Corazza, G. 69, 70
- Crovella, M. E. 96, 97
Cugini, F. 42
- Dante, R. G. 41
Datta, D. 41, 43
Deering, S. 67, 75
Deogun, J. 41
Di Caro, G. 6, 9, 11-13, 79, 93
Dijkstra, E. W. 6, 102
Dorigo, M. 2, 6, 9, 11-13, 79, 93
Druschel, P. 6
Ducatelle, F. 93
- Farrel, A. 103
Feng, H. 41, 43
Fontana, E. 41
Foster, I. 55
Fredette, A. 101
- Gambardella, L. M. 93
Garlick, R. M. 29
Glover, F. 11
Gonzáles, F. 29
Grassé, P. P. 3
Green Jr., P. 47
Grover, W. D. 4, 19, 36
Guo, M. 29
- Harchol-Balter, M. 97
Harle, D. 92
Heritage, J. 41, 43
Hinden, R. 67, 75

- Holland, O. 92
Horiguchi, S. 29
Hunter, D. K. 69

Iannone, E. 41
Iwata, A. 103

Jacobson, V. 9
Jiang, X. 29
Jue, J. 25, 26, 102

Karels, M. 9
Karmouch, A. 3
Kesselman, C. 55
Kompella, K. 20, 100

Laguna, M. 11
Lang, J. 101
Lange, D. B. 3
Leboucher, V. 41
Listanti, M. 41

Magalhães, M. 50
Malcolm, J. 1
Mannie, E. 19, 20, 99
Martins-Filho, J. F. 41
Marzo, J. L. 92
Matsumoto, M. 96
Matsuo, H. 6
McManus, J. 1
Miller, G. J. 96
Mori, K. 6
Mukherjee, B. 19, 25, 26, 30, 41, 43, 102
Murta, C. D. 97

Ngo, S. 29
Nishimura, T. 96
Nunes, F. D. 41

O'Dell, M. 1
Oliveira, J. P. G. 41
Oliveira, S. C. 41
O'Mahony, M. J. 69
Oshima, M. 3

Papadimitriou, D. 19, 20
Pasquini, R. 50

Pavani, G. S. 41, 44, 46, 50, 53
Penninckx, D. 41
Pham, V. A. 3
Prehofer, C. 92
Press, A. 92

Ramamurthy, B. 41, 43
Ramaswami, R. 26, 41, 45
Rekhter, Y. 20, 100
Rothkrantz, L. 92

Sabella, R. 41
Sahasrabuddhe, L. 19, 30
Satyanarayana, A. 103
Savasini, M. S. 50
Schoonderwoerd, R. 92
Schopf, J. M. 56, 58
Sim, K. M. 14, 88, 91
Simeonidou, D. 69
Sinclair, M. 29
Sivarajan, K. 26, 41, 45
Sosa, R. 21
Spanke, R. 47
Springer-Verlag 29
Strand, J. 41, 42
Stützle, T. 2
Subramanian, D. 6
Sun, W. H. 14, 88, 91

Thompson, K. 96
Tkach, R. 41, 42
Tzanakaki, A. 69

Valcarenghi, L. 42
Varela, G. 29
Verdi, F. L. 50
Vilà, P. 92

Waldman, H. 26, 27, 41, 44, 46, 53
Wang, J. 2, 19, 30
Wilder, R. 96

Yen, J. 102

Zang, H. 25, 26, 102
Zuliani, L. G. 50

Referências Bibliográficas

- [1] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus, "Requirements for Traffic Engineering Over MPLS," RFC 2702 (Informational), Sept. 1999. [Online]. Available: <http://www.ietf.org/rfc/rfc2702.txt>
- [2] M. Dorigo and T. Stützle, *Ant Colony Optimization*. MIT Press, 2004.
- [3] G. Beni and J. Wang, "Swarm intelligence in cellular robotic systems," in *Proceedings of the NATO Advanced Workshop on Robots and Biological Systems*, 1989.
- [4] P. P. Grassé, "La reconstruction du nid et les coordinations inter-individuelles chez *bellcoitermes natalensis* et *cubitermes* sp. la théorie de la stigmergie: Essai d'interprétation des termites constructeurs," *Insectes Sociaux*, vol. 6, pp. 41–81, 1959.
- [5] V. A. Pham and A. Karmouch, "Mobile software agents: An overview," *IEEE Communications Magazine*, vol. 36, no. 7, pp. 26–37, July 1998.
- [6] D. B. Lange and M. Oshima, "Seven good reasons for mobile agents," *Communications of the ACM*, vol. 42, no. 3, pp. 88–89, March 1999.
- [7] M. Clouqueur and W. D. Grover, "Availability analysis of span-restorable mesh networks," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 4, pp. 810–821, May 2002.
- [8] E. W. Dijkstra, "A note on two problems in connection with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.
- [9] G. Di Caro and M. Dorigo, "AntNet: distributed stigmergetic control for communications networks," *Journal of Artificial Intelligence Research*, vol. 9, pp. 317–365, 1998.
- [10] —, "Two ant colony algorithms for best-effort routing in datagram networks," in *10th International Conference on Parallel and Distributed Computing and Systems (PDCS'98)*, October 1998.
- [11] D. Subramanian, P. Druschel, and J. Chen, "Ants and reinforcement learning: A case study in routing in dynamic networks," in *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, 1997, pp. 832–839.

- [12] H. Matsuo and K. Mori, "Accelerated ants routing in dynamic networks," in *Proceedings of the Second International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, August 2001, pp. 333–339.
- [13] G. Cincotti, "Full optical encoders/decoders for photonic IP routers," *Journal of Lightwave Technology*, vol. 22, no. 2, pp. 337–342, February 2004.
- [14] V. Jacobson and M. Karels, "Congestion avoidance and control," *ACM Computer Communication Review*, vol. 18, no. 4, pp. 314–329, August 1990.
- [15] F. Glover and M. Laguna, *Tabu Search*. Kluwer Academic Publishers, 1997.
- [16] K. M. Sim and W. H. Sun, "Ant colony optimization for routing and load-balancing: survey and new directions," *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol. 33, no. 5, pp. 560–572, September 2003.
- [17] J. Wang, L. Sahasrabudde, and B. Mukherjee, "Path vs. subpath vs. link restoration for fault management in IP-over-WDM networks: Performance comparisons using GMPLS control signaling," *IEEE Communications Magazine*, vol. 40, no. 11, pp. 80–87, November 2002.
- [18] D. Papadimitriou and E. Mannie, "Analysis of Generalized Multi-Protocol Label Switching (GMPLS)-based Recovery Mechanisms (including Protection and Restoration)," RFC 4428 (Informational), Mar. 2006. [Online]. Available: <http://www.ietf.org/rfc/rfc4428.txt>
- [19] W. D. Grover, *Mesh-based Survivable Transport Networks: Options and Strategies for Optical, MPLS, SONET and ATM Networking*. Prentice Hall, 2003.
- [20] E. Mannie and D. Papadimitriou, "Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)," RFC 4427 (Informational), Mar. 2006. [Online]. Available: <http://www.ietf.org/rfc/rfc4427.txt>
- [21] K. Kompella and Y. Rekhter, "Routing Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)," RFC 4202 (Proposed Standard), Oct. 2005. [Online]. Available: <http://www.ietf.org/rfc/rfc4202.txt>
- [22] —, "OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)," RFC 4203 (Proposed Standard), Oct. 2005. [Online]. Available: <http://www.ietf.org/rfc/rfc4203.txt>
- [23] B. Barán and R. Sosa, "AntNet - routing algorithm for data networks based on mobile agents," *Revista Iberoamericana de Inteligencia Artificial*, no. 12, pp. 75–84, 2001.
- [24] H. Zang, J. Jue, and B. Mukherjee, "A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks," *Optical Networks Magazine*, vol. 1, no. 1, pp. 47–60, January 2000.

- [25] R. Ramaswami and K. Sivarajan, "Routing and wavelength assignment in all-optical networks," *IEEE/ACM Transactions on Networking*, vol. 3, no. 5, pp. 489–500, October 1995.
- [26] H. Waldman, D. Campelo, and R. Camelo, "Dynamic priority strategies for wavelength assignment in WDM rings," in *Proceedings of the Global Telecommunications Conference*, vol. 2, November 2000, pp. 1288–1292.
- [27] G. Varela and M. Sinclair, "Ant colony optimisation for virtual-wavelength-path routing and wavelength allocation," in *Proceedings of the Congress on Evolutionary Computation*, vol. 3, July 1999, pp. 1809–1816.
- [28] F. González and et al., "Lightpath routing and wavelength assignment by means of ant colony optimization," in *Proceedings of the Seventh IFIP Working Conference on Optical Network Design and Modelling (ONDM'03)*, vol. 2, February 2003, pp. 855–864.
- [29] R. M. Garlick and R. S. Barr, "Dynamic wavelength routing in WDM networks via ant colony optimization," in *Third International Workshop on Ant Algorithms - LNCS 2463*, Springer-Verlag, Ed., vol. 3, September 2002, pp. 250–255.
- [30] S. Ngo, X. Jiang, S. Horiguchi, and M. Guo, "Dynamic routing and wavelength assignment in WDM networks with ant-based agents," in *International Conference on Embedded and Ubiquitous Computing (EUC'04) - LNCS 3207*, Springer-Verlag, Ed., August 2004, pp. 829–838.
- [31] S. Ngo, X. Jiang, and S. Horiguchi, "Adaptive routing and wavelength assignment using ant-based algorithm," in *12th IEEE International Conference on Networks (ICON'04)*, vol. 2, November 2004, pp. 482–486.
- [32] R. Sabella, E. Iannone, M. Listanti, M. Berdusco, and S. Binetti, "Impact of transmission performance on path routing in all-optical transport network," *IEEE/OSA Journal of Lightwave Technology*, vol. 16, no. 11, pp. 1965–1972, November 1998.
- [33] B. Ramamurthy, D. Datta, H. Feng, J. Heritage, and B. Mukherjee, "Impact of transmission impairments on the teletraffic performance of wavelength-routed optical networks," *IEEE/OSA Journal of Lightwave Technology*, vol. 17, no. 10, pp. 1713–1723, October 1999.
- [34] M. Ali, B. Ramamurthy, and J. Deogun, "Routing and wavelength assignment with power considerations in optical networks," *Computer Networks*, vol. 32, pp. 539–555, 2000.
- [35] J. Strand, A. Chiu, and R. Tkach, "Issues for routing in the optical layer," *IEEE Communications Magazine*, vol. 39, no. 2, pp. 81–87, February 2001.
- [36] M. Ali, V. Leboucher, and D. Penninckx, "Intelligent lightpath selection schemes," in *28th European Conference on Optical Communications - ECOC'02*, September 2002.

- [37] J. F. Martins-Filho, C. J. A. Bastos-Filho, E. A. J. Arantes, S. C. Oliveira, L. D. Coelho, J. P. G. Oliveira, R. G. Dante, E. Fontana, and F. D. Nunes, "Novel routing algorithm for transparent optical networks based on noise figure and amplifier saturation," in *Proceedings of SBMO/IEEE Microwave and Optoelectronics Conference (IMOC 2003)*, vol. 2, September 2003, pp. 919–923.
- [38] G. S. Pavani and H. Waldman, "Using genetic algorithms in constrained routing and wavelength assignment," in *8th IFIP Working Conference on Optical Network Design and Modelling – ONDM'04*, vol. 1, February 2004, pp. 565–584.
- [39] R. Ramaswami and K. Sivaraajan, *Optical Networks: a Practical Perspective*, 2nd ed. Morgan Kaufmann Publishers, 2002.
- [40] M. Ali, *Transmission-efficient Design and Management of Wavelength-routed Optical Networks*. Kluwer Academics Publishers, 2001.
- [41] G. S. Pavani, "Roteamento e alocação de comprimentos de onda com restrições de potência usando algoritmos genéticos," Master's thesis, Universidade Estadual de Campinas (Unicamp), 2003.
- [42] F. Cugini, N. Andriolli, L. Valcarenghi, and P. Castoldi, "A novel signaling approach to encompass physical impairments in GMPLS networks," in *IEEE Global Telecommunications Conference (Globecom 2004)*, 2004, pp. 369–373.
- [43] G. Agrawal, *Fiber-Optic Communication Systems*, 3rd ed. Wiley Interscience, 2002.
- [44] R. Spanke, "Architectures for guidedwave optical space switching systems," *IEEE Communications Magazine*, vol. 25, no. 5, pp. 42–48, May 1987.
- [45] P. Green Jr., *Optical Fiber Communications*. Prentice Hall, 1993.
- [46] L. G. Zuliani, M. S. Savasini, G. S. Pavani, R. Pasquini, F. L. Verdi, and M. Magalhães, "An implementation of an OSPF-TE to support GMPLS-controlled all-optical WDM networks," in *Proceedings of the International Telecommunications Symposium (ITS 2006)*, September 2006.
- [47] I. Foster and C. Kesselman, Eds., *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers, 1998.
- [48] J. M. Schopf, *Grid resource management: state of the art and future trends*. Kluwer Academic Publishers, 2004, ch. Ten actions when Grid scheduling: the user as a Grid scheduler, pp. 15–23.
- [49] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification," RFC 2460 (Draft Standard), Dec. 1998. [Online]. Available: <http://www.ietf.org/rfc/rfc2460.txt>

- [50] M. J. O'Mahony, D. Simeonidou, D. K. Hunter, and A. Tzanakaki, "The application of optical packet switching in future communication networks," *IEEE Communications Magazine*, vol. 39, no. 3, pp. 128–135, March 2001.
- [51] F. Callegati, W. Cerroni, and G. Corazza, "Optimization of wavelength allocation in WDM optical buffers," *Optical Networks Magazine*, vol. 2, no. 6, pp. 66–72, November 2001.
- [52] L. Carrillo, J. L. Marzo, D. Harle, and P. Vilà, "A review of scalability and its application in the evaluation of the scalability measure of AntNet routing," in *Proceedings of IASTED Communication Systems and Networks (CSN 2003)*, A. Press, Ed., September 2003, pp. 317–323.
- [53] C. Prehofer and C. Bettstetter, "Self-organization in communication networks: principles and design paradigms," *IEEE Communications Magazine*, vol. 43, no. 7, pp. 78–85, July 2005.
- [54] R. Schoonderwoerd, O. Holland, J. Bruten, and L. Rothkrantz, "Ant-based load balancing in telecommunications networks," *Adaptive Behaviour*, vol. 5, no. 2, pp. 169–207, 1997.
- [55] G. Di Caro, F. Ducatelle, and L. M. Gambardella, "AntHocNet: An adaptive nature-inspired algorithm for routing in mobile ad hoc networks," *European Transactions on Telecommunications, Special Issue on Self-organization in Mobile Networking*, vol. 16, no. 5, pp. 443–455, 2005.
- [56] M. Matsumoto and T. Nishimura, "Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Transactions on Modeling and Computer Simulation*, vol. 8, no. 1, pp. 3–30, January 1998.
- [57] K. Thompson, G. J. Miller, and R. Wilder, "Wide-area internet traffic patterns and characteristics," *IEEE Network*, vol. 11, no. 6, pp. 10–23, November/December 1997.
- [58] M. E. Crovella and A. Bestavros, "Self-similarity in world wide web traffic: evidence and possible causes," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 835–846, 1997.
- [59] M. Harchol-Balter, M. E. Crovella, and C. D. Murta, "On choosing a task assignment policy for a distributed server system," *Journal of Parallel and Distributed Computing*, vol. 59, no. 2, pp. 204–228, November 1999.
- [60] E. Mannie, "Generalized Multi-Protocol Label Switching (GMPLS) Architecture," RFC 3945 (Proposed Standard), Oct. 2004. [Online]. Available: <http://www.ietf.org/rfc/rfc3945.txt>
- [61] L. Berger, "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions," RFC 3473 (Proposed Standard), Jan. 2003, updated by RFC 4003. [Online]. Available: <http://www.ietf.org/rfc/rfc3473.txt>

- [62] J. Lang, "Link Management Protocol (LMP)," RFC 4204 (Proposed Standard), Oct. 2005. [Online]. Available: <http://www.ietf.org/rfc/rfc4204.txt>
- [63] A. Fredette and J. Lang, "Link Management Protocol (LMP) for Dense Wavelength Division Multiplexing (DWDM) Optical Line Systems," RFC 4209 (Proposed Standard), Oct. 2005. [Online]. Available: <http://www.ietf.org/rfc/rfc4209.txt>
- [64] J. Yen, "Finding the k shortest loopless paths in a network," *Management Science*, vol. 17, no. 11, pp. 712–716, July 1971.
- [65] A. Farrel, A. Satyanarayana, A. Iwata, and G. R. Ash, "Crankback Signaling Extensions for MPLS and GMPLS RSVP-TE," Internet Draft, May 2005. [Online]. Available: <http://www.ietf.org/internet-drafts/draft-ietf-ccamp-crankback-05.txt>