



UNIVERSIDADE ESTADUAL DO CEARÁ - UECE
MESTRADO ACADÊMICO EM CIÊNCIA DA COMPUTAÇÃO

ERICK AGUIAR DONATO

**ROTEAMENTO E ATRIBUIÇÃO DO COMPRIMENTO DE ONDA
DINÂMICOS UTILIZANDO COLÔNIA DE FORMIGAS EM REDES
ÓPTICAS COMUTADAS POR RAJADAS**

FORTALEZA - CEARÁ
2010

ERICK AGUIAR DONATO

**ROTEAMENTO E ATRIBUIÇÃO DO COMPRIMENTO DE ONDA
DINÂMICOS UTILIZANDO COLÔNIA DE FORMIGAS EM REDES
ÓPTICAS COMUTADAS POR RAJADAS**

Dissertação à Coordenação do Curso de Mestrado em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará, como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Joaquim Celestino Júnior.

**FORTALEZA - CEARÁ
2010**

ERICK AGUIAR DONATO

**ROTEAMENTO E ATRIBUIÇÃO DO COMPRIMENTO DE ONDA
DINÂMICOS UTILIZANDO COLÔNIA DE FORMIGAS EM REDES
ÓPTICAS COMUTADAS POR RAJADAS**

Dissertação à Coordenação do Curso de Mestrado em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará, como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação.

Aprovada em: __/__/____

Banca Examinadora

Prof. Dr. Joaquim Celestino Júnior – UECE
(Orientador)

Prof. Dr. Anilton Salles Garcia - UFES
(Membro externo)

Prof. Dr. Gustavo Augusto Lima de Campus - UECE
(Membro)

Prof. Dr. Jorge Luiz Castro e Silva - UECE
(Membro)

Dedico este trabalho à minha esposa, pela paciência e apoio dedicados.

AGRADECIMENTOS

Aos colegas Alyson, Antoniel, Fabio, Lourival, Paz Veras e Walison, pelas viagens, pela convivência, pelo apoio, pelos conhecimentos compartilhados.

Aos colegas Neuton e Sergio Vieira, pela contribuição nas idéias e na implementação do meu trabalho.

Ao professor Celestino, pela atenção, pela compreensão, pela intensa dedicação e empenho na orientação do meu trabalho, dando conselhos e apoio em todos os momentos.

Aos demais alunos, professores e técnicos do curso, que contribuíram ou participaram da elaboração e conclusão do meu trabalho.

RESUMO

Muitos trabalhos mostram o potencial das fibras ópticas e, conseqüentemente, das redes ópticas. A grande capacidade de transmissão de dados e baixa taxa de erros são características de uma fibra óptica. Portanto, para um bom desempenho de uma rede óptica, é necessário que seus recursos sejam tratados adequadamente. Esta dissertação trata de um dos principais problemas das redes ópticas: o roteamento e atribuição do comprimento de onda (RWA). Este define as rotas e comprimentos de onda a serem utilizados. A solução adequada desse problema evita que requisições feitas sejam rejeitadas por falta de recurso da rede. Estudos mostram que a comutação por rajadas é uma alternativa viável para contornar o problema de desperdício de recursos de um circuito estabelecido. Além disso, evita que os dados passem para o domínio eletrônico tornando a comutação puramente óptica. Dentro de uma colônia de formigas, como em outros tipos de insetos, a atividade dos indivíduos, bem como da sociedade como um todo, não são reguladas por uma entidade centralizadora. Portanto, possuem um comportamento adaptativo e robusto. Este comportamento global é resultado da auto-organização dinâmica feita pelas interações locais e pelas comunicações entre os indivíduos. É proposto uma nova abordagem para solucionar o RWA em redes ópticas comutadas por rajada (OBS) utilizando colônia de formigas (ACO). O AntOBS é uma nova abordagem para o roteamento e atribuição do comprimento de onda. Este é inspirado no comportamento natural e coletivo da colônia de formigas. A dinamicidade e a auto-organização são as principais características do AntOBS. Esse comportamento leva a diminuição da probabilidade de bloqueio das requisições da rede. Vários experimentos foram realizados usando o AntOBS para o roteamento. Os resultados do AntOBS foram satisfatórios, o que mostra a viabilidade da aplicação do algoritmo no problema de RWA em redes OBS.

Palavras-chave: Redes Ópticas. Colônia de Formigas. Comutação por Rajadas. RWA.

ABSTRACT

Many studies show the potential of optical fibers and hence the optical networks. The large capacity of data transmission and low error rate are characteristic of an optical fiber. Therefore, for the proper performance of an optical network, it is necessary that your assets are handled appropriately. This dissertation is a major concern for optical networks: routing and wavelength assignment (RWA). This defines the routes and wavelengths to be used. The proper solution of this problem prevents requests made are rejected for lack of network resources. Studies show that the burst switching is a viable alternative to circumvent the problem of wasting resources on a circuit established. In addition, prevents data to pass to the electronic field making purely optical switching. Within a colony of ants, as in other types of insects, the activity of individuals and society as a whole, are not governed by a centralized entity. So, have a robust and adaptive behavior. This global behavior is the result of self-organization dynamics made by local interactions and communications among individuals. We propose a new approach to solve the RWA in optical networks burst switched (OBS) using ant colony (ACO). The AntOBS is a new approach for routing and wavelength assignment. This is inspired by nature and collective behavior of ant colony. The dynamics and self-organization are the main features of AntOBS. This behavior leads to reduction in blocking probability of requests from the network. Several experiments were performed using AntOBS for routing. The results of AntOBS were satisfactory, which shows the feasibility of applying the algorithm on the problem of RWA in OBS networks.

Keywords: Optical Networks. Ant Colony. Optical Burst Switching. RWA.

LISTA DE FIGURAS

Figura 1 - Experimento com dois caminhos iguais	22
Figura 2 - Experimento com caminhos diferentes	23
Figura 3 - Experimento de adicionar um novo caminho	24
Figura 4 – Exemplo de rede óptica	30
Figura 5 - Evolução das arquiteturas da integração IP/Óptico	30
Figura 6 - Comutação por Circuitos	32
Figura 7 - Comutação por Pacotes	33
Figura 8 - Comutação por rajadas	35
Figura 9 - Funcionamento do JIT	36
Figura 10 – Funcionamento do JET	37
Figura 11 – Cenário para o problema de recuperação	42
Figura 12 – Comportamento da formiga IAnt	50
Figura 13 – Comportamento da formiga MAnt	51
Figura 14 – Exemplo do envio da rajada e reserva dos recursos	55
Figura 15 – Topologia da rede de teste 1	56
Figura 16 – Topologia 2 (NSFNET)	56
Figura 17 – Topologia utilizada no exemplo da seção	56
Figura 18 – Gráfico da Probabilidade de Bloqueio x Carga no Cenário 1 da Topologia 1	62
Figura 19 – Gráfico da Probabilidade de Bloqueio x Carga no Cenário 2 da Topologia 1	63
Figura 20 – Gráfico da Probabilidade de Bloqueio x Carga no Cenário 1 da Topologia 2	64
Figura 21 – Gráfico da Probabilidade de Bloqueio x Carga no Cenário 2 da Topologia 2	64

LISTA DE TABELAS

Tabela 1 - Tabela de roteamento com nível de feromônio	18
Tabela 2 - Tabela de rotas e comprimentos de onda disponíveis do nó atual para o nó destino	18
Tabela 3 - Tabela de Roteamento/Feromônio	47
Tabela 4 - Tabelas de Roteamento iniciais	57
Tabela 5 - Tabelas de Roteamento para a movimentação das formigas IAnt ...	57
Tabela 6 - Tabelas de Roteamento para a movimentação da formiga MAnt	58
Tabela 7 - Tabelas de Roteamento para a movimentação do BCP	59
Tabela 8 – Parâmetros do algoritmo AntOBS	60

LISTA DE ABREVIATURAS E SIGLAS

ACO – *Ant Colony Optimization*;

ANN – *Artificial Neural Networks*;

AS – *Ant System*;

AG – *Algoritmos Genéticos*;

CWDM – *Coarse Wavelength Division Multiplexing*;

DLE - *Dynamic Lightpath Establishment*;

DWDM – *Dense Wavelength Division Multiplexing*;

JET – *Just-Enough-Time*;

JIT – *Just-in-Time*;

IP – *Internet Protocol*;

OBS – *Optical Bursts Switching*;

OCS – *Optical Circuit Switching*;

OPS – *Optical Packet Switching*;

OXC – *Optical Crossconnect*;

QoS – *Quality of Service*;

SLE - *Static Lightpath Establishment*;

TTL – *Time-to-Live*;

RWA – *Routing and Wavelength Assignment*;

WDM – *Wavelength Division Multiplexing*;

SUMÁRIO

LISTA DE FIGURAS.....	7
LISTA DE TABELAS.....	8
LISTA DE ABREVIATURAS E SIGLAS.....	9
1 INTRODUÇÃO.....	13
1.1 Redes Ópticas.....	14
1.2 Roteamento e Atribuição do Comprimento de Onda.....	15
1.3 Otimização por Colônia de Formigas.....	15
1.4 Objetivo.....	16
1.5 Estrutura do Trabalho.....	16
2 Trabalhos Relacionados.....	17
3 OTIMIZAÇÃO POR COLÔNIA DE FORMIGAS.....	21
3.1 <i>Swarm Intelligence</i>	21
3.2 Introdução à ACO	22
3.3 Experimento das Duas Pontes	23
3.4 Modelo Artificial	25
3.4.1 <i>Ant System</i>	26
3.5 Mecanismos de Funcionamento	27
3.6 Algoritmo ACO para Escolha do Menor Caminho	27
4 REDES ÓPTICAS.....	31
4.1 Comutação em Redes Ópticas.....	33
4.1.1 Comutação por Circuitos.....	33

4.1.2 Comutação por Pacotes.....	34
4.1.3 Comutação por Rajadas.....	35
4.2 Protocolos de Sinalização.....	37
4.2.1 <i>Just-inTime</i> (JIT).....	37
4.2.2 <i>Just-Enough-Time</i> (JET).....	38
4.3 Roteamento e Atribuição do Comprimento de Onda (RWA).....	39
4.3.1 Roteamento.....	41
4.3.2 Atribuição do Comprimento de Onda.....	42
4.3.3 Solução Conjunta do RWA.....	43
4.3.4 Problema de Recuperação.....	43
5 ANTOBS.....	46
5.1 Modelo Utilizado.....	48
5.1.1 A Estrutura das Formigas.....	48
5.1.2 Escolha do Próximo Nó.....	49
5.1.3 Atualização do Feromônio.....	49
5.1.4 Evaporação.....	50
5.2 Detalhes do Protocolo AntOBS.....	50
5.3 Criação e Manutenção das Rotas.....	51
5.3.1 Fase de Inicialização.....	51
5.3.2 Fase de Manutenção.....	52
5.3.3 Reserva dos Recursos e Envio da Rajada.....	54
5.4 Exemplo de Funcionamento do Protocolo.....	56
6 SIMULAÇÕES E RESULTADOS	60

6.1 Topologia 1	62
6.2 Topologia 2 (NSFNET)	63
6.3 <i>Overhead</i>	65
7 CONCLUSÃO E TRABALHOS FUTUROS	66
7.1 Conclusão	66
7.2 Trabalhos Futuros	67
REFERÊNCIAS BIBLIOGRÁFICA	68

Capítulo 1

Introdução

A transmissão de informação através de fibras ópticas tem crescido e amadurecido bastante nas últimas décadas, se tornando, em termos de meio de transmissão, a opção mais atraente nas redes de alta velocidade. Inclusive, a evolução das tecnologias empregadas nesta categoria de redes deve-se, em parte, à utilização das fibras ópticas. Muitas tecnologias surgiram com o objetivo de otimizar seu uso nesse meio de transmissão, objetivando usufruir de suas características, como por exemplo, capacidade de transmissão, confiabilidade, distâncias alcançadas, etc. De acordo com Silveira (2003), as fibras ópticas impactaram na classificação das redes em locais, metropolitanas e de longa distância. Onde tal classificação, leva em consideração a distância geográfica alcançada pela rede. Tecnologias criadas na década de 70 e 80, como Frame Relay e ATM, utilizam fibras ópticas como principal meio de transmissão.

Juntamente, existe também uma evolução nas aplicações usadas nas redes. A internet é um exemplo disso, que teve um crescimento mundial do número de usuários na ordem de 399,3% entre 2000 a 2009 (INFORMAÇÕES ESTATÍSTICAS, 2010). Como novas aplicações sempre estão aparecendo, a necessidade de meios de comunicação para suprir a necessidade das aplicações dos usuários motiva a evolução das redes.

As necessidades dos usuários aumentam em quantidade e qualidade. Algumas aplicações necessitam de determinados requisitos mínimos para ter um bom funcionamento, o que nem sempre a internet pode fornecer. Como a internet é uma rede de datagramas de melhor esforço, não pode oferecer nenhum tipo de garantia às aplicações dos usuários. Aplicações multimídia são exemplos de aplicações que precisam que alguns recursos da rede sejam garantidos para terem uma perfeita execução. Chamamos essa garantia de níveis de recursos de Qualidade de Serviço (QoS – *Quality of Service*).

As redes ópticas possuem características que atendem as necessidades de QoS das aplicações. Um exemplo dessas características é a alta taxa de transmissão de dados. Estudos indicam que essas redes conseguem, com a multiplexação por comprimento de onda, ter uma transmissão na ordem de 10 Tb/s (Redes de Alta Velocidade).

Essa alta taxa de transmissão é conseguida, entre outros fatores, devido à utilização da fibra óptica, que tem a sílica como principal material para sua fabricação. Isso se deve à principal vantagem da fibra óptica que é a grande capacidade de largura de banda. Outras vantagens são a maior segurança dos dados, baixa interferência entre os canais, a imunidade a interferências eletromagnéticas (Sivarajan et al, 2002).

1.1 Redes Ópticas

A primeira geração das redes ópticas era conhecida como redes opacas. Todas as operações eram feitas no domínio eletrônico, como por exemplo, a comutação ou qualquer operação de gerência. Essa conversão para o domínio eletrônico é conhecida como conversão O-E-O (Óptico-Elétrico-Óptico) e é feita por *transponders*. Esse tipo de rede possui a vantagem de facilmente converter comprimentos de ondas e regenerar o sinal. Porém, possui a desvantagem de tornar a rede não transparente aos protocolos utilizados, além do aumento do custo pela utilização dos *transponders* (Zhang et al, 2001; Strand et al, 2001).

A segunda geração é conhecida como redes ópticas transparentes, ou totalmente ópticas (*all-optical network*) (Sivarajan et al, 2002). Nesse tipo de rede, as operações, como roteamento e comutação, são realizadas no meio óptico, sem a necessidade de conversão para o meio eletrônico. Conseqüentemente, não necessitará dos *transponders* para levar os dados para o meio eletrônico. Isso leva à diminuição dos custos e ao aumento do desempenho, além de evitar o gargalo gerado pelas transformações O-E-O.

Uma alternativa para aumentar a capacidade de transmissão da rede é a multiplexação por comprimento de onda (WDM - Wavelength Division Multiplexing). WDM é uma tecnologia que permite enviar vários sinais numa mesma transmissão. Na origem, vários sinais são multiplexados em comprimentos de onda diferentes. E quando chegam ao destino, são demultiplexados e transformados para o estado inicial. Essa tecnologia aumenta a utilização e o desempenho da rede.

Para ampliar os benefícios das redes ópticas é necessário evitar as conversões para o meio eletrônico, possibilitando que os dados sejam transmitidos totalmente no domínio óptico, ou seja, fazer a comutação puramente óptica. Uma importante característica da comutação totalmente óptica é a propriedade de continuidade de comprimento de onda. Essa propriedade força a utilização de um mesmo comprimento de onda em todo o caminho da conexão. Isso é imprescindível devido à inexistência de um conversor de comprimento de onda no meio óptico que possua um custo adequado.

Existem três tipos de comutação em redes ópticas bem exploradas na literatura (Mukherjee, 2006; Yoo et al, 1997; Wei et al, 2000): comutação por circuito (OCS - *Optical Circuit Switching*), comutação por pacote (OPS - *Optical Packet Switching*) e comutação por rajada (OBS - *Optical Burst Switching*).

Redes OBS possuem características que contornam os problemas gerados pelos outros dois tipos de comutação. Esses problemas serão detalhados no capítulo 4. Uma das principais vantagens das redes OBS é a separação do plano de dados do plano de controle permitindo que as informações de controle não interfiram no tráfego de rede. Protocolos de sinalização são responsáveis por fazer a reserva dos recursos para a requisição. Por essa ser considerada a melhor opção entre as comutações ópticas, essa dissertação trabalha no cenário de redes OBS. Os detalhes sobre esse assunto serão abordados no capítulo 4.

Além da comutação, a forma, o roteamento e a atribuição do comprimento de onda afetam o desempenho de rede óptica. Esse problema é um dos principais desafios em redes ópticas. A próxima seção descreve esse problema rapidamente e este é detalhado no capítulo 4.

1.2 Roteamento e Atribuição do Comprimento de Onda

Juntamente com as altas taxas de transmissão, a rede óptica deve apresentar a capacidade de tratar falhas, possuir algoritmos de roteamento que levem em consideração o estado atual da rede e achem o melhor caminho entre os nós de origem e de destino. Além disso, tem que possuir algoritmos de atribuição do comprimento de onda que devem elevar a utilização dos recursos da rede minimizando a probabilidade de bloqueio. Esse bloqueio acontece quando a rede não possui recursos suficientes para suprir as necessidades das requisições.

Os problemas de roteamento e atribuição do comprimento de onda são problemas clássicos na área das redes ópticas e a junção dos dois é conhecida como RWA (*Routing and Wavelength Assignment*) (Sivarajan et al, 2002; Zang et al, 2000). O problema de RWA tem o objetivo de escolher o caminho entre dois nós da rede e definir o comprimento de onda a ser utilizado na comunicação. Vários caminhos são possíveis entre dois nós. Porém, o algoritmo de roteamento deve ser capaz de escolher uma rota que leve em consideração o estado atual da rede, ou seja, deve observar o estado e a carga dos enlaces e dos nós da rede. O algoritmo de atribuição do comprimento de onda deve ser capaz de escolher um comprimento de onda de forma a minimizar a probabilidade de bloqueio da requisição. A escolha do comprimento de onda deve elevar o grau de utilização dos recursos da rede.

O RWA é um dos principais problemas das redes ópticas. O desempenho dos algoritmos de RWA compromete diretamente o desempenho dessas redes. Uma escolha equivocada de uma rota ou de um comprimento de onda pode gerar atrasos e perdas de grande quantidade de informações que trafegam na rede.

Algumas métricas utilizadas por algoritmos de roteamento para calcular as melhores rotas podem ser o número de saltos ou a carga de utilização dos comprimentos de onda dos enlaces. No entanto, essas métricas podem não ser as melhores alternativas para solucionar esse problema.

Solucionar satisfatoriamente o problema descrito anteriormente motivou a elaboração desse trabalho. Essa dissertação tem o objetivo de propor um novo algoritmo de RWA dinâmico utilizando a técnica de Otimização por Colônia de Formigas (ACO – *Ant Colony Optimization*) para redes ópticas de comutação por rajada e comparar o desempenho do mesmo com os algoritmos atuais. A técnica ACO é descrita rapidamente na próxima seção, sendo detalhada no capítulo 3.

1.3 Otimização por Colônia de Formigas

Ant Colony Optimization (ACO) é uma técnica inspirada no comportamento coletivo das formigas, sendo utilizada para resolver problemas de otimização. No contexto de redes de computadores, ACO usa os agentes inteligentes para selecionar os melhores caminhos na rede de acordo com o estado atual da mesma. Portanto, é uma técnica dinâmica que se adapta às possíveis modificações. É baseada em uma equação de indução que leva em consideração o acúmulo de um hormônio deixado pelos agentes no caminho percorrido. Esse hormônio é chamado de feromônio (Dorigo et al 1998).

ACO é um sistema bio-inspirado que é baseado no comportamento social das formigas. A estrutura do algoritmo que representa o funcionamento da colônia de formigas é simples. A organização da colônia e a forma como é feita a comunicação interna das formigas torna essa meta-heurística funcional e apropriada para achar a melhor solução do problema. As formigas percorrem os caminhos do “ninho” até a “fonte de alimento” e depositam neles o feromônio. Dessa forma, possibilita que outras sejam induzidas aos caminhos que possuem a maior quantidade do hormônio. Essa indução é probabilística, o que propicia a diversidade na escolha do caminho pela formiga. Esse cenário permite que a meta-heurística consiga convergir para as melhores soluções dentro de um conjunto de alternativas viáveis.

1.4 Objetivo

De acordo com as pesquisas feitas durante a fase inicial desse trabalho, o problema de RWA em redes OBS foi escolhido como tema dessa dissertação. O algoritmo AntOBS trata esse problema através da aplicação da técnica de ACO. E, portanto, usufrui de todos os benefícios trazidos por tal utilização.

O objetivo dessa dissertação é utilizar colônia de formigas para alimentar e manter as tabelas de roteamento de forma dinâmica. As formigas irão percorrer a rede atualizando as tabelas de roteamento utilizando um canal fora da banda (*out-of-band*), evitando o aumento do tráfego no plano de dados. O problema do RWA será tratado levando em consideração as informações difundidas pelas formigas na rede. Essa abordagem traz a vantagem de diminuir a probabilidade de bloqueio devido à utilização de informações, levadas pelas formigas, que representam o estado atual da rede.

Podem ser feitos vários percursos entre um nó origem e um nó destino. Ao ocorrer algum problema num determinado enlace, apenas as formigas que utilizarem os caminhos que não passem pelo enlace com problemas conseguirão atualizar as tabelas de roteamento. Isso acarreta uma mudança do estado do roteamento da rede. Portanto, a movimentação das formigas levará ao melhor caminho dinamicamente.

1.5 Estrutura do trabalho

Essa dissertação possui a seguinte organização: o próximo capítulo descreve as principais características da meta-heurística baseada em Colônia de Formigas; no capítulo 3, será exposto um resumo sobre redes ópticas, esclarecendo os principais tópicos, como por exemplo, os tipos de comutação existentes; o capítulo 4 explica, em detalhes, o problema de RWA, suas variações e principais características; no capítulo 5, será apresentado o AntOBS, a nova abordagem do problema RWA utilizando ACO; o capítulo 6 mostra os resultados das simulações feitas comparando as diferentes abordagens do problema RWA; o capítulo 7 finaliza a dissertação com a conclusão e os trabalhos futuros.

Capítulo 2

Trabalhos Relacionados

Muitas pesquisas vêm aplicando técnicas de inteligência computacional para solucionar o problema de roteamento em redes de computadores. Nesse caso, seria possível considerar diversas características da rede como, por exemplo, ruídos, ocupação e alterações de topologia, de forma mais eficiente que os algoritmos utilizados atualmente sendo assim possível atender as requisições dos usuários. Entre essas técnicas aplicadas no contexto de roteamento de redes de computadores, podem ser citadas: Redes Neurais Artificiais (ANN – *Artificial Neural Networks*) (Bastos Filho et al, 2007), Algoritmos Genéticos (GA – *Genetic Algorithms*) (Bisbal, 2004), Otimização por Colônias de Formigas (ACO – *Ant Colony Optimization*) (Dorigo et al 1998; Ngo et al, 2004; Pavani et al, 2008), Otimização por Enxames de Partículas (PSO – *Particle Swarm Optimization*) (Mohammed et al, 2008) e algoritmos híbridos (Bastos Filho et al, 2008) que combinam técnicas diferentes na mesma solução. No contexto de redes ópticas foram encontrados alguns trabalhos sobre RWA e recuperação de rajadas juntamente com ACO, quando Shi et al (2005) fizeram o primeiro trabalho nessa área.

No trabalho citado anteriormente, ACO foi utilizado para resolver os problemas do RWA e recuperação dinamicamente. Nesse trabalho, o nó de origem envia o pacote de controle (*Burst Control Packet - BCP*) para o destino com o objetivo de reservar os recursos para a rajada que virá em seguida. Inicialmente, como os níveis de feromônio (tabelas de roteamento) não estarão iniciados, cada salto do caminho será escolhido randomicamente. Quando o nó de destino receber o pacote de controle (BCP), ou seja, quando as reservas para a rajada estiverem efetivadas, o destinatário responde ao nó de origem com uma mensagem que percorrerá o caminho inverso, atualizando os níveis de feromônio dos nós intermediários. A partir disso, todos os outros pacotes de controle levarão em consideração o nível de feromônio para a escolha do caminho até o destinatário. O nível de feromônio nas tabelas reduz com o tempo.

Para conseguir isso, os autores do artigo criaram dois tipos de pacote de controle: BCP-REQ e BCP-ACK. O primeiro reserva os recursos no caminho da origem para o destino. O segundo retorna do destino para a origem atualizando os níveis de feromônio dos computadores intermediários. O BCP-ACK percorre o mesmo caminho do BCP-REQ, mudando apenas o sentido.

Essa solução traz pouco *overhead* e, de acordo com as simulações feitas no trabalho, resolve satisfatoriamente o problema. Porém, nos casos em que a rede ficar inativa ou sem enviar rajadas com sucesso, embora que por alguns instantes, isto levará a uma situação em que os níveis de feromônio poderão não representar o estado atual da rede. Nessa solução, as tabelas de roteamento somente estarão bem configuradas se houver tráfego na rede, já que é o próprio pacote de controle que faz o papel de configuração. Sendo necessária, portanto, a existência de rajadas para o bom

funcionamento e configuração da rede óptica. Se as tabelas de roteamento não estiverem próximas da configuração ideal, muitas rajadas podem ser perdidas.

A proposta desta dissertação não possui esse problema. As formigas não têm ligação com a rajada e com o pacote de controle, ou seja, não depende da existência de rajadas para as tabelas de roteamento estarem configuradas. As formigas são responsáveis por configurar as rotas e são independentes do plano de dados e do pacote de controle.

Outra vantagem dessa dissertação em relação ao trabalho anterior é que o mesmo usa ACO somente para achar o melhor caminho. A atribuição do comprimento de onda é feita randomicamente. Triay e Cervell'ó-Pastor (2010) atualizaram o trabalho anterior e utilizaram colônia de formigas para resolver o problema da atribuição do comprimento de onda.

Em Ngo et al (2006) é apresentada uma solução para os problemas de roteamento e atribuição de comprimento de onda dinâmicos em redes ópticas WDM utilizando ACO. Nesse trabalho, as formigas atuam separadamente do pacote de controle e alimentam dois tipos de tabelas: uma armazena as rotas completas nos nós de borda e outra tabela armazena os dados de roteamento nos nós de núcleo que possuem os níveis de feromônio. As Tabelas 1 e 2 abaixo exemplificam a tabela de roteamento com feromônio e a tabela de rotas respectivamente.

Destino/Vizinho	1	4	5
0	0.6	0.3	0.1
1	0.8	0.2	0.0
...
5	0.0	0.2	0.8

TABELA 1: Tabela de roteamento com nível de feromônio

Ordem	Caminho	Saltos	λ Disponíveis
1	2-0	2	λ_1, λ_2
2	4-2-0	3	λ_2, λ_3
3	3-1-0	3	λ_1, λ_3
...			

TABELA 2: Tabela de rotas e comprimentos de onda disponíveis do nó atual para o nó destino

Essa dissertação difere do trabalho acima em alguns aspectos. O primeiro deles é que o AntOBS utiliza apenas uma tabela de roteamento. Outra diferença é que o AntOBS armazena e considera o número de colisões no nó para fazer a atualização do feromônio.

Huo et al (2004) trazem uma proposta de solução para o problema da recuperação em redes ópticas. Nesse caso, o trabalho propõe uma solução baseada no tempo de *offset* para melhorar o desempenho nas situações de recuperação da rede óptica.

Em Hsu et al (2002) é feita uma análise do desempenho do roteamento por deflexão em redes OBS baseada no protocolo de sinalização JET. No entanto, o trabalho evidencia a utilização de *buffers* (FDL) no caso de deflexão, ou seja, sempre que houver bloqueio de recursos. A opção de usar *buffers* será evitada em nosso trabalho, devido ao alto custo. Essa dissertação não utiliza *buffers* e tem o objetivo de diminuir a probabilidade de bloqueio através da utilização de ACO.

Yang e Rouskas (2006) apresentam a estratégia chamada de caminho adaptativo em redes OBS, e tem por objetivo diminuir a probabilidade de bloqueio. A idéia assume que cada nó fonte possui uma lista de caminhos alternativos para cada destino e, juntamente com isso, usa informações recentes do estado da rede, como congestionamento da rede, para ordenar os caminhos. As informações são passadas pelo caminho que tiver o menor congestionamento no momento. Várias estratégias de escolha de caminho para diferentes informações da rede são mostradas e analisadas.

Chin et al (2006) apresentou um trabalho que utiliza ACO para a resolução de disputas em redes ópticas WDM comutada por pacote (OPS). Neste caso, o trabalho também utiliza um mínimo número de linhas de atraso de fibras (FDL). Embora que pequena, a utilização de FDL deve ser evitada. Esse trabalho trata de disputas de pacotes, unidades menores que rajadas. ACO é utilizada para a escolha da porta de saída adequada no momento da ocorrência da disputa, evitando ainda que torne o atraso fim-a-fim grande demais. Ainda levando em consideração as redes OPS, (Pavani et al, 2006) apresenta um algoritmo baseado em colônia de formigas que suporta engenharia de tráfego e processos de recuperação. Porém, neste trabalho, os cenários de implementação possuem capacidade completa de conversão do comprimento de onda, ou seja, todos os nós são capazes de converter o comprimento de onda dos sinais. Apesar desta dissertação está no contexto das redes OBS, este trabalho é importante para desafios futuros quando as redes OPS e os buffers forem uma realidade.

Em Carvalho (2008) foi proposto um novo algoritmo de roteamento para aplicá-lo ao problema do RWA em redes ópticas baseado em otimização por colônia de formigas. Além disso, esse algoritmo leva em consideração diversos efeitos da camada física. Nesse contexto, alguns parâmetros são utilizados para configurar o algoritmo. Para achar os melhores valores desses parâmetros, o autor utilizou uma técnica baseada em Inteligência Computacional de Enxames (*Computational Swarm Intelligence - CSI*) chamada de Otimização por Enxame de Partículas (*Particle Swarm Optimization - PSO*). Além deste trabalho não ter o foco em redes OBS, a abordagem centralizada é a principal diferença entre este trabalho e a dissertação atual. Outra diferença é que esta dissertação não utilizou técnica alguma para achar os valores dos parâmetros. Para isso, os valores dos parâmetros foram achados empiricamente, ou seja, levando em consideração o resultado das simulações efetuadas. Isso foi feito para evitar o *overhead* causado pelo cálculo excessivo dos mesmos. Isso poderia diminuir o desempenho da rede.

A quantidade de trabalhos citados ilustra a viabilidade da solução do problema RWA dinâmico juntamente com ACO. No entanto, alguns desses trabalhos utilizam conversores de comprimento de onda em cada nó da rede sem levar em consideração o custo da utilização de tais recursos. Dessa forma, a ausência de conversores impõe a necessidade de aplicação de algoritmos para calcular o próximo salto para o destino dos dados. Logo, existe a necessidade de algoritmos totalmente distribuídos capazes de lidar

com a dinâmica do futuro das redes ópticas, sendo essa a motivação principal desta dissertação.

Capítulo 3

Otimização por Colônia de Formigas

3.1 *Swarm Intelligence*

De acordo com Bonabeau et al (1999), *Swarm Intelligence* é uma técnica baseada no comportamento coletivo de sistemas descentralizados e auto-organizáveis. Normalmente, os agentes da população agem localmente no ambiente seguindo um determinado conjunto de regras de forma que, interagindo entre si, carregam e utilizam informações deixadas por outros agentes. Mesmo não havendo nenhuma forma centralizada de controle, o comportamento local dos agentes faz surgir uma inteligência global. O resultado global do processo é desconhecido pelos agentes que agem apenas localmente.

Algumas sociedades de insetos são tomadas como base para técnicas de *Swarm Intelligence*, como formigas, abelhas e vespas. Além de insetos, outras sociedades de animais também servem como base, por exemplo, bando de pássaros e cardume de peixes.

Um dos principais atrativos dessas técnicas baseadas no comportamento social de insetos é a capacidade de resolução de problemas complexos. Isso é conseguido através da cooperação entre os agentes, que fazem individualmente tarefas locais simples e cumprem tarefas complexas globais. O comportamento coordenado e a auto-organização da sociedade é consequência das simples atividades individuais dos agentes. Pode-se citar como exemplo desse tipo de comportamento o das formigas, que levam pedaços de folhas para os ninhos, para que surjam fungos que servirão de alimentos para as larvas (Dorigo et al, 2006). Outro exemplo das formigas seria o modo como as mesmas se orientam entre o ninho e a fonte de comida.

Os métodos de *Swarm Intelligence* têm grande sucesso na resolução de problemas para a área de otimização. A idéia é aplicar esses métodos para simplificar a solução desse tipo de problema. Um dos mais conhecidos é o problema do caixeiro viajante (*Traveling Salesman Problem* - TSP). Nesse problema, o caixeiro deve visitar um determinado número de cidades e passar por elas apenas uma vez. A solução é descobrir o caminho que satisfaz as regras anteriores e que possua o menor tamanho possível.

Existem problemas de otimização que não possuem um algoritmo que encontre uma solução em tempo polinomial. Esses são conhecidos como problemas NP-Completo. Apesar de existirem algoritmos que resolvam esse tipo de problema, tais algoritmos não possuem um tempo de processamento computacionalmente viável, tornando-se necessário procurar alternativas para esses casos.

Uma opção seria aplicar um método que consiga uma solução muito próxima da solução ótima e que forneça um algoritmo que possua um desempenho computacional viável, isto é, abrindo mão da solução ótima e buscando soluções que sejam satisfatórias.

Os pesquisadores têm aplicado as técnicas de *Swarm Intelligence* para resolver esse tipo de problema.

Uma das mais utilizadas técnicas de *Swarm Intelligence* para solucionar problemas de otimização é a Otimização de Colônia de Formigas (ACO – *Ant Colony Optimization*). Essa será uma das técnicas utilizadas nessa dissertação para ajudar a resolver o problema de roteamento em redes ópticas.

3.2 Introdução ao ACO

Como já foi dito, na literatura atual muitas técnicas inspiradas no comportamento coletivo de animais estão sendo usadas para resolver problemas de otimização. Formigas, como outros insetos, possuem uma forma particular de comunicação indireta. Isso é chamado de estigmergia (Dorigo et al, 1998). Essa comunicação é feita através da deposição de uma substância química chamada feromônio. Essa substância é deixada durante o caminho percorrido pelas formigas. O que servirá para induzir outras formigas que poderão perceber os níveis de feromônio e ser guiadas pelos mesmos. Como consequência, o resultado gerado por essas atividades individuais é a coordenação da colônia.

Muitas pesquisas foram desenvolvidas, onde foram criados algoritmos para solucionar problemas de otimização. Por exemplo, roteamento em redes de comunicação (Dorigo et al, 1998), problema do caixeiro viajante (Dorigo et al, 1997), balanceamento de carga em redes (Schoonderwoerd et al, 1997), entre outros.

Otimização por Colônia de Formigas (Dorigo et al, 2004; Dorigo et al, 2006; Bonabeau et al, 1999) foi umas das primeiras técnicas para otimização baseada em *Swarm Intelligence*. ACO é uma técnica baseada no comportamento de formigas e é aplicada para resolver problemas complexos. Além de ACO, outros algoritmos são frequentemente utilizados, por exemplo, computação evolucionária, tempera simulada e busca tabu.

Marco Dorigo e Cols introduziram o primeiro algoritmo ACO no início dos anos 1990 (Bonabeau et al, 1999). As principais características do ACO são:

- Simulações repetitivas e concorrentes executadas por uma população de formigas artificiais para criar novas possíveis soluções para um problema;
- Buscas estocásticas locais para construir soluções de forma incremental;
- Informações coletadas em simulações passadas para levar buscas posteriores para soluções melhores, ou seja, mais adequadas ao ambiente.

Iterativamente, cada formiga cria uma solução utilizando informações específicas do ambiente. Além disso, utiliza informações deixadas por outras formigas durante iterações passadas do algoritmo. Cada formiga coleta informações sobre as características do ambiente e as utiliza para modificar a visão atual da solução do problema. Essa visão é alterada para que informações contidas em soluções satisfatórias construídas anteriormente sejam exploradas para se construir melhores soluções no futuro. Sendo assim, as atividades das formigas são induzidas através de uma comunicação indireta de acordo com o estado do ambiente.

Existem algumas diferenças entre o comportamento das formigas artificiais e o comportamento das formigas reais (Blum et al, 2008). Primeiro, as formigas reais

depositam feromônio por onde passam, independentemente se estão saindo do ninho ou voltando para o mesmo. As formigas artificiais somente depositam no caminho de volta ao ninho. Segundo, as formigas reais se movem de forma assíncrona e as formigas artificiais se locomovem de forma síncrona em cada iteração do sistema. Terceiro, o depósito de feromônio da formiga real é sempre com a mesma intensidade, enquanto que a intensidade do depósito de feromônio da formiga artificial pode depender de algumas características do ambiente, por exemplo, o nível de sobrecarga do caminho a ser escolhido ou algum custo associado ao mesmo.

3.3 O Experimento das Duas Pontes

Como citado anteriormente, no percurso entre o formigueiro e a fonte de alimento, as formigas são induzidas pelo rastro de feromônio depositado por outras. E tendem a escolher caminhos, probabilisticamente, levando em consideração as concentrações de feromônio. Ou seja, os caminhos que possuírem maiores quantidades de feromônio terão maior probabilidade de serem escolhidos.

Deneubourg et al (1990) e Goss et al (1989) fizeram um experimento com duas pontes. Essas pontes ligavam o ninho a uma fonte de alimento. Isso é ilustrado na Figura 1. Nesse experimento, l_1 é o caminho mais longo e l_s o caminho mais curto. Durante o experimento, variou-se a taxa $r = l/l_s$, ou seja, a proporção entre os tamanhos dos caminhos foi alterada.

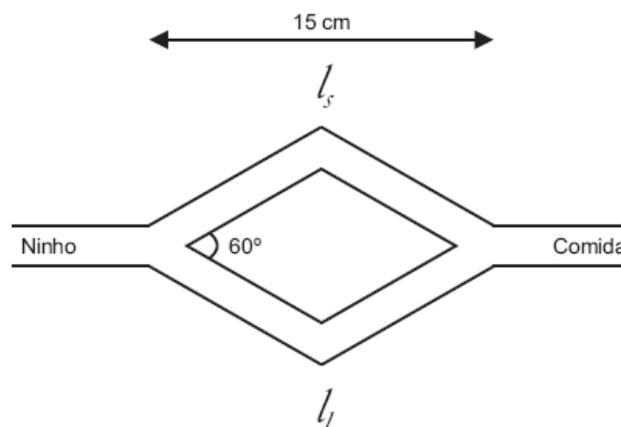


FIGURA 1: Experimento com dois caminhos iguais

No teste acima - Figura 1- o experimento foi realizado com caminhos do mesmo tamanho, ou seja, nesse caso, a taxa r possui o valor igual a 1. Inicialmente, as formigas escolherão um caminho para chegar à fonte de alimento de forma aleatória. Esse fato acontece devido à falta de feromônio para induzi-las a algum caminho. Neste caso, foi constatado que as formigas não têm preferência na escolha, portanto elas selecionam um dos caminhos com a mesma probabilidade. Com um grande número de testes nesse

ambiente, percebeu-se que as formigas utilizavam cada caminho praticamente o mesmo número de vezes.

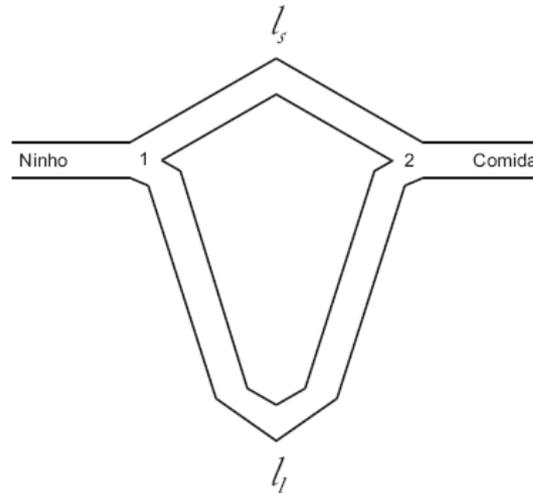


FIGURA 2: Experimento com caminhos diferentes

Já no segundo teste, ilustrado na Figura 2, a taxa r possui valor igual a 2. Nesse caso, um dos caminhos possui o dobro do tamanho do outro (Goss et al, 1989). Novamente, no início dos testes, as formigas escolhem de forma aleatória o caminho a ser seguido. Como ambos os caminhos não possuem concentração alguma de feromônio, para as formigas, os caminhos possuem o mesmo atrativo. Pode-se esperar que, inicialmente, metade das formigas, em média, escolha l_1 e o restante escolha l_s .

A diferença entre os dois testes é que as formigas que escolheram o menor caminho chegam à fonte de alimento antes das formigas que escolheram o maior caminho. Sendo assim, as formigas que escolheram o menor caminho, retornarão ao ninho mais rapidamente do que o restante. Logo, o menor caminho terá a sua concentração de feromônio aumentada. Isso induzirá a decisão das formigas que estão saindo em busca de comida a favor desse caminho.

Esse processo de indução poderia levar todas as formigas a utilizar o caminho mais curto. No entanto, os testes demonstraram que uma pequena quantidade de formigas ainda escolhe o caminho mais longo. Isso se deve a capacidade de exploração de novos caminhos. É essa capacidade que torna a colônia de formigas dinâmica e adaptável a mudanças do ambiente. Essa é uma das principais características do comportamento das formigas.

A Figura 3 abaixo ilustra um terceiro teste. Inicialmente, o menor caminho L_s foi retirado, restando apenas o caminho L_l mais longo. O teste foi executado, nesse cenário, durante 30 minutos. Em seguida foi adicionado o trecho mais curto ao ambiente. Percebeu-se que, mesmo sendo menor, poucas formigas utilizaram o caminho L_s . O maior caminho continuou sendo utilizado pela grande maioria das formigas.

Existem dois motivos para tal constatação: a alta concentração de feromônio no maior caminho e a baixa taxa de evaporação do feromônio nos dois caminhos. Para que seja possível evitar a convergência para caminhos sub-ótimos e possibilitar a exploração

é necessário que a taxa de evaporação do feromônio seja mais elevada, possibilitando o descobrimento de novas rotas menores.

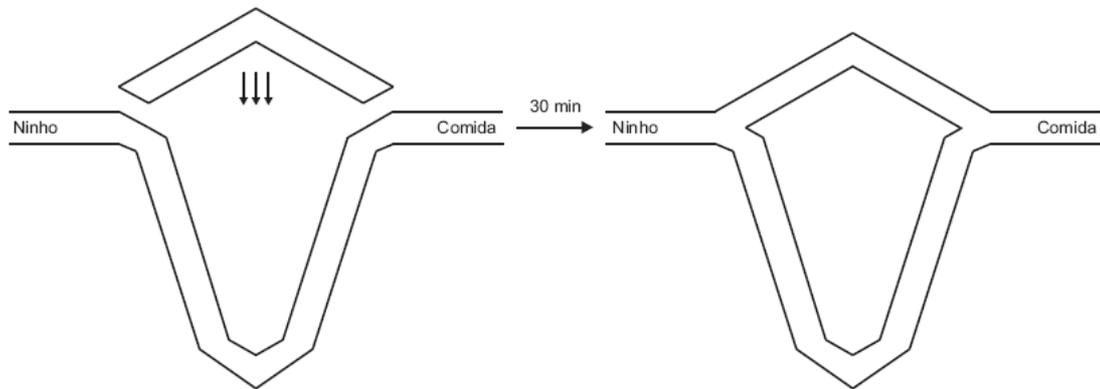


FIGURA 3: Experimento de Adicionar um novo caminho

3.4 Modelo Artificial

Uma das formas para modelar o ambiente do ACO é utilizando um grafo para representar os espaços possíveis para o problema das formigas. De acordo com Blum et al, (2008), podemos considerar o grafo $G = (V, E)$, onde V é o conjunto de nós da rede e E é o conjunto de vértices. Assumiu-se a existência de dois nós v_1 e v_2 , representando respectivamente o ninho e a fonte de alimentos. O conjunto de arestas é formado por duas arestas e_1 e e_2 , onde e_2 possui comprimento maior que e_1 . O valor artificial τ_i representa o nível de intensidade de feromônio, sendo este o resultado dos depósitos feitos pelas passagens das formigas na aresta i , onde $i = 1, 2$.

De acordo com os dados acima, a probabilidade de cada formiga do ninho escolher um dos caminhos e_1 e e_2 é:

$$P_i = \frac{\tau_i}{\tau_1 + \tau_2}, \quad i = 1, 2 \quad (3.1)$$

Como pode ser constatado na equação 3.1, a aresta que possuir o maior nível de feromônio, ou seja, maior τ_i terá a maior probabilidade de ser escolhida. Após chegar ao destino, a formiga retornará pelo mesmo caminho. Onde é necessário atualizar o nível do feromônio das arestas do caminho escolhido. O valor do feromônio é incrementado da seguinte maneira:

$$\tau_i \leftarrow \tau_i + \frac{Q}{l_i} \quad (3.2)$$

Onde l_i é o comprimento da aresta i e Q é um valor constante e positivo, e pode ser representado como um parâmetro do modelo. Nesse caso, a equação 3.2 ilustra que a quantidade de depósito feito pela formiga é inversamente proporcional ao comprimento da aresta, ou seja, quanto menor o comprimento do enlace, maior o depósito feito pela formiga.

Seguindo o modelo natural, onde o nível de feromônio das arestas diminui com o tempo, a evaporação do feromônio artificial é dada pela equação a seguir:

$$\tau_i \leftarrow (1 - p) \cdot \tau_i, \quad i = 1, 2 \quad (3.3)$$

Onde p é o parâmetro que representa a constante de evaporação do ambiente e este pode variar entre 0 e 1.

Ainda em Blum et al (2008), os resultados das simulações feitas no citado trabalho mostraram que, ao longo do tempo, a colônia artificial converge para o menor caminho. Os parâmetros utilizados na simulação foram $L1 = 1$, $L2 = 2$, $Q = 1$ e o nível de feromônio em cada aresta foi de 0,5. E ainda houve uma variação no número de formigas, primeiramente com 10 e depois com 100. No cenário com 100 formigas, houve menos rotas randômicas das formigas.

3.4.1 Ant System

Considerando a explanação anterior, um aspecto deve ser ressaltado: as possíveis soluções para ir do ninho para a fonte de alimento já eram conhecidas. No caso, só existem duas opções, isto é, dois caminhos a escolher. Isso não acontece nos problemas de otimização onde as soluções não são conhecidas e devem ser construídas. Portanto, o modelo ilustrado na seção anterior deve ser adaptado para resolver problemas de otimização.

Ant System (AS) (Dorigo et al, 2006) é o primeiro algoritmo ACO e será explorado para ilustrar a adaptação do modelo ACO a um problema a ser solucionado. AS teve o objetivo de resolver o problema do caixeiro viajante (TSP).

No cenário do problema TSP, as arestas são parte da solução e a cada uma delas é atribuído um valor inicial de feromônio. Ou seja, cada aresta e_{ij} receberá inicialmente o valor $\tau_{i,j}$, e o objetivo da colônia de formigas passa a ser construir uma solução satisfatória para o problema de otimização em questão. Sendo assim, deixam de existir as entidades ninho e fonte de alimentos.

Inicialmente, um dos nós do grafo é escolhido como origem. A partir dele a formiga terá que percorrer todos os nós não visitados do grafo e cada aresta visitada é

adicionada à solução atual. A solução é formada quando não houver mais nós não visitados.

As equações 3.3, 3.4 e 3.5 representam respectivamente a probabilidade de escolher um nó, a evaporação do nível de feromônio dos enlaces e a atualização do nível de feromônio de um enlace.

$$p(e_{i,j}) = \frac{\tau_{i,j}}{\sum_{\{k \in \{1, \dots, |V|\} \mid v_k \notin T\}} \tau_{i,k}}, \forall j \in \{1, \dots, |V|\}, v_j \notin T \quad (3.4)$$

$$\tau_{i,j} \leftarrow (1 - p) \cdot \tau_{i,j}, \forall \tau_{i,j} \in \tau \quad (3.5)$$

$$\tau_{i,j} \leftarrow \tau_{i,j} + \frac{Q}{f(s)} \quad (3.6)$$

Onde Q é uma constante positiva e $f(s)$ é o valor da função objetivo. Os passos do algoritmo são repetidos iterativamente até chegar numa condição de parada. O número de formigas da colônia também deve ser definido.

Na equação 3.3, a probabilidade do enlace $e_{i,j}$ ser escolhido é determinado pela divisão entre $\tau_{i,j}$ e o somatório de $\tau_{i,k}$ onde k são os vizinhos de i diferentes de j . Na equação 3.4, o fator p é o valor que diminuirá o nível de feromônio de $\tau_{i,j}$.

3.5 Mecanismos de Funcionamento

O principal mecanismo dos algoritmos inspirados em formigas é fazer uso das características das formigas reais, permitindo que as formigas possam memorizar caminhos que foram percorridos e características associadas a eles, como o custo, por exemplo. Alguns outros comportamentos foram adicionados, como a capacidade de construir soluções, a análise de qualidade da solução e a deposição proporcional de feromônio. Outro comportamento importante é a evaporação de feromônio, que também foi adicionada com o intuito de evitar a convergência para caminhos sub-ótimos (Dorigo et al, 2004).

Como vimos nas seções anteriores e ainda segundo Dorigo et al (2004), o algoritmo padrão de ACO possui vários mecanismos dentro do ambiente gerado pela colônia de formigas. A partir das definições desses mecanismos, torna-se fácil a compreensão do ambiente e do comportamento das formigas.

Alguns mecanismos importantes que determinam o comportamento das formigas na colônia são o movimento das formigas, a deposição de feromônio e a evaporação de feromônio.

3.6 Algoritmo ACO para Escolha do Menor Caminho

Através da definição dos mecanismos citados na seção anterior, o algoritmo de ACO pode resolver o problema do menor caminho em redes. Esses mecanismos definem os processos do algoritmo. Para facilitar a compreensão, a seguir serão apresentados algoritmos em pseudo-linguagem, cujas variáveis serão consideradas com valores iniciais já atribuídos. Nesses algoritmos, o custo de cada aresta representa a distância entre os nós.

O Algoritmo 1 define como a colônia de formigas chega à solução. A colônia é composta por diversos grupos contendo a mesma quantidade de formigas. Os grupos de formigas percorrerão o grafo para achar o caminho utilizando uma quantidade de passos pré-definida. Cada formiga sempre é lançada no nó de origem com o intuito de chegar ao nó de destino. A formiga se move através do método `anda()`, e este incorpora a memorização da formiga e a regra de decisão. No momento em que uma formiga k chega a seu destino, a mesma é lançada de volta ao seu ponto de partida. No retorno, a formiga deposita a quantidade de feromônio calculada em relação à qualidade do caminho encontrado. No fim de cada passo, o método de evaporação é chamado.

Algoritmo 1: Colônia se movimentando dentro do grafo em busca do caminho de menor custo.

Entrada: Um par (origem, destino)

Saída: Caminho de menor custo

```

1 enquanto passo <= numeroDePassos 1 faça
    // varre cada grupo da colônia
2   para n = 0 até colonia.tamanho() faça
    // varre cada formiga do grupo
3     para k = 0 até grupo.tamanho() faça
4       grupoAtual ← colonia [n]
    // preenche grupo de formigas
5     se grupoAtual.tamanho() < numeroDeFormigasPorGrupo então
6       lancaFormigaPraFrente(n, noOrigem, noDestino)
7     fim
8     formiga ← grupoAtual [k]
    // movimento da formiga
9     formiga.anda()
    // formiga chega ao destino
10    se formiga.noAtual = noDestino então
11      lancaFormigaPraTras(n, k)
12      caminhoDaFormiga ← formiga.caminho
    // melhor caminho achado
13      se caminhoDaFormiga.custo < caminhoSolucao.custo então
14        caminhoSolucao ← caminhoDaFormiga
15    fim

```

```

16     mataFormiga(formiga)
17     fim
18     fim
19     fim
20     evaporaFeromonio()
21     passo ← passo + 1
22 fim
23 retorna caminhoSolucao

```

3.6.1 Movimento das Formigas

Normalmente, a regra para a formiga se movimentar de um nó para o outro é baseada no nível de feromônio. Porém, para o problema de menor caminho entre dois nós, a distância entre esses nós deve influenciar na escolha do caminho, juntamente com a taxa de feromônio. Abaixo, a equação ilustra tal comportamento.

$$p_{ij}^k = \begin{cases} r \cdot \frac{[\tau_{ij}]^\alpha}{[d_{ij}]^\beta}, & \text{se } j \in N_i^k ; \\ 0, & \text{se } j \notin N_i^k ; \end{cases} \quad (3.7)$$

Os índices k , i e j representam, respectivamente, a formiga, o nó de origem e o nó de destino. O termo d_{ij} representa a distância entre os nós i e j , e r é um valor aleatório entre 0 e 1. Esse fator tem o papel de inserir a diversidade na decisão. Isso mantém a capacidade de exploração.

Os valores de α e β são constantes que representam a importância de cada termo na equação. O Algoritmo 2 apresenta os passos para o processo de regra de decisão. Nele a formiga analisa quantos nós podem ser escolhidos e faz a análise de qual deles é o melhor, levando em consideração a capacidade de indução dos mesmos.

Algoritmo 2: Método em que a formiga decide o próximo nó a ser visitado.

Saída: Próximo nó a ser visitado ou um erro de decisão

```

// formiga sem possíveis próximos nós
1 se nosPossiveis.vazio() = Verdadeiro 1 então
2     retorna ERRO_DE_DECISAO
3 fim
4 proximoNo ← nosPossiveis [0]
5 arcoEscolhido ← grafo [noAtual][nosPossiveis [0]]
6 para  $i = 1$  até nosPossiveis.tamanho() faça
7     arcoAtual ← grafo [noAtual ][nosPossiveis [i]]

```

```

8  se calculaInducao(arcoAtual) > calculaInducao(arcoEscolhido) então
9  arcoEscolhido ← arcoAtual
10 proximoNo ← nosPossiveis [i]
11 fim
12 fim
13 retorna proximoNo

```

3.6.2 Deposição e evaporação

O Algoritmo 3 representa o processo de deposição de feromônio que é feito no caminho de volta ao ninho. A atualização leva em consideração a taxa de feromônio e a distância entre os nós representada pelo custo do enlace. Cada aresta tem o valor acrescentado da mesma quantidade de feromônio.

Como visto anteriormente, o processo de evaporação, representado pelo algoritmo 4, é feito no fim de cada passo do algoritmo, ou seja, após o movimento de todas as formigas da colônia. Atua retirando uma parte do valor do feromônio nas arestas do grafo.

Algoritmo 3: Método usado por cada formiga para depositar feromônio no caminho durante a viagem de volta.

Entrada: Caminho onde será depositado o feromônio

```

1 para i = 0 até caminho.1 tamanho() faça
2  arco ← caminho [i]
3  arco.feromonio ← arco.feromonio + (unidadesDeFeromonio / caminho.custo )
4 fim

```

Algoritmo 4: Método como o feromônio é evaporado de cada arco do grafo.

```

1 para i = 0 até grafo.numeroDeNos faça
2  para j = 0 até grafo.numeroDeNos faça
3   se grafo [i][j].existe() então
4    arco ← grafo [i][j]
5    arco.feromonio ← arco.feromonio * (1 - taxaDeEvaporacao )
6  fim
7 fim
8 fim

```

Os algoritmos mostrados nessa seção ilustram a simplicidade dos mecanismos da ACO, o que torna possível a adaptação dos mesmos a vários problemas de otimização. É

mostrado também que as tarefas individuais realizadas pelas formigas são muito simples, porém, com o trabalho coletivo podem chegar à solução de tarefas complexas.

Capítulo 4

Redes Ópticas

Redes ópticas são redes de computadores que utilizam fibras ópticas como meio de transmissão. Isto permite que as redes desfrutem de todas as vantagens que esse meio de transmissão proporciona, como por exemplo, alta largura de banda, confiabilidade, baixa taxa de erro, baixa latência, entre outras (Sivarajan et al, 2002).

Para haver comunicação entre dois nós, primeiramente, é necessário estabelecer um caminho de luz entre os mesmos. Um caminho de luz ou *lightpath* é formado pelo conjunto de nós e enlaces entre o nó de origem e o nó de destino através do uso de um comprimento de onda. Outra questão importante é definir se a comunicação utilizará o mesmo comprimento de onda durante todo o caminho ou se poderão ser utilizados vários comprimentos de onda. No primeiro caso, deve-se escolher um comprimento de onda que esteja disponível em todos os enlaces. Caso a escolha seja equivocada, ou seja, o comprimento de onda não esteja disponível em algum enlace do caminho, haverá bloqueio na comunicação. No segundo caso, a rede deverá possuir conversores de comprimento de onda capazes de receber sinais em um comprimento de onda e transmiti-los em outro. Isso diminuiria a probabilidade de bloqueio da comunicação. No entanto, o custo de utilização dos conversores torna os mesmos pouco viáveis atualmente. A Figura 4 ilustra um exemplo de uma rede óptica.

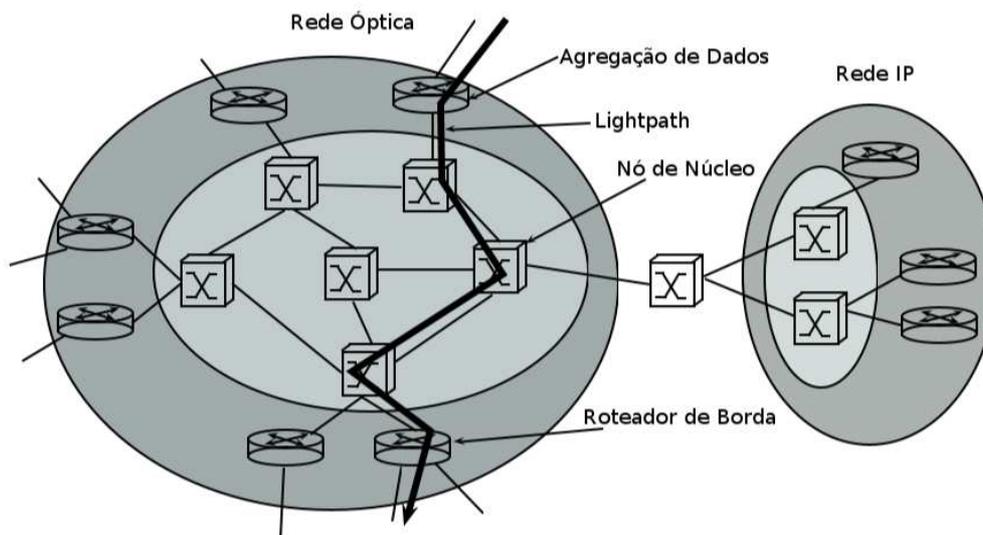


FIGURA 4: Exemplo de Rede Óptica

Na Figura 4 acima, o *lightpath* e os componentes da rede óptica são destacados. Os dados dos clientes são armazenados no roteador de borda onde a rota e o comprimento de onda do *lightpath* são definidos. Do roteador de borda, o *lightpath* segue para os roteadores de núcleo que são chamados de *Optical CrossConnect (OXC)* ou comutadores ópticos. Esses são responsáveis por comutar os dados até o roteador de borda de destino. Deste, os dados são repassados aos clientes.

O processo de definição do caminho de luz, que consiste na definição da rota e do comprimento de onda, é conhecido como problema de Roteamento e Atribuição do Comprimento de Onda (RWA – Routing and Wavelength Assignment). Há muita pesquisa em cima desse problema (Silveira et al, 2003) e ele foi escolhido como o foco dessa dissertação.

Algoritmos de roteamento são utilizados para configurar e manter as tabelas de roteamento. O desempenho da rede depende diretamente da qualidade desses algoritmos. Além disso, é necessário aumentar o grau de utilização dos recursos da rede.

A tecnologia WDM (*Wavelength Division Multiplexing*) é muito utilizada em conjunto com as redes ópticas com o objetivo de aumentar a utilização dos recursos da rede. Nas redes WDM a informação é levada simultaneamente através de vários canais, ou seja, através dos comprimentos de onda, o que permite um aumento substancial na largura de banda. Os sinais são multiplexados no início da fibra e demultiplexados no final da mesma.

Existem algumas variações da WDM: a DWDM (*Dense Wavelength Division Multiplexing*) e a CWDM (*Coarse Wavelength Division Multiplexing*). A diferença entre elas é basicamente a densidade de canais, isto é, a quantidade de comprimentos de onda numa faixa de frequência. Isso permite uma maior capacidade de transmissão (Silveira et al, 2003).

A tecnologia WDM é usada para fazer a integração entre a rede óptica e a rede cliente, por exemplo, rede IP. A Figura 5 ilustra essa idéia e a evolução das arquiteturas utilizadas (Oliveira et al, 2005).

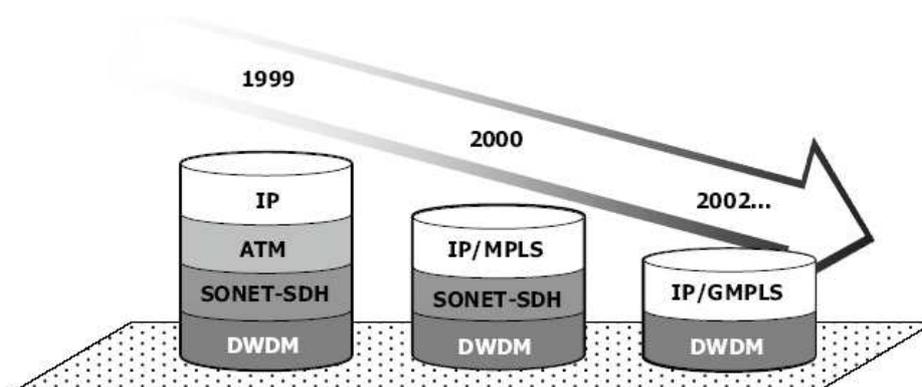


FIGURA 5: Evolução das arquiteturas da integração IP/Óptico

Além de usar a tecnologia WDM, para que os benefícios das redes ópticas sejam ampliados, é necessário evitar as conversões para o meio eletrônico, ou seja, fazer com que os dados sejam transmitidos totalmente no domínio óptico. Isso é conseguido numa comutação puramente óptica. As seções a seguir irão explicar os detalhes dos tipos de comutação em redes ópticas.

4.1. Comutação em Redes Ópticas

Como citado anteriormente, na literatura, existem três tipos de comutação em redes ópticas: comutação por circuito (*Optical Circuit Switching* - OCS), comutação por pacote (*Optical Packet Switching* - OPS) e comutação por rajada (*Optical Burst Switching* - OBS) (Wei et al, 2000; Mukherjee, 2006; Yoo et al, 1997; Wei et al, 2000). Nas seções a seguir, serão apresentados cada um dos tipos de comutação em redes ópticas.

4.1.1. Comutação por Circuitos

A comutação por circuitos (*Optical Circuit Switching* – OCS) necessita que uma conexão seja estabelecida para possibilitar a transmissão dos dados. Nessa comutação, o nó inicial envia um pacote de controle para o nó de destino que, ao receber o pacote de controle, responde confirmando ou não o estabelecimento da conexão. Se a resposta for positiva, o nó inicial pode enviar os dados até a conexão ser finalizada. O pacote de controle tem a função de reservar os recursos durante o caminho da conexão, além de levar informações úteis para tomadas de decisão na rede e evitar colisão.

A desvantagem desse tipo de comutação é o desperdício de recursos, pois, durante o tempo da conexão (desde o estabelecimento até o encerramento), nem sempre os recursos da rede estarão sendo usados. Além da latência gerada pelo tempo de estabelecimento da conexão. Abaixo, a Figura 6 ilustra o funcionamento da OCS.

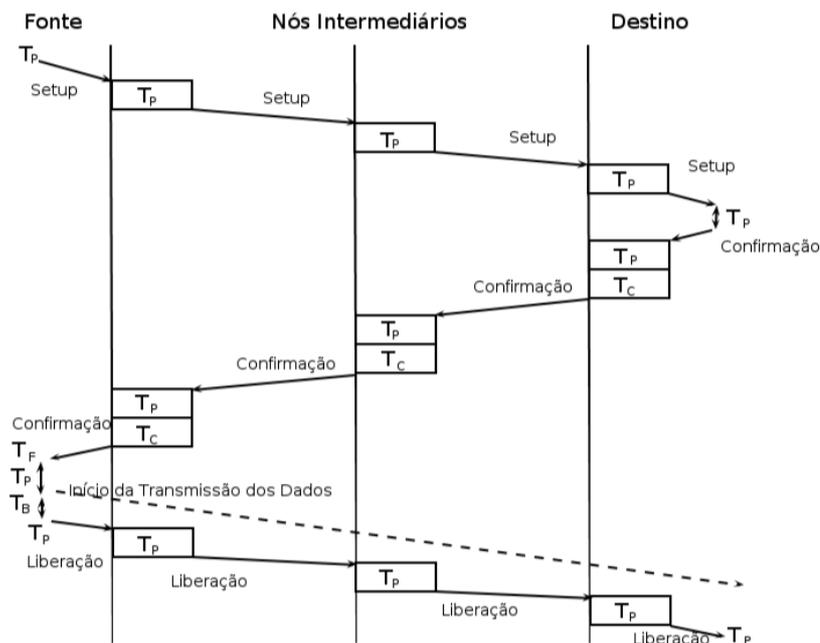


FIGURA 6: Comutação por Circuitos

Considerando t_p o tempo de processamento, t_c o tempo de configuração da matriz óptica e t_b é o tempo para transmitir a rajada, após o envio da rajada, um pacote de liberação do recursos (*release*) dos recursos deve ser enviado.

Assim, a latência para o envio de uma mensagem em redes desse tipo pode ser calculada da seguinte maneira:

$$\text{Latência}_{\text{Total}} = 6t_f + 3(n-1)t_l + (2n+3)t_p + nt_c + t_b$$

Onde t_f é o atraso de propagação da estação de acesso até o nó de borda, t_l é o atraso de propagação dos enlaces entre os nós.

4.1.2. Comutação por Pacotes

Outra opção é a comutação por pacote (*Optical Packet Switching - OPS*) que possui a vantagem de evitar o desperdício de recursos. Nesse tipo de comutação, não existe a necessidade de estabelecimento de conexão. Os pacotes são roteados nó a nó até o seu destino. Porém, para haver roteamento, é necessário que haja o processamento de dados. Com a tecnologia atual ainda não é possível processar dados no meio óptico, ou seja, sem antes transformá-los em sinal eletrônico. Essa transformação é chamada de O-E-O e deve ser evitada, pois o tempo necessário para transformar os dados para o meio eletrônico, fazer o roteamento através do cabeçalho do pacote e depois transformar de volta para o meio óptico aumenta consideravelmente a latência da rede. Isso causaria um grande atraso na comunicação, por ser a velocidade de processamento muito inferior à velocidade de propagação da fibra óptica.

Como a transformação O-E-O não é eficiente, esse tipo de comutação é inviável para o momento. Além disso, ainda seria necessário utilizar *buffers* para armazenar as informações quando houvesse formação de filas, ou seja, quando houvesse mais dados que a capacidade máxima de processamento. *Buffers* devem ser evitados, se possível, devido ao alto custo dos mesmos. Aqui o pacote de liberação não é utilizado, pois não há reserva de recursos. Esses são utilizados imediatamente após a leitura do cabeçalho e depois liberados automaticamente. A Figura 7 ilustra o funcionamento da comutação por pacotes.

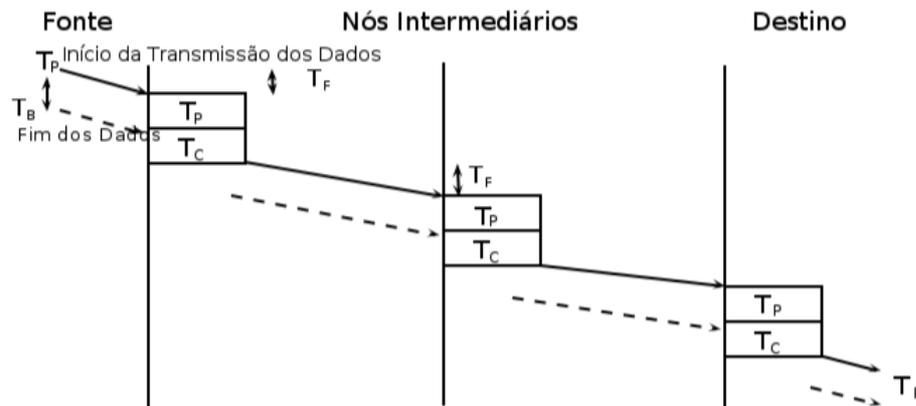


FIGURA 7: Comutação por Pacotes

Considere que t_p é o tempo de processamento, t_c é o tempo de configuração da matriz óptica e t_b é o tempo de transmissão da rajada. Pode-se mostrar que a latência da rede é dada por:

$$\text{Latência}_{\text{Total}} = 2t_f + (n-1)t_l + (n+2)t_p + nt_c + t_b$$

Onde t_f é o atraso de propagação da estação de acesso até o nó de borda, t_l é o atraso de propagação dos enlaces entre os nós.

4.1.3 Comutação por Rajadas

A terceira opção é a comutação por rajadas ópticas (*Optical Burst Switching – OBS*) que tem o objetivo de evitar os problemas dos tipos de comutação anteriores, ou seja, o desperdício da comutação por circuito e a utilização de tecnologias inviáveis da comutação por pacotes.

Nesse tipo de comutação, primeiramente, os dados são armazenados no nó de borda da rede, ou seja, o nó que faz fronteira entre a rede óptica e outras redes - por exemplo, uma rede IP. Neles, os dados são agregados até um determinado ponto, dependendo da estratégia utilizada. Esse acúmulo pode acontecer até um tempo especificado, ou até uma determinada quantidade de dados, ou até uma abordagem híbrida das duas anteriores (Mukherjee, 2006). Esse problema é conhecido como Problema de Montagem da Rajada (Mukherjee, 2006).

Depois que a rajada é montada, é determinado um comprimento de onda e o caminho a ser percorrido. Esse é o problema chamado de Roteamento e Atribuição do Comprimento de Onda (RWA) e será explicado em detalhes nas próximas seções. Logo após, um pacote de controle (*Burst Control Packet* - BCP) é enviado para o nó de destino. Esse pacote de controle usa um canal independente dos dados (*out-of-band*), que percorre o caminho fora do plano de dados num comprimento de onda específico.

O objetivo do BCP é reservar os recursos da rede para a rajada que será enviada poucos instantes após o envio do mesmo. Para reservar os recursos, o pacote de controle deve ser transformado para meio eletrônico, pois exigirá o processamento do mesmo. O nó que montou a rajada não espera a confirmação do nó de destino. Após o envio do pacote de controle, o nó de borda aguarda um tempo de ajuste, calculado previamente, e depois envia a rajada. A forma de calcular esse tempo de ajuste também é um problema explorado na literatura (Sousa et al, 2008; Yoo et al, 1997; Wei et al, 2000), mas não é o foco dessa dissertação, apesar desse tempo também influenciar na probabilidade de bloqueio da rede.

Diferentemente do pacote de controle, a rajada é apenas comutada entre os nós intermediários, não havendo a necessidade de processá-la. Isso implica que não há necessidade da transformação O-E-O. Portanto, o cálculo do tempo de ajuste é um importante ponto para concluir a comunicação com sucesso, pois a latência da rajada é bem menor que a do pacote de controle. Caso o tempo de ajuste seja mal calculado, a rajada poderá alcançar o pacote de controle antes que este consiga reservar os recursos. Nesse caso, a rajada é perdida, pois o pacote de controle não teve tempo suficiente para reservar os recursos necessários.

A principal vantagem dessa comutação é a separação dos planos de dados e de controle (Sousa et al, 2008), o que possibilita um bom gerenciamento da rede e evita o desperdício de recursos, pois não estabelece uma conexão. Isso diminui a latência e aumenta a eficiência da rede, ou seja, aumenta a utilização dos recursos da rede, além de utilizar uma tecnologia viável. Como essa comutação supera os problemas encontrados nas anteriores, ela é a mais adequada para o momento.

Protocolos de sinalização são utilizados para definir como os pacotes de controle vão atuar. Os principais são o JIT (*Just-In-Time*) e o JET (*Just-Enough-Time*) (Yoo et al, 1997; Wei et al, 2000). Essa dissertação usará o protocolo JET e este será definido melhor na próxima seção.

Pode-se mostrar que a latência da rede é dada por:

$$\text{Latência}_{\text{Total}} = 2T_f + (n-1)T_1 + (n+2)T_p + T_b$$

A Figura 8 abaixo ilustra o funcionamento de uma rede OBS.

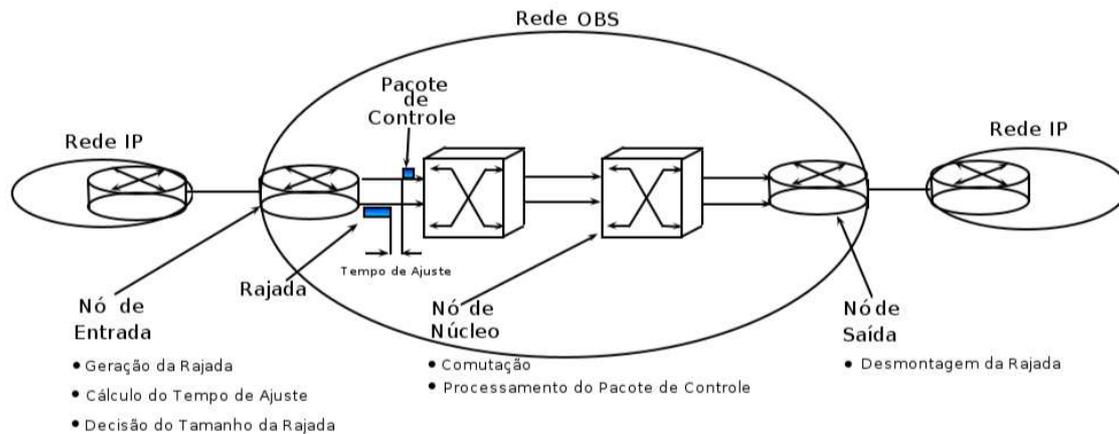


FIGURA 8: Comutação por rajadas

4.2. Protocolos de Sinalização

Os protocolos de sinalização são responsáveis por determinar como os recursos da rede serão reservados e desalocados. Para isso, um pacote de controle é utilizado e é enviado do nó de origem ao nó de destino.

Existem dois modos de reservar recursos, *two-way reservation* e *one-way reservation*. No primeiro caso, o pacote de controle é enviado para o destino que responde confirmando a alocação dos recursos. Um exemplo desse tipo de protocolo é o *Tell-And-Wait* (TAW). No segundo modo de reserva, o pacote de controle reserva os recursos sem a necessidade de resposta do nó destino. Os protocolos JET (*Just-Enough-Time*) e JIT (*Just-In-Time*) são exemplos desse tipo de reserva (Yoo et al, 1997). A seguir serão apresentados estes dois protocolos de sinalização.

4.2.1. Just-In-Time (JIT)

O protocolo JIT usa um canal fora da banda para enviar as mensagens de sinalização. A rajada é mantida no nó de borda, e no momento em que esta é montada um pacote de controle é enviado com uma mensagem de *setup*, com o objetivo de reservar os recursos no caminho onde a rajada irá passar. Após isso, o nó de origem aguarda um tempo de ajuste e depois envia a rajada. Quando o pacote de controle chegar ao nó de destino, este responde com uma mensagem de *connect* ao nó de origem da rajada. Se essa mensagem não chegar, o nó de origem assume que os recursos não foram reservados e inicia o processo novamente.

Uma das características marcante do JIT é a reserva imediata (Qiao et al, 1999). A partir do processamento do pacote de controle, o nó envia o mesmo para o próximo nó do caminho e imediatamente configura a matriz de comutação reservando o canal, caso o comprimento de onda esteja disponível no momento.

Abaixo a Figura 9 ilustra o funcionamento do JIT. Perceba que após o pacote de controle sair de algum nó, o mesmo é configurado (T_{oxc}) e fica reserva a partir desse momento. O JIT utiliza uma mensagem de liberação de recursos após o envio da rajada. Apenas com o recebimento dessa mensagem o nó libera os recursos.

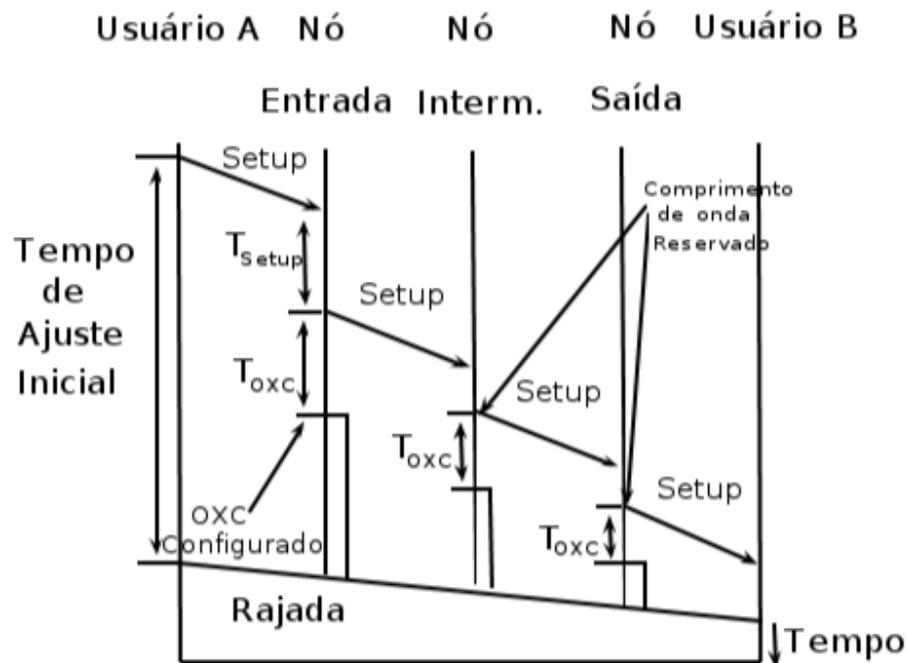


FIGURA 9: Funcionamento do JIT

4.2.2. Just-Enough-Time (JET)

O JET tem como principal característica a reserva atrasada (Qiao et al, 1999). O pacote de controle contém a informação do tamanho da rajada e do tempo de ajuste. Isso possibilitará que a reserva seja feita apenas durante o tempo da rajada. Outros protocolos, como o JIT, descrito na seção anterior, fazem a chamada reserva imediata. Isso pode acarretar um desperdício, pois a rajada não chegará imediatamente e ainda será esperado um outro pacote para a liberação dos recursos.

Com a informação do tempo de ajuste, a reserva no JET é feita apenas para quando a rajada chegar. A reserva só terá a duração do tempo suficiente para passar a rajada, pois o pacote de controle possui o tamanho da rajada. Portanto, não precisa de um pacote de liberação de recursos.

A Figura 10 abaixo mostra o funcionamento do protocolo JET, enfatizando a existência de espaços vazios (*voids*) entre a chegada do pacote de controle e da rajada, o que caracteriza a reserva atrasada. O T_{oxc} é o tempo para configurar a matriz de comutação.

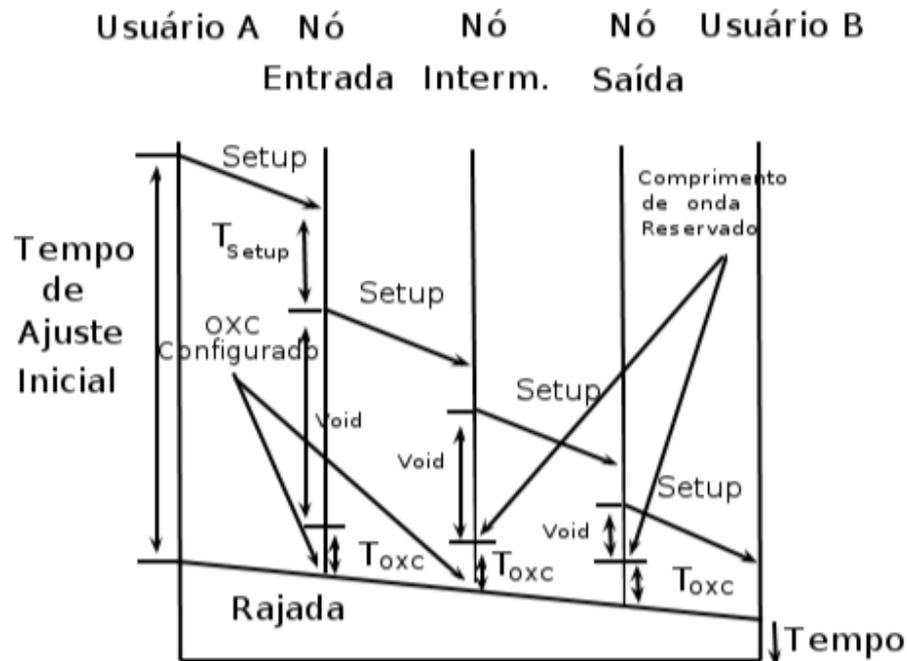


FIGURA 10: Funcionamento JET

Como visto na figura 10 acima, primeiramente, é enviada uma mensagem de *setup*. Esta precisa de certo tempo t_{setup} para ser processada, ou seja, para reservar os recursos. Depois, os espaços vazios, ou seja, *voids* são formados até o momento em que o comutador é configurado para receber a rajada. Este precisa de um tempo t_{oxc} para efetuar a configuração antes da rajada chegar.

Os protocolos de sinalização são responsáveis pela reserva dos recursos, mas não são responsáveis por escolher qual rota deve ser seguida e qual o comprimento de onda a ser utilizado pela rajada (Mukherjee, 2006). Essas definições formam um problema chamado RWA (*Routing and Wavelength Assignment*) que é detalhado na próxima seção.

4.3. Roteamento e Atribuição do Comprimento de Onda (RWA)

Uma das características das redes OBS é a montagem da rajada de dados na borda da rede, ou seja, os dados são armazenados no roteador de borda. Finalizado o período de montagem, uma rota deve ser escolhida entre a origem e o destino. Esse caminho é chamado de caminho de luz ou *lightpath*. Devido à utilização da multiplexação por comprimento de onda, a tarefa seguinte é a escolha de um comprimento de onda por onde os dados devem ser enviados.

O problema de definir a rota e o comprimento de onda é conhecido na literatura como roteamento e atribuição do comprimento de onda ou RWA (*Routing and Wavelength Assignment*) (Mukherjee, 2006; Silveira et al, 2003; Oliveira et al, 2005). As

informações sobre rota e comprimento de onda são calculadas e colocadas no pacote de controle. Este é enviado de acordo com o protocolo de sinalização. A forma como é determinada a rota e o comprimento de onda afeta diretamente o desempenho da rede. Pois, se a rota for mal configurada ou o comprimento de onda for mal escolhido, existirá uma grande probabilidade de bloqueio dos recursos.

Embora redes ópticas possuam grande capacidade de transmissão, é necessário gerenciar a alocação dos recursos de uma forma adequada com o objetivo de diminuir a probabilidade de perda da rajada.

Os algoritmos de RWA devem seguir algumas regras:

- Dois *lightpaths* podem possuir enlaces em comum, desde que não seja usado o mesmo comprimento de onda nos enlaces comuns;
- Caso não seja possível fazer conversão do comprimento de onda, o mesmo comprimento de onda deve ser utilizado em todos os enlaces do caminho.

Para gerenciar eficientemente os recursos da rede, um bom algoritmo de RWA é necessário (Silveira et al, 2003). O desempenho da rede óptica depende de uma boa escolha do comprimento de onda e da escolha da rota. Caso contrário, em um determinado momento, pode acontecer que os recursos estejam ocupados, ou seja, o comprimento de onda esteja alocado à outra rajada durante o instante pedido, naquela rota escolhida. Nesse caso, ocorre o bloqueio da rajada por falta de recursos na rede.

Pode-se definir que os algoritmos de RWA buscam a redução da probabilidade de bloqueio da rajada. Portanto, têm o objetivo de tentar fazer com que todas, ou o maior número possível de requisições da rede sejam satisfeitas sem que haja conflitos. O bloqueio da rajada é caracterizado quando, num dado momento, os recursos da rede não são suficientes para atender a novas requisições devido aos recursos da rede estarem alocados às requisições anteriores.

De acordo com (Silveira et al, 2003), os algoritmos RWA podem atuar de duas formas: centralizada e distribuída. No centralizado, existe um controlador central que possui e mantém todas as informações do estado da rede. Utilizando essas informações, ele é responsável pela atribuição do comprimento de onda e pela escolha da rota. Nesse caso, os nós da rede não têm conhecimento das informações do estado da rede. Na distribuída, os nós utilizam uma abordagem distribuída de busca rápida para selecionar as informações, ou ainda usar rotas pré-definidas e procurar por comprimentos de onda livres na rota selecionada.

Algoritmos de RWA podem ser classificados como estáticos (*Static Lightpath Establishment* - SLE) e dinâmicos (*Dynamic Lightpath Establishment* - DLE) (Shi et al, 2005). O RWA estático é caracterizado por rotas permanentes. Nesse caso, as rotas são configuradas antecipadamente de acordo com as requisições que são conhecidas previamente. O objetivo é criar rotas de forma que se atenda o maior número de requisições possíveis. No caso do RWA dinâmico, as rotas variam de acordo com o tempo, com a demanda de requisições e com o estado da rede. A idéia é evitar a existência de conflitos ou diminuir a probabilidade de bloqueio.

Algoritmos de roteamento são utilizados para definir as tabelas de roteamento da rede óptica. Porém, devido à necessidade de rapidez e otimização, uma abordagem dinâmica deve ser utilizada para manter as tabelas de tal forma que a rede seja robusta o

suficiente para adaptar-se às mudanças durante o tempo. Por exemplo, no caso em que a rede esteja congestionada numa determinada rota, ou no caso em que um determinado nó tenha saído da rede temporariamente.

4.3.1. Roteamento

Apesar dos grandes recursos disponíveis numa rede óptica, o crescimento da demanda de tráfego de dados faz com que a escolha de rotas de menor custo se torne uma tarefa chave para um bom desempenho dessas redes.

Diversas métricas podem ser usadas no cálculo dos custos de uma rota. Por exemplo, o cálculo pode ser baseado no número de saltos ou em pesos atribuídos aos enlaces ou na quantidade de comprimentos de onda disponíveis.

De acordo com a literatura, a seguir serão feitas descrições de tipos de algoritmos de roteamento utilizados nos trabalhos de pesquisas (Zang et al, 2000; Sivarajan et al, 2002; Mukherjee, 2006):

- Roteamento com rota fixa: esse é o tipo de roteamento mais simples. Para cada par de nós origem e destino uma única rota é escolhida. As rotas são definidas antecipadamente. As mesmas são estáticas, ou seja, não mudam com o tempo. Dijkstra (Tanenbaum, 2003) é um exemplo de algoritmo que pode ser aplicado nesse tipo de roteamento. O primeiro problema encontrado é que o estado atual da rede não é levado em consideração. Caso ocorra um congestionamento num setor da rede ou algum problema num enlace qualquer, o roteamento da rede não será capaz de contornar o problema. Existirá também uma má utilização dos recursos da rede. A carga de uma rota pode ser alta enquanto outras rotas podem estar sem utilização. Esses problemas causam uma alta probabilidade de bloqueio de rajada. Para evitá-los, devem ser aplicados algoritmos de roteamento com rotas alternativas ou adaptativas;
- Roteamento com rota fixa alternada: nesse caso, múltiplas rotas são definidas para cada par de nós origem e destino. As rotas também são estáticas e estabelecidas antecipadamente. Além disso, são disjuntas, ou seja, para o mesmo par origem-destino, não possuem enlaces em comum. As opções de rotas são ordenadas, de forma que, quando chega uma requisição, as mesmas são testadas seguindo essa ordem estabelecida. O bloqueio da rajada ocorre somente no caso em que todas as opções de rotas não estejam disponíveis. Esse tipo de roteamento possui uma probabilidade de bloqueio menor que o roteamento com rotas fixas. Apesar disso, a rede ainda não representa o seu estado atual e o problema do balanceamento de carga da rede ainda continua.
- Roteamento adaptativo: as rotas são definidas dinamicamente. A chegada de uma nova requisição de conexão inicia o processo do cálculo da rota. Esse cálculo leva em consideração o estado atual da rede, o que torna o roteamento dinâmico. Ainda pode ser utilizada alguma heurística para fazer parte do roteamento, como por exemplo Algoritmos Genéticos, Redes Neurais, *Ant Colony Optimization* –

ACO, entre outros. Para isso deve ser utilizado um protocolo de gerenciamento e controle para tornar as tabelas de roteamento atualizadas, levando em consideração o estado da rede. A probabilidade de bloqueio é baixa comparada aos tipos anteriores e esse tipo de roteamento é altamente resistente a falhas. Embora uma carga de dados extra seja gerada pelo protocolo de gerenciamento, esse tipo de roteamento possui um bom desempenho, levando em consideração os anteriores.

No problema de roteamento, rotas são calculadas de acordo com alguma heurística aplicada às informações da rede. O objetivo é encontrar rotas que consigam satisfazer as requisições que chegam à rede. Nas redes WDM, cada enlace possui vários comprimentos de onda, e cada um deles pode levar informações distintas. Logo, além de definir a rota, deve-se definir o comprimento de onda a ser utilizado.

4.3.2. Atribuição do Comprimento de Onda

No problema de atribuição de comprimento de onda, um dos comprimentos de onda disponíveis deve ser escolhido para uma determinada rota. O critério de escolha pode variar dependendo do algoritmo ou heurística utilizados.

Embora a melhor rota entre dois nós tenha sido escolhida, uma atribuição equivocada de um comprimento de onda para a rota pode levar ao bloqueio da rajada. Isso pode acontecer se, em algum dos enlaces, o comprimento de onda tiver sido previamente reservado para outra rajada e não houver conversores de comprimento de onda.

Existem vários algoritmos de atribuição de comprimento de onda. Abaixo seguem alguns exemplos encontrados na literatura (Silveira *et al*, 2003; Zang *et al*, 2000; Sivarajan *et al*, 2002; Mukherjee, 2006):

- Escolha Randômica: Primeiramente, os comprimentos de onda disponíveis são agrupados. Quando uma requisição chega, é feita uma escolha aleatória seguindo uma distribuição de probabilidade uniforme, onde todos terão a mesma probabilidade de ser escolhidos. O objetivo dessa estratégia é distribuir igualmente a utilização dos comprimentos de onda;
- Ordem fixa: Nessa estratégia, os comprimentos de onda são colocados numa lista com ordem fixa, previamente estabelecida. Na busca por um comprimento de onda disponível, o primeiro da lista é escolhido. Caso este já esteja alocado, o segundo é testado e assim continua até achar um disponível. Informações globais não são necessárias. O custo computacional é menor que na Escolha Randômica. Na maioria dos casos, o desempenho, em termos de probabilidade de bloqueio, também é menor. Devido a sua pequena complexidade e o baixo custo computacional é colocado em prática;

- Menos utilizado: os comprimentos de onda disponíveis são organizados de forma que o menos utilizado é escolhido. O objetivo dessa estratégia é balancear a carga de utilização dos comprimentos de onda. Porém, possui um desempenho menor que o Escolha Randômica. Para saber quem é o menos utilizado é necessário manter o número de vezes que os comprimentos de onda foram utilizados. Como essa é uma informação global da rede, isso traz um custo computacional adicional. Como consequência, esse método não é muito utilizado na prática;
- Mais utilizado: através de buscas semelhantes às da estratégia anterior, usa o comprimento de onda mais utilizado entre todos os disponíveis. É oposto ao Menos Utilizado, mas tem custo computacional e necessidade de informações extras semelhantes. Possui um desempenho, em termos de probabilidade de bloqueio, melhor que o Ordem Fixa e que o Menos Utilizado.

4.3.3. Solução conjunta do RWA

Nos itens anteriores vimos uma solução individual e isolada dos problemas de roteamento e atribuição de comprimento de onda. Entretanto, muitas vezes, uma escolha ruim de um comprimento de onda pode prejudicar uma escolha ótima de uma rota. A escolha ruim do comprimento de onda λ_i significa que o mesmo pode não garantir a propriedade de continuidade entre o nó origem e o nó destino, ou seja, ter uma grande probabilidade de bloqueio durante o caminho. Embora o algoritmo de roteamento encontre a melhor rota entre um par origem-destino, a escolha ruim de um comprimento de onda trará uma grande probabilidade de perda da rajada. Da mesma forma, uma escolha ruim de uma rota pode prejudicar uma escolha de um bom comprimento de onda.

Portanto, tentando evitar os problemas citados acima, pesquisadores propuseram resolver os dois problemas de forma conjunta (Strand et al, 2001). Neste caso, podemos calcular os custos da seguinte forma: suponha um *lightpath* com 10 enlaces e 10 comprimentos de onda. Para cada uma das 100 possibilidades de caminho é calculado um custo e é escolhido o menor. O custo computacional dessa abordagem é maior, porém traz uma boa melhora no desempenho de um modo geral.

Esse trabalho propõe uma escolha conjunta do comprimento de onda e da rota, para que seja feita uma escolha adequada. Essa abordagem fará uma escolha mais equilibrada, evitando que possíveis escolhas individuais prejudiquem o desempenho do algoritmo de RWA.

4.3.4. Problema de Recuperação

Casos de bloqueio de rajada são exceções nas requisições em redes OBS (Hsu et al, 2002). Isso pode ser verificado em simulações em vários artigos como os de Shi et al (2005), Huo et al (2004) e Ngo et al (2006). As taxas de probabilidade são geralmente baixas em situações normais. No entanto, a probabilidade existe e deve ser tratada.

Existem três possibilidades de ocorrer um bloqueio da rajada: por falta de tempo para processar o pacote de controle durante o caminho, por falta de comprimento de onda disponível e por queda de algum enlace do caminho escolhido.

No primeiro caso, o problema está mais relacionado ao cálculo do tempo de ajuste. Alguns artigos tratam do problema do cálculo do tempo de ajuste, como, por exemplo, o de Sousa et al (2008). Nosso foco não é esse problema, apesar dele também influenciar na probabilidade de bloqueio. Trataremos os outros dois casos.

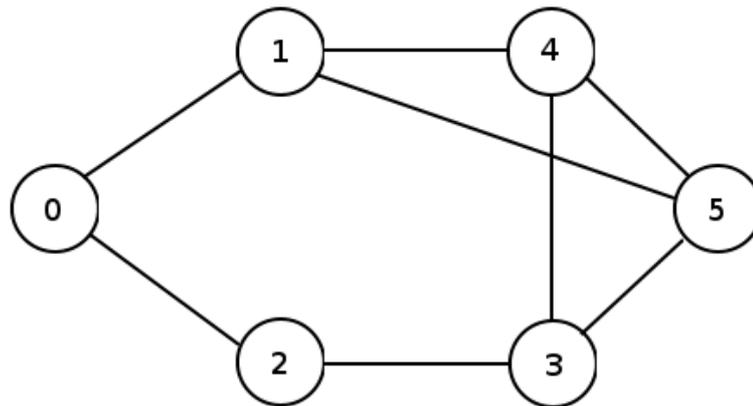


FIGURA 11: Cenário para o problema de recuperação

Quando falta comprimento de onda disponível, deve-se usar alguma estratégia para desviar o caminho (ou calcular um novo), de forma que a rajada não seja perdida. Na topologia da Figura 11, supondo que um pacote de controle foi enviado do nó 2 para o nó 1 pelo comprimento de onda λ_1 e a rota escolhida foi 2-3-4-1, caso o pacote chegue no nó 3 e não possua o comprimento de onda λ_1 disponível para o nó 4, o nó 3 tem que ser hábil o suficiente para desviar a rota para o nó 5 e deste seguir para o nó 1 também utilizando o λ_1 . Nesse caso, a rota mudaria dinamicamente para 2-3-5-1, por exemplo. Para esse exemplo, o número de saltos é o mesmo. Porém, podem existir casos em que o número de saltos seja maior. Nesses casos, a probabilidade do pacote de controle não ter tempo suficiente para reservar os recursos aumenta, pois o tempo de ajuste foi calculado antes do pacote de controle sair do nó de origem, portanto, não podendo ser recalculado. O ideal seria que o novo caminho calculado tivesse no máximo o mesmo número de saltos do restante do anterior, pois, o próprio tempo para computar o novo caminho poderia ser suficiente para causar a perda da rajada. Se o processamento do pacote de controle, num determinado nó intermediário onde aconteceu o bloqueio do recurso, demorar muito no cálculo de uma rota alternativa, provavelmente o tempo de ajuste não será suficiente e a rajada será perdida.

4.4. Roteamento Dinâmico com ACO

Como já citado, existem muitas técnicas de computação inteligente aplicadas para melhorar o desempenho de redes. Muitos trabalhos na literatura, como alguns dos trabalhos citados no capítulo 2, mostram que a aplicação de ACO tem se mostrado eficiente para tal tarefa.

Juntamente com a eficiência do ACO, os problemas de RWA em redes OBS motivaram essa dissertação a propor o AntOBS. Este é um algoritmo baseado em ACO que trata o problema do roteamento e atribuição de comprimento de onda em redes OBS dinamicamente. Os detalhes de como essa técnica foi aplicada no problema de RWA em redes OBS são mostrados no próximo capítulo. Os resultados das simulações são apresentados no capítulo 6.

Capítulo 5

AntOBS

Algoritmos como o de Dijkstra (Tanaenbaum, 2003) são algoritmos de roteamento não adaptáveis, dessa forma, não conseguem se adequar a alterações da rede. Para viabilizar um processo mais eficiente em tempo real, características da computação inteligente podem ser empregadas para melhorar o desempenho em redes de comunicação. Várias técnicas têm sido aplicadas. Entre essas técnicas está ACO, Algoritmos Genéticos, Redes Neurais, entre outras. A técnica ACO será usada nessa dissertação e, nesta seção, serão mostrados os motivos para essa escolha.

Os pesquisadores têm apresentado diversos modelos baseados na inteligência coletiva de formigas para resolver problemas de otimização, como o problema do caixeiro viajante assimétrico (Dorigo et al, 1997), de coloração de grafos (Israel et al, 2000), roteamento de veículos (Dorigo et al, 1999) e roteamento em redes de comunicações (Caro et al, 1998).

Essa dissertação tem como objetivo a criação e aplicação de um algoritmo distribuído para resolver o problema do RWA dinâmico em redes ópticas de comutação por rajada, buscando, com isso, diminuir a probabilidade de bloqueio de recursos em redes OBS, bem como aumentar o grau de utilização dos recursos da rede óptica. Além disso, é também uma meta, obter dinamicidade nos momentos em que ocorre bloqueio, isto é, agilizar o re-roteamento quando não for possível reservar recursos na rede óptica. Essa dinamicidade é conseguida através da utilização de ACO, e o foco é atuar em cima de dois problemas: RWA e recuperação de bloqueio da rajada. Será evitada a utilização de *buffers* e conversores de comprimento de onda devido ao alto custo de utilização dos mesmos (Silveira et al, 2003).

Será utilizado o protocolo de sinalização JET (Yoo et al, 1997), apesar de ser possível a aplicação de outros protocolos, como por exemplo, o JIT (Wei et al, 2000). Porém, o primeiro foi escolhido por sua característica principal: a reserva atrasada que aumenta a utilização dos recursos da rede. Em trabalhos futuros, outros protocolos poderão também ser analisados. De acordo com os fundamentos do JET (Yoo et al, 1997), será utilizado o método de reserva *one-way*.

Em redes WDM, como explicado anteriormente, o problema do RWA pode ser dividido em duas partes: roteamento e atribuição do comprimento de onda (Oliveira et al, 2005). Em redes ópticas de rajadas (OBS), outra questão importante para a sobrevivência da rede óptica é a recuperação no caso de bloqueio da reserva dos recursos (Huo et al, 2004). Esse bloqueio pode ser causado pela falta de recursos na rede ou pela queda de algum enlace ou comutador (*Optical Cross Connect - OXC*). Nesses casos, mudanças de rotas são necessárias, tendo em vista que a dinamicidade e a rapidez na solução do problema são vitais para o bom desempenho da rede óptica.

Pesquisas mostram que várias soluções para o problema do RWA foram apresentadas e estas utilizam diversos algoritmos e meta-heurísticas como em Ngo et al (2006), Yang et al (2006) e Triay et al (2010). Esse trabalho propõe a aplicação de colônia de formigas (ACO) para resolver os problemas citados. Existem várias características que motivaram a utilização dessa estratégia:

- Possibilidade de atualização dinâmica (Controle Dinâmico da Rede);
- Informação local suficiente para a execução da tarefa de cada formiga (utilizamos dois tipos de tabelas de roteamento e comprimentos de onda disponíveis, em cada nó);
- Existe um compartilhamento de informações da rede sem causar um grande *overhead*;
- Não exige muito esforço computacional no processamento das formigas em cada nó;
- Ainda é possível coletar e trocar informações adicionais da rede através das rotas das formigas. Ex.: Probabilidade de bloqueio.

O comportamento da colônia de formigas pode ser visto como a seguir. Inicialmente, uma formiga sai do ninho em busca de alimento sem uma orientação determinada. À medida que percorre seu caminho, deposita determinadas quantidades de um hormônio, feromônio. Isso possibilitará uma futura identificação do caminho por outras formigas nas próximas buscas por alimento. As formigas seguintes que saírem do ninho levarão em consideração os níveis de feromônio que já foram depositados e ainda atualizarão o nível de feromônio do caminho depositando mais hormônio. Quanto mais formigas passarem por um determinado caminho, maior será o nível de feromônio e maior a possibilidade de outras formigas passarem por lá.

Como consequência desse processo, os menores caminhos entre o ninho e o alimento terão os maiores níveis de feromônio, ou seja, as formigas acharão, com o tempo, os menores caminhos. Para possibilitar a diversificação dos caminhos, os níveis de hormônio diminuem com o tempo, tornando possível uma mudança de caminho (Corne et al, 1999).

Na nossa abordagem, as formigas são caracterizadas por pacotes que alimentam as tabelas de roteamento, e estas armazenam os níveis de feromônio nas rotas. Nesse caso, o nível de feromônio representa a probabilidade de sucesso da rajada tendo escolhido o nó, ou seja, quanto maior o nível de feromônio, maior será a probabilidade de sucesso da rajada e menor a probabilidade de bloqueio. As formigas são geradas nos nós e são enviadas para destinos escolhidos aleatoriamente. À medida que seguem o caminho para um determinado destino, vão atualizando o nível de feromônio em cada nó. Foram criados dois tipos de formigas a EAnt e a MAnt. A primeira é responsável por criar as tabelas de roteamento e a segunda por manter os níveis de feromônio atualizados. Os detalhes do comportamento de cada tipo serão explicados nas seções seguintes.

Importante perceber que as formigas atualizam as tabelas de roteamento com a própria movimentação. No entanto, quem utiliza essas informações é o pacote de controle. Este, ao chegar ao nó, examina a tabela e, levando em consideração as informações nela contida, escolhe o próximo salto. Quanto mais formigas passarem naquele caminho, maior será a probabilidade dos pacotes de controle utilizarem esse

caminho. Caso as formigas possam compartilhar o mesmo canal dos dados, poderiam aumentar a probabilidade de bloqueio, pois os canais poderiam estar ocupados com as formigas, além dos dados.

O percurso das formigas é feito *out-of-band*, ou seja, fora do plano de dados. Isso torna a movimentação das formigas independente dos dados da rede. Além disso, evita que as formigas gerem um tráfego extra nos canais de dados e, com isso, prejudiquem o desempenho da rede.

Teoricamente, em ACO, as formigas saem do ninho em busca de alimentos. E quando os acham, retornam ao ninho reforçando o caminho com o feromônio. No caso dessa dissertação, a formiga só faz o percurso entre dois nós uma vez. Nesse caso, a atualização é feita apenas percorrendo o caminho origem-destino. Portanto, a formiga não fará o percurso de retorno. O objetivo dessa estratégia é evitar o *overhead* dos dois percursos.

5.1 Modelo Utilizado

O modelo pode ser definido como um grafo $G = (V, E)$, onde V representa o conjunto de nós da rede e E representa o conjunto de arestas ou enlaces da rede. Cada enlace $e_{i,j}$ representa uma ligação entre o nó i e o nó j .

A seguir serão apresentadas algumas definições sobre os mecanismos básicos necessários para o funcionamento da colônia de formigas.

5.1.1 A Estrutura das Formigas

O protocolo AntOBS possui dois tipos de formigas. A primeira delas é a Formiga de Inicialização (IAnt). Esta formiga é lançada durante a fase de inicialização da rede óptica. Possui o objetivo de comunicar aos outros nós da rede a existência do próprio nó de origem do lançamento. Inicialmente, os nós apenas conhecem seus vizinhos imediatos, sendo necessário, portanto, conhecer todos os outros nós da rede para que o protocolo possa agir de forma correta. Essa formiga possui um funcionamento baseado no comportamento de um pacote broadcast. A IAnt possui a seguinte estrutura: código identificador, nó de origem e o número de saltos.

O segundo tipo é a Formiga de Manutenção (MAnt). Essa formiga deve manter e descobrir novas rotas para o nó de origem, sendo lançada com um destino escolhido aleatoriamente. A MAnt carrega informações sobre o caminho e sobre o destino com o objetivo de marcar as arestas ao longo do percurso com feromônio. Ao chegar ao nó, este interpreta o endereço de origem da formiga MAnt como endereço de destino, ou seja, o nó irá atualizar a tabela de roteamento do nó corrente tendo o nó de origem do MAnt como destino.

As informações levadas por uma formiga MAnt são: código identificador, nó de origem, nó de destino, pilha, TTL (*Time-To-Live*) e máscara de bits. O TTL é utilizado para evitar que a formiga fique na rede indefinidamente. A partir de um valor inicial, o TTL é diminuído de uma unidade a cada salto da formiga. Caso o TTL tenha valor 0, a formiga é eliminada da rede. A pilha é utilizada para armazenar o caminho percorrido

pela formiga. A cada salto o nó verifica se existe um ciclo no caminho, caso encontre, a formiga é eliminada. A máscara de bits tem a função de levar os comprimentos de onda disponíveis para aquela rota.

Além do caso em que o TTL é zero, existem mais três situações para evitar que as formigas MAnt sejam retiradas da rede: quando chegam ao destino, ou quando não acham comprimento de onda disponível ou quando detectam um *loop* no caminho. Nesse último caso, as formigas utilizam as informações da pilha que armazena cada nó já visitado.

5.1.2 Escolha do Próximo Nó no Caminho

A equação abaixo ilustra a probabilidade da formiga MAnt, estando no nó i , escolher o nó j como próximo salto do caminho.

$$p_{ij}^k = \begin{cases} r \cdot \frac{[\tau_{ij}]^\alpha}{[d_{ij}]^\beta}, & \text{se } j \in N_i^k ; \\ 0, & \text{se } j \notin N_i^k . \end{cases} \quad (5.1)$$

Considere r uma constante que seria escolhida aleatoriamente com o objetivo de dar diversidade à solução. N_i^k o conjunto de vizinhos do nó i , e τ_{ij} e d_{ij} os níveis de feromônio e o custo associado ao enlace e_{ij} . O custo associado a cada enlace pode ter várias métricas, como por exemplo, a distância entre os nós. Porém, nesse caso, assumiremos que o custo será o número de colisões ocorridas quando o nó j foi escolhido. Essa métrica foi escolhida para levar em consideração o histórico de colisões naquele enlace. Os valores de α e β são constantes responsáveis por decidir o grau de importância dos termos da equação. Essas constantes são parâmetros no algoritmo AntOBS.

Para evitar estagnação em soluções sub-ótimas, foi criada uma segunda regra de transição da formiga. A segunda regra seria a escolha randômica. Nesse caso, a formiga decidiria o próximo salto sem levar em consideração o nível de feromônio dos enlaces. Sendo assim, para definir o próximo salto, primeiramente a formiga teria que decidir qual regra usar. Essa escolha é feita estocasticamente, onde cada regra teria um índice de prioridade. Isso foi acrescentado na solução para haver diversificação na escolha das rotas.

5.1.3 Atualização do Feromônio

Após receber uma formiga MANT, o nó faz as verificações das informações que esta carrega, e, em seguida, o mesmo atualiza o nível de feromônio associado ao enlace. A equação abaixo ilustra o reforço de feromônio adicionado. Assuma que $\Delta\tau^k$ é uma

constante e representa a atualização do feromônio. Devido à importância de tal constante, esta é um dos parâmetros do sistema.

$$\tau_{ij} \leftarrow \tau_{ij} + \Delta\tau^k \quad (5.2)$$

Para atualizar a tabela, o nó corrente assume o nó de origem da formiga MAnt como destino e altera o registro da tabela correspondente ao mesmo. A chegada da formiga ao nó corrente implica que existe uma rota do nó corrente ao nó de origem da formiga MAnt.

5.1.4 Evaporação

Seguindo o modelo natural, a cada período de tempo os níveis de feromônio devem ser evaporados. Isso permite que as escolhas sejam mais diversificadas e que novas soluções sejam encontradas. A equação abaixo mostra o mecanismo de evaporação.

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij}, \quad \forall (i, j) \in A \quad (5.3)$$

onde ρ é uma constante responsável pelo nível de evaporação do feromônio nos enlaces. Juntamente α e β , ρ é um também é um parâmetro do algoritmo AntOBS.

5.2 Detalhes do Protocolo AntOBS

AntOBS possui um algoritmo que possibilita diversificar os caminhos entre um par de nós. Inicialmente, os nós recebem informações sobre os outros nós da rede através da movimentação das formigas IAnt. Tomando como base essas informações, as tabelas de roteamento são criadas. A Tabela 3 abaixo ilustra a estrutura de uma tabela de roteamento do protocolo. De acordo com a coluna de destino, os vizinhos possuem as respectivas probabilidades, representadas pelo feromônio, de serem escolhidos. Além da coluna de feromônio, a tabela possui o próximo salto e o número de saltos até o destino.

Destinos	Próximo Salto	Nº de Saltos	Feromônio
N_1	0	3	P^1_1
N_2	1	1	P^1_2
...
N_n	3	2	P^1_n

TABELA 3: Tabela de Roteamento/Feromônio

Após a ação inicial da formigas IAnt, a rede estará estabelecida com as tabelas de roteamento em cada nó devidamente preenchida com os valores iniciais. Assim, a manutenção das tabelas fica por conta da movimentação das formigas MAnt. Essa operação é feita de acordo com as equações de deposição e evaporação mostradas nas seções anteriores.

5.3 Criação e Manutenção das Rotas

5.3.1 Fase de Inicialização

Logo após o início da simulação, cada nó i envia uma formiga IAnt por *broadcast* para seus vizinhos. Essa operação cria novos registros e valores das tabelas de roteamento dos nós. Um nó j pode tomar duas decisões quando recebe uma determinada IAnt. Para isso, o nó leva em consideração o campo identificador da formiga. Se for a primeira vez que a formiga passa pelo nó, um novo registro é inserido na tabela de roteamento. O nó j interpreta o endereço de origem da IAnt como endereço de destino e o endereço do nó anterior como próximo salto, e então inicializa o valor do feromônio do enlace e_{ji} . Se não for a primeira vez, a formiga é excluída e nada é feito. Nesse caso, o nó está recebendo uma duplicação, logo, ignora e exclui a formiga. Por fim, o nó atualiza o valor do campo número de saltos e a formiga IAnt é reenviada para seus vizinhos.

A Figura 12 abaixo ilustra o comportamento das formigas IAnt. Na Figura 12 (a), o nó 0 está disparando duas formigas IAnt, uma para cada vizinho. Estes, ao receberem, primeiramente, verificam se o identificador da formiga é válido, ou seja, se é a primeira vez que esta passa pelo nó. Logo em seguida, criam uma nova entrada na tabela de roteamento tendo como destino o nó 0, além de guardarem o enlace de chegada da formiga como enlace de saída para o nó 0. Depois enviam a formiga para seus vizinhos, exceto para o próprio nó 0. Na Figura 12 (b), os nós 1 e 2 receberam novamente a formiga vinda do nó 0, mas enviada por outro nó intermediário. Esses nós excluem a formiga, pois o seqüencial da formiga não é mais válido. Quando os nós 3 e 4 receberem a formiga, respectivamente do nó 1 e 2, irão fazer a verificação do identificador da formiga, criarão uma nova entrada na tabela de roteamento tendo como destino o nó 0 e acrescentarão o enlace de chegada da formiga como enlace de saída para o nó 0.

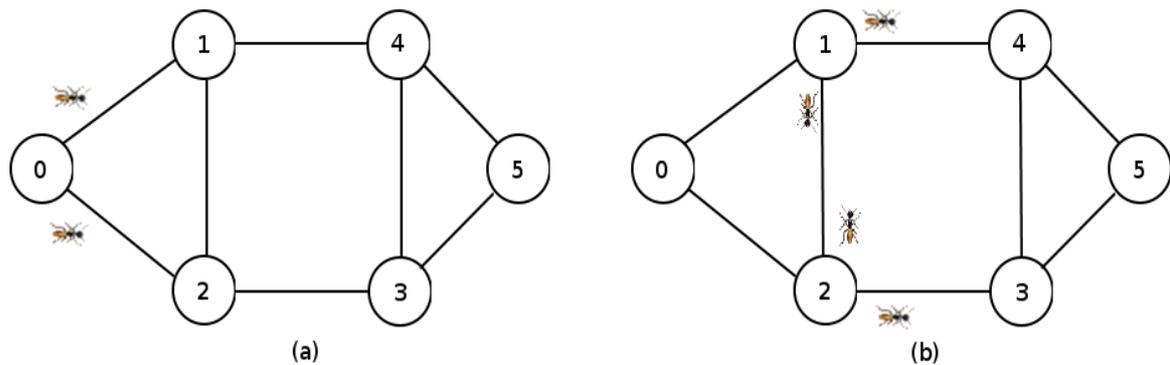


FIGURA 12: Comportamento da formiga IAnt

Após o fim dessa fase, todas as rotas terão a mesma probabilidade, portanto, terão o mesmo nível de feromônio. A partir daqui, é necessário que as rotas sejam otimizadas. Isso é conseguido através do comportamento das formigas MAnt.

5.3.2 Fase de Manutenção

As formigas MAnt são responsáveis por manter as rotas atualizadas. Estas circulam pela rede atualizando os níveis de feromônio dos enlaces. Periodicamente, os nós enviam as formigas MAnt com destinos aleatórios. A fase de inicialização é importante para isso. É nela que os nós ficam sabendo da existência dos outros nós da rede, através do funcionamento das formigas IAnt, como explicado na seção anterior. A periodicidade do envio das formigas é um parâmetro do sistema, juntamente com α , β e ρ , descritos anteriormente, que devem ser regulados para evitar que o número de formigas afete o desempenho da rede.

Quando um determinado nó i recebe uma formiga MAnt, ele pode ser tanto o destino da formiga como um nó intermediário do caminho até o destino. Os passos, em cada um dos casos, são descritos a seguir.

- Nó i não é o destino da formiga MAnt
 - Verifica o TTL. Se o mesmo tiver valor 0, a formiga será excluída
 - Se não houver, cria entrada na tabela de roteamento
 - Atualiza o nível de feromônio
 - Escolhe um destino através da regra de transição.
 - Incrementa o número de saltos
 - Decrementa o TTL
 - Re-envia a formiga para o próximo salto, escolhido pelo passo acima
- Nó i é o destino da formiga MAnt
 - Atualiza o nível de feromônio
 - Exclui a formiga

A Figura 13 abaixo ilustra o funcionamento da formiga MAnt. De acordo com a frequência parametrizada pelo algoritmo, o nó envia uma formiga com o destino aleatório. Na figura 13(a), o nó zero, após gerar a formiga, escolheu aleatoriamente o nó 5 como destino. De acordo com a tabela de roteamento e com os níveis de feromônio, o nó 0 escolhe probabilisticamente o próximo salto através da equação 5.1. No caso, o enlace e_1 para o nó 1 foi escolhido. O nó 1 recebe a formiga e, após verificar o TTL, vendo que não é o destino da formiga, atualiza a própria tabela de roteamento tendo o nó 0 como destino e o enlace e_1 como saída para o nó 0. Além disso, o nível de feromônio da entrada é acrescido. Depois disso, o nó 1 escolhe probabilisticamente, de acordo com o nível de feromônio, o próximo salto para a formiga através da equação 5.1. No caso, o enlace e_3 para o nó 4 foi escolhido. Na Figura 13(b), o nó 4 recebe a formiga e se comporta da mesma forma que o nó 1. Atualiza a própria tabela de roteamento tendo o nó 0 como destino e enlace e_3 como saída. Além de atualizar o nível de feromônio. O nó 4 escolhe probabilisticamente, de acordo com o nível de feromônio, o nó 5 como próximo salto da formiga. Na Figura 13(c), o nó 5 recebe a formiga e verifica que é o destino da mesma, atualiza a própria tabela de roteamento para o nó 0, e então elimina a formiga.

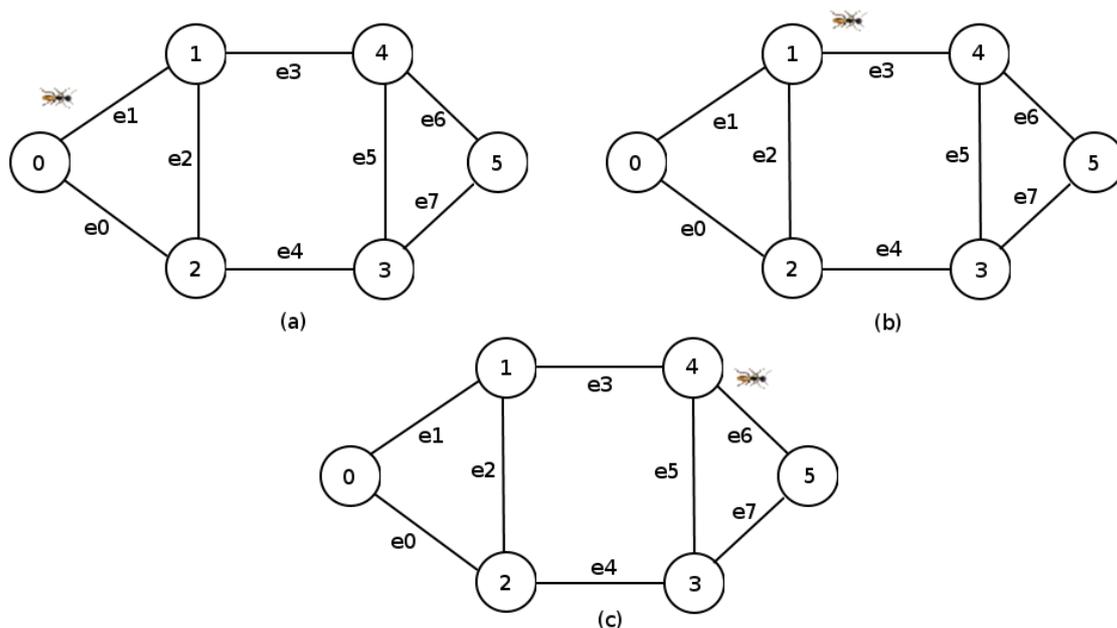


FIGURA 13: Comportamento da formiga MAnt

A forma de escolha do próximo nó e a atualização do feromônio atuam de acordo com as seções 5.1.2 e 5.1.3, respectivamente. O comportamento da formiga MAnt difere um pouco do comportamento da formiga real, pois, esta percorre o caminho ninho-alimento e volta pelo caminho inverso. No caso da formiga MAnt, esta apenas faz um caminho. Isso diminui o *overhead* causado pela utilização da formiga na rede.

Uma importante característica do protocolo AntOBS é que o pacote de controle é apenas um usuário dos dados gerados pelas formigas. Ou seja, as formigas criam e atualizam rotas, enquanto que os pacotes de controle utilizam essas rotas para tentar reservar os recursos da rede.

O processo de manutenção das rotas pelas formigas acontece paralelamente ao uso normal da rede. O principal benefício disso é a garantia que as rotas estarão sempre de acordo com o estado atual da rede.

5.3.3 Reserva dos Recursos e Envio da Rajada

As formigas iniciam e atualizam as tabelas de roteamento da rede. Porém, o pacote de controle utilizará essas rotas criadas para reservá-las para a passagem da rajada. Os passos do comportamento do pacote de controle e da rajada no AntOBS são mostrados a seguir. Primeiramente, os passos abaixo ilustram o comportamento no roteador de borda.

1. Os dados das redes clientes são armazenados no roteador de borda para formar a rajada;
2. A montagem é concluída quando o tempo máximo de espera estourar ou a rajada chegar ao tamanho máximo;
3. Nesse momento, o pacote de controle é criado. Alguns passos são necessários antes de enviá-lo;
 - a. De acordo com o destino dos dados, calcular o tempo de ajuste;
 - b. De acordo com o nível de feromônio, escolher o próximo salto para o destino. Nesse caso, a escolha é probabilística, o que dá uma possibilidade de diversificação da solução;
 - c. Logo após, verificar qual o comprimento de onda disponível;
 - d. Por fim, o pacote de controle é enviado;
4. Inicia um temporizador para o tempo de ajuste;
5. Quando finaliza o temporizador, a rajada é enviada utilizando WDM.

Como explicado anteriormente, após sair do roteador de borda, a rajada é apenas comutada até o nó de destino. Caso não seja possível reservar os recursos em algum nó da rede, a rajada é perdida. Os recursos podem não ser reservados pela falta de comprimento de onda disponível ou pelo tempo de ajuste insuficiente.

Os passos abaixo ilustram as possibilidades quando o pacote de controle chega a um determinado nó i . Esse nó i pode ser o destino ou não.

- O nó i não é o destino do pacote de controle
 1. De acordo com o destino, escolhe o próximo salto na tabela de roteamento levando em consideração o nível de feromônio. Nesse caso, a escolha não é probabilística. Devido ao cálculo do tempo de ajuste que ocorre antes do envio da rajada, o pacote de controle não deve mudar o número de saltos do caminho. Isso poderia causar o bloqueio da rajada;

2. De acordo com o próximo salto, verifica-se a possibilidade de reservar o comprimento de onda para a rajada através do algoritmo *First-Fit*:
 - Se for possível
 - Efetuar a reserva;
 - Enviar o pacote de controle;
 - Caso não seja possível
 - Retornar ao passo 1 para escolher uma segunda opção para o próximo salto;
- O nó i é o destino do pacote de controle
 1. A rajada é recebida e o sinal é demultiplexado;
 2. Os dados são enviados para as redes cliente.

A Figura 14 abaixo ilustra os passos descritos anteriormente. No caso, o roteador de borda A envia dados para o roteador de borda E. Para isso, o pacote de controle reserva os recursos pelos seguintes nós de núcleo: B-C-D. Os dados são acumulados e multiplexados em A, e, quando recebidos, são demultiplexados em E.

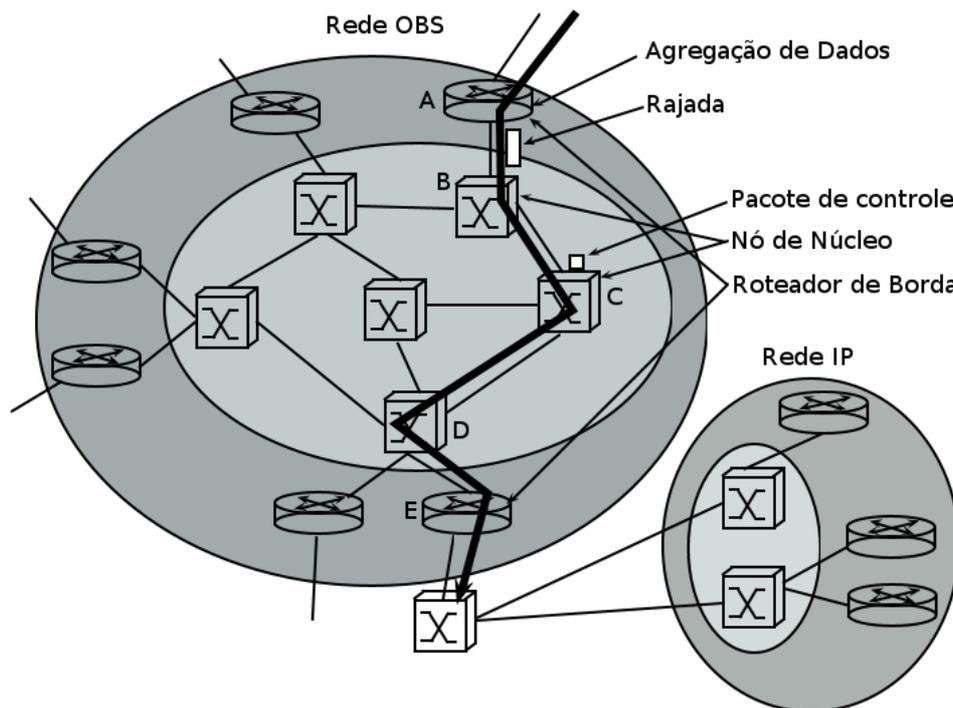


FIGURA 14: Exemplo do envio da rajada e reserva dos recursos

Ainda no exemplo ilustrado na Figura 14, quando a rajada é montada, o roteador de borda A escolhe o enlace mais adequado para chegar ao destino. Essa escolha é feita de forma não probabilística, ou seja, escolhe o enlace com menor probabilidade de

bloqueio. Depois disso, o pacote de controle é enviado, antes da rajada, para reservar os recursos. A rajada é enviada logo após o término do tempo de ajuste. Esses passos são repetidos nos nós de núcleo B, C e D. A rajada é comutada nesse caminho até chegar ao destino, isto é, o roteador de borda E onde os dados são demultiplexados e enviados para os destinos.

De acordo com os passos acima, quando um pacote de controle não consegue reservar o comprimento de onda, o nó tenta buscar uma segunda opção para o próximo salto do pacote de controle. Nesse caso, devido a esse processamento extra, existe uma grande possibilidade da rajada ser perdida, pois, como o tempo de ajuste foi calculado anteriormente no roteador de borda e não pode ser mais alterado, pode ser que este não seja suficiente para que a rajada alcance o pacote de controle.

No entanto, como não haveria comprimento de onda disponível na primeira escolha e isso acarretaria a perda da rajada, a segunda tentativa é uma chance a mais, embora que pequena, de evitá-la.

5.4 Exemplo de Funcionamento do Protocolo

Considere a Figura 15 abaixo a topologia da rede óptica que será utilizada para mostrar o comportamento do protocolo AntOBS. No exemplo, cinco nós formam a rede óptica e será ilustrado a seguir o comportamento das formigas IAnt e MAnt do pacote de controle (BCP) e da rajada.

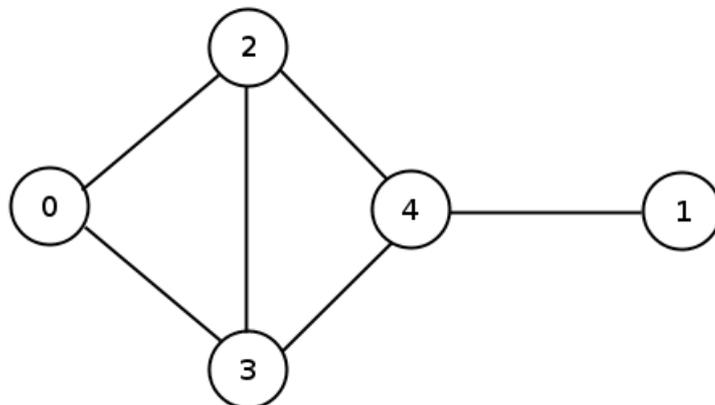


FIGURA 15: Topologia utilizada no exemplo da seção

A Tabela 4 abaixo mostra as tabelas de roteamento dos nós inicialmente. A partir dessas tabelas, as formigas IAnt irão iniciar as rotas entres os nós.

Nó 0			
Dest	NHop	Hop	Fer
2	2	0	1
3	3	0	1

Nó 1			
Dest	NHo	Hop	Fer
4	4	0	1

Nó 2			
Dest	NHop	Hop	Fer
0	0	0	1
3	3	0	1
4	4	0	1

Nó 3			
Dest	NHop	Hop	Fer
0	0	0	1
2	2	0	1
4	4	0	1

Nó 4			
Dest	NHop	Hop	Fer
2	2	0	1
3	3	0	1
1	1	0	1

Tabela 4: Tabelas de roteamento iniciais

Os quatro campos da tabela de roteamento acima representam, na ordem, destino, próximo salto, número de saltos, nível de feromônio. O valor inicial do nível de feromônio é 1.

As formigas IAnt irão achar novos destinos para as tabelas anteriores. Cada nó gera uma formiga IAnt e passa para seus vizinhos. Esses, ao receberem a formiga IAnt, verificam se já possuem um registro para o nó de origem da formiga. Caso não possuam, criam um registro na tabela. Cada vizinho passa a formiga IAnt para os seus respectivos vizinhos. Esse processo segue até não haver nós a serem visitados. A formiga IAnt leva um identificador para impedir que um nó processe duas vezes a mesma formiga. A seguir, a Tabela 5 ilustra o resultado da movimentação das formigas IAnt.

Nó 0			
Dest	NHop	Hop	Fer
2	2	0	1
3	3	0	1
4	2	1	1
4	3	1	1
1	2	2	1
1	3	2	1

Nó 1			
Dest	NHo	Hop	Fer
4	4	0	1
2	4	1	1
3	4	1	1
0	4	2	1

Nó 2			
Dest	NHop	Hop	Fer
0	0	0	1
3	3	0	1
4	4	0	1
0	3	1	1
4	3	1	1
1	4	1	1
1	3	2	1

Nó 3			
Dest	NHop	Hop	Fer
0	0	0	1
2	2	0	1
4	4	0	1
4	2	1	1
1	4	1	1
1	2	1	1

Nó 4			
Dest	NHop	Hop	Fer
2	2	0	1
3	3	0	1
1	1	0	1
0	2	1	1
0	3	1	1

Tabela 5: Tabelas de roteamento após a movimentação das formigas IAnt

A seguir os passos que mostram o comportamento da formiga MAnt. Essa formiga possui a tarefa de manter atualizadas as tabelas de roteamento. Cada nó cria uma formiga MAnt periodicamente e a escolha do destino da formiga é aleatória.

Suponha que o acréscimo da atualização do nível de feromônio seja 0,1 e que o nó 0 irá mandar uma formiga MAnt para o nó 1. No nó 0, a formiga MAnt deve escolher o próximo salto para chegar ao nó 1. Nesse caso, a formiga tem duas opções: o nó 2 e o nó 3. A escolha é feita estocasticamente levando em consideração o nível de feromônio dos caminhos, ou seja, o campo Fer da tabela de roteamento. Como para os dois nós o nível de feromônio é o mesmo e igual a 1, então os dois irão concorrer com iguais probabilidades de serem escolhidos. Suponha que o nó 2 foi escolhido, o nó 0 envia a formiga para o nó 2. Ao chegar ao nó 2, a formiga atualiza o registro da tabela de roteamento do nó 2 que possui o nó 0 como destino. Essa atualização é representada pelo acréscimo de 0,1 ao nível de feromônio. O processo de escolha do próximo salto será repetido no nó 2. Supondo que a formiga percorreu o caminho 0-2-4-1, as tabelas de roteamento ficaram da forma como mostrado na Tabela 6 abaixo.

Nó 0			
Dest	NHop	Hop	Fer
2	2	0	1
3	3	0	1
4	2	1	1
4	3	1	1
1	2	2	1
1	3	2	1

Nó 1			
Dest	NHo	Hop	Fer
4	4	0	1
2	4	1	1
3	4	1	1
0	4	2	1.1

Nó 2			
Dest	NHop	Hop	Fer
0	0	0	1.1
3	3	0	1
4	4	0	1
0	3	1	1
4	3	1	1
1	4	1	1
1	3	2	1

Nó 3			
Dest	NHop	Hop	Fer
0	0	0	1
2	2	0	1
4	4	0	1
4	2	1	1
1	4	1	1
1	2	1	1

Nó 4			
Dest	NHop	Hop	Fer
2	2	0	1
3	3	0	1
1	1	0	1
0	2	1	1.1
0	3	1	1

Tabela 6: Tabelas de roteamento após a movimentação de uma formiga MAnt

Como citado anteriormente, as formigas MAnt são geradas periodicamente de cada nó da rede para destinos aleatórios. O próximo comportamento a ser ilustrado é o do BCP.

Suponha que o nó 0 montou uma rajada e quer enviá-la para o nó 1 e que as tabelas de roteamento estão ilustradas na Tabela 7 abaixo. Essa tabela está representando um instante qualquer da rede, ou seja, após a ocorrência dos processos de atualização e evaporação dos níveis de feromônio.

O BCP, no nó 0 deve escolher qual a melhor rota até o nó 1. Porém, este só tem conhecimento da tabela de roteamento do nó 0. Para escolher a melhor rota, o BCP escolhe o caminho para o nó 1 com melhor nível de feromônio. No caso do nó 0, existem duas opções: o nó 2 e nó 3. Como o nível de feromônio do nó 3 é maior, então este é escolhido como próximo salto. O nó escolhe o comprimento de onda, através do método FIRST-FIT, reserva os recursos e manda o BCP para o nó 3. Chegando ao nó 3, o BCP é avaliado para fazer as reservas necessárias. Se houver recursos, o nó 3 deve escolher o próximo salto do BCP. Nesse caso, existem duas opções para o nó 1: o nó 2 e o nó 4. Porém, como o nó 4 possui maior nível de feromônio, este será escolhido como o próximo salto e o BCP é enviado. Ao chegar ao nó 4, o BCP é novamente processado e se houver recursos, a escolha do próximo salto é realizada. No nó 4, existe apenas uma opção: o nó 1. Nesse contexto, o BCP chega ao nó 1 e fecha o caminho para a rajada que vem em seguida. Estando todo o caminho reservado pelo BCP, a rajada é apenas comutada nos nós do caminho.

Nó 0			
Dest	NHop	Hop	Fer
2	2	0	1
3	3	0	0.9
4	2	1	0.9
4	3	1	0.9
1	2	2	1
1	3	2	1.1

Nó 1			
Dest	NHo	Hop	Fer
4	4	0	1
2	4	1	0.9
3	4	1	1
0	4	2	1.1

Nó 2			
Dest	NHop	Hop	Fer
0	0	0	1.1
3	3	0	1
4	4	0	1
0	3	1	1
4	3	1	1.1
1	4	1	1
1	3	2	1

Nó 3			
Dest	NHop	Hop	Fer
0	0	0	1
2	2	0	1
4	4	0	0.9
4	2	1	0.9
1	4	1	1.1
1	2	1	0.9

Nó 4			
Dest	NHop	Hop	Fer
2	2	0	1
3	3	0	0.9
1	1	0	1
0	2	1	1.1
0	3	1	1

Tabela 7: Tabelas de roteamento para a movimentação do BCP

Capítulo 6

Simulações e Resultados

Este capítulo apresenta os resultados obtidos nas simulações feitas para testar o desempenho do algoritmo de roteamento AntOBS. Para isso, foram feitas comparações com o algoritmo que chamaremos de OBS Puro, um algoritmo de roteamento em redes OBS que não é baseado em colônia de formigas.

Para efetuar as simulações foi utilizado um computador cuja configuração possui um processador Core 2 Duo e 2Gb de memória RAM. Esse trabalho ilustra apenas uma parte das simulações feitas, mas que permitem perceber o bom desempenho do AntOBS.

Será utilizado o Network Simulator 2 (NS-2) cuja versão é a 2.34, a versão mais atual. O NS-2 é um simulador baseado em eventos discretos, muito utilizado em pesquisas de redes de computadores.

Os resultados exibidos foram formados pela média de 10 simulações feitas para cada ponto do cenário com intervalo de confiança de 95%. Isso garante uma maior confiabilidade dos resultados. O tráfego de dados gerado para cada simulação, ou seja, os pedidos de conexão ou chamadas à rede chegam de forma dinâmica. Portanto, o processo de geração de tráfego é estocástico e, para essa dissertação, segue a distribuição de Pareto.

A quantidade de comprimentos de ondas utilizada nas simulações varia entre seis e doze comprimentos de onda em cada enlace, dependendo do cenário. Estes serão detalhados nas seções seguintes.

Os parâmetros do algoritmo foram definidos de acordo com a Tabela 8 abaixo. O parâmetro α significa o grau de importância do nível de feromônio na equação de escolha do próximo salto da formiga. O parâmetro β significa o grau de importância do número de colisões na equação de escolha do próximo salto da formiga. O parâmetro ρ representa o nível de evaporação do algoritmo. O parâmetro Δ representa a quantidade de feromônio adicionada na atualização com a passagem da formiga. O último parâmetro é a frequência de criação das formigas MAnt em cada nó da rede óptica.

Parâmetro	Valor
α (Importância do nível de feromônio)	0.5
β (Importância do número de colisões)	0.5
ρ (Grau de evaporação)	0.4
Δ (Acréscimo da atualização do feromônio)	0.6
Frequência de criação das formigas MAnt	0.6 s

TABELA 8: Parâmetros do algoritmo AntOBS

Durante as simulações foram utilizadas duas topologias. As figuras 16 e 17 ilustram tais topologias. A Figura 16 apresenta uma topologia de uma pequena rede de teste composta por seis nós. A partir de agora, essa topologia será chamada de Topologia 1. A Figura 17 apresenta a topologia da rede NSFNET composta por treze nós e essa será a Topologia 2.

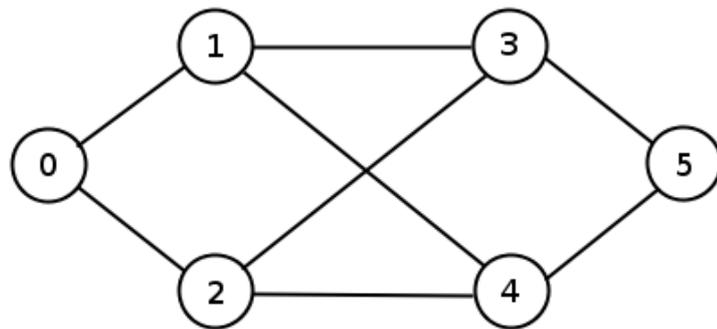


FIGURA 16: Topologia de teste

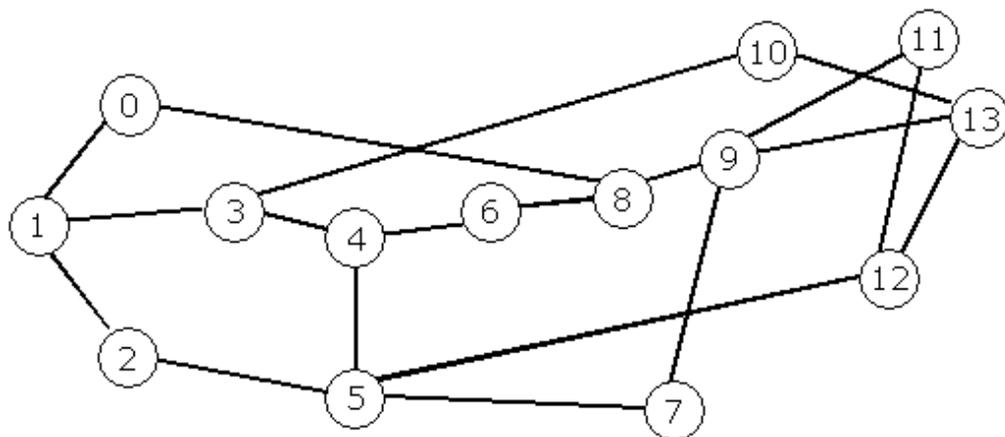


FIGURA 17: Topologia da rede NSFNET

As métricas utilizadas para analisar as simulações foram a probabilidade de bloqueio da rajada e o *overhead* gerado pela adição das formigas na rede óptica. A probabilidade de bloqueio é calculada através da divisão entre o número de rajadas perdidas pelo número de total de rajadas enviadas.

Como foi explicado nos capítulos anteriores, o principal objetivo de um algoritmo de RWA é encontrar uma rota e um comprimento de onda que satisfaça as requisições dos clientes. Portanto, é importante que tal algoritmo faça essas escolhas de forma a minimizar a quantidade de rajadas perdidas, ou seja, minimizar a probabilidade de perda da rajada. Por isso, a probabilidade de bloqueio é a métrica mais utilizada para calcular o

desempenho de um algoritmo de RWA. As seções seguintes detalham os resultados e as análises das simulações para cada topologia utilizada.

6.1. Topologia 1

Para fazer os primeiros testes com o AntOBS, foi usada a rede cuja topologia é representada pela Figura 16, onde foram utilizados dois cenários. O que muda em cada cenário é a quantidade de comprimentos de onda disponíveis em cada enlace. O primeiro cenário possui seis comprimentos de onda disponíveis em cada enlace. O segundo possui doze comprimentos de onda disponíveis em cada enlace. Essa variação permite demonstrar a robustez do algoritmo de RWA em diferentes casos. As quantidades de comprimento de onda foram definidas empiricamente, quando foram levadas em consideração as características da topologia.

As simulações mostram o comportamento dos algoritmos AntOBS e OBS Puro de acordo com a variação da carga da rede, onde, para cada cenário, será apresentado o gráfico da probabilidade de bloqueio das rajadas em função da carga da rede óptica.

A Figura 18 apresenta o gráfico do comportamento da probabilidade de bloqueio em função da carga da rede para o cenário 1 da Topologia 1. Como era de se esperar, com cargas baixas, a taxa de probabilidade de bloqueio é sensivelmente menor do que me cargas altas. Devido ao reduzido número de comprimentos de onda, com cargas altas a rede fica com uma acentuada probabilidade de bloqueio da rajada.

Pode-se perceber também que a probabilidade de bloqueio do algoritmo AntOBS tem uma pequena diferença em relação ao algoritmo OBS Puro. A probabilidade de bloqueio é menor principalmente a partir de 20 *Erlangs*.

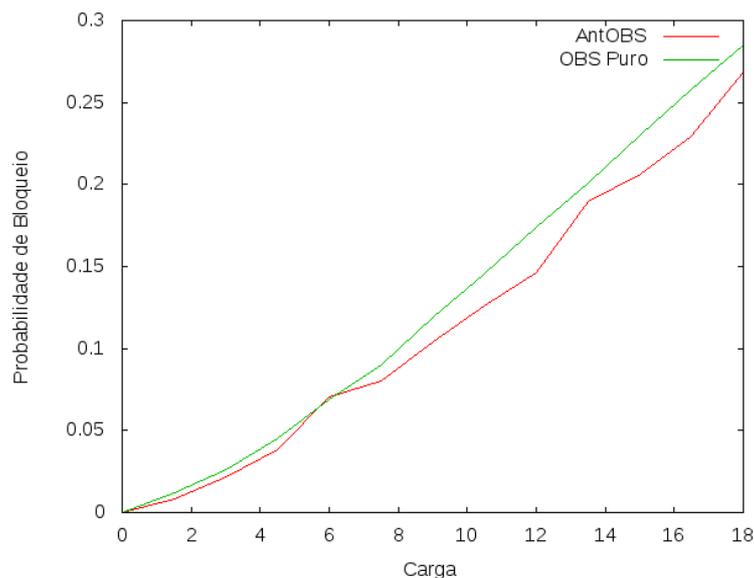


FIGURA 18: Gráfico da Probabilidade de Bloqueio x Carga no Cenário 1 da Topologia 1

A Figura 19 apresenta o gráfico da probabilidade de bloqueio em função da carga da rede óptica no cenário 2 utilizando a Topologia 1. O comportamento é semelhante ao gráfico da Figura 18, quando são usados seis comprimentos de onda. Porém, a probabilidade de bloqueio no gráfico abaixo é menor. Isso é justificado pelo maior número de comprimentos de onda disponíveis nos enlaces - no caso, doze comprimentos de onda. Como há mais comprimentos de onda disponíveis, a disputa é menor e logo a probabilidade de haver o bloqueio é menor. Isso fica evidente nos pontos inicial e final dos gráficos. Por exemplo, a maior probabilidade de bloqueio da Figura 19 é menor que 0.2 e o respectivo ponto da Figura 18 é quase 0,3.

Ainda na Figura 19, comparando os dois algoritmos, mais uma vez, o algoritmo AntOBS conseguiu probabilidades de bloqueio abaixo do que o algoritmo OBS Puro.

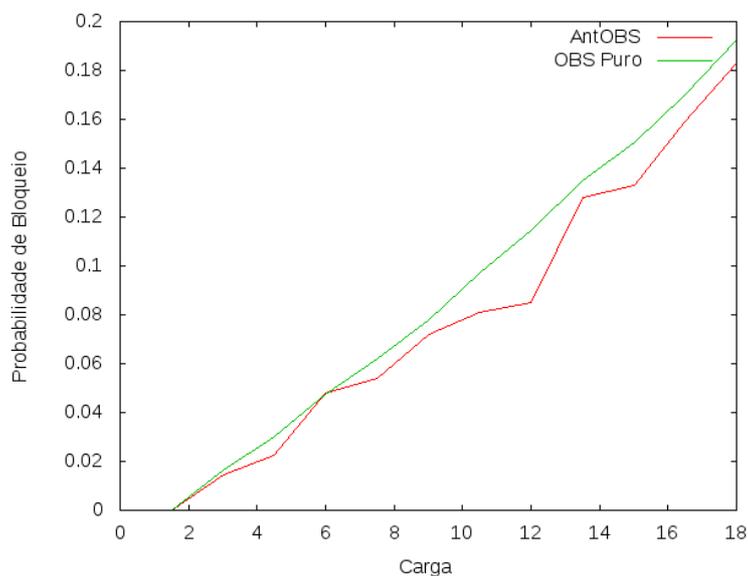


FIGURA 19: Gráfico da Probabilidade de Bloqueio x Carga no Cenário 2 da Topologia 1

6.2. Topologia 2 (NSFNET)

A partir dos primeiros testes com a Topologia 1, o próximo passo da simulação é aplicar os algoritmos AntOBS e OBS Puro em uma rede óptica maior. No caso dessa dissertação, a topologia semelhante a da rede NSFNET foi a escolhida. Essa é ilustrada na Figura 17.

Novamente, dois cenários foram definidos. Tais cenários são diferenciados pela quantidade de comprimentos de onda disponíveis nos enlaces. O primeiro cenário possui oito comprimentos de onda disponíveis e o segundo cenário possui doze comprimentos de onda disponíveis.

A Figura 20 abaixo apresenta a comparação entre os algoritmos AntOBS e OBS Puro, ilustrando o gráfico do comportamento da probabilidade de bloqueio em função da carga da rede para o cenário 1 da Topologia 2, ou seja, com oito comprimentos de onda disponíveis.

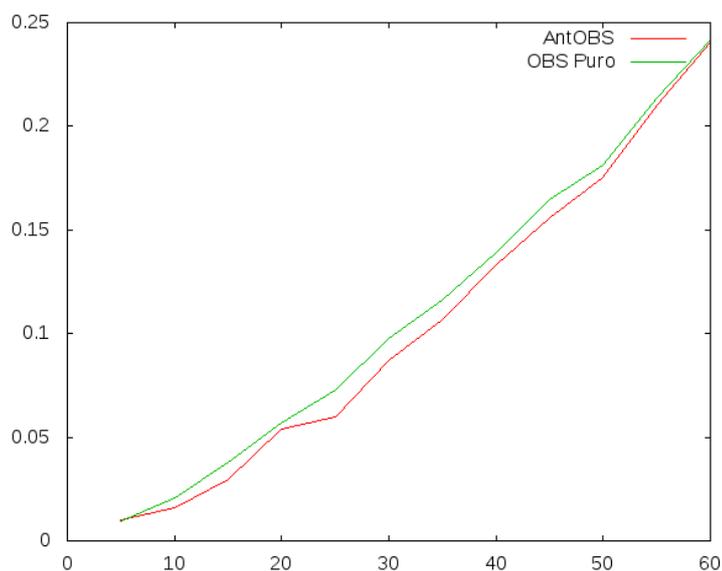


FIGURA 20: Gráfico da Probabilidade de Bloqueio x Carga no Cenário 1 da Topologia 2

O comportamento do gráfico é semelhante aos mostrados anteriormente nesse capítulo. E, novamente, o algoritmo AntOBS teve uma pequena diferença em relação ao OBS Puro. O AntOBS conseguiu uma probabilidade de bloqueio menor em todos os pontos do gráfico, onde pode-se destacar a partir da carga 20 Erlangs.

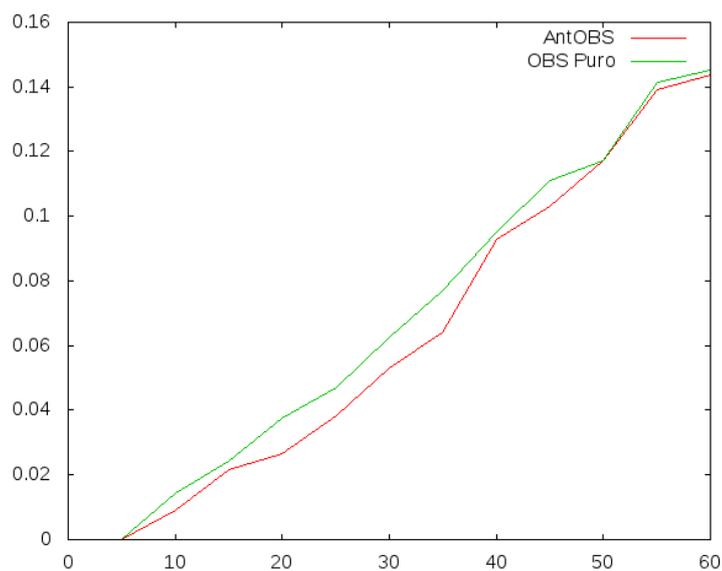


FIGURA 21: Gráfico da Probabilidade de Bloqueio x Carga no Cenário 2 da Topologia 2

A Figura 21 acima compara os algoritmos AntOBS e OBS Puro, ilustrando o gráfico do comportamento da probabilidade de bloqueio em função da carga da rede para o cenário 2 da Topologia 2, com doze comprimentos de onda disponíveis.

Da mesma forma que aconteceu na Topologia 1, o cenário 2 da Topologia 2 apresentou uma menor probabilidade de bloqueio. Isso é consequência do maior número de comprimentos de onda disponíveis do cenário 2. Nesse cenário, o AntOBS obteve uma probabilidade de bloqueio menor que o OBS Puro novamente.

Apesar de possuir desempenhos semelhantes, podemos ver que em todos os cenários o AntOBS teve um desempenho melhor, podendo comprovar a eficiência do AntOBS em relação ao OBS Puro. Outro ponto importante é o *overhead* gerado pela utilização das formigas no AntOBS. A próxima seção detalha esse assunto.

6.3. Overhead

Normalmente, analisa-se o *overhead* gerado pela utilização de pacotes de controle com relação aos dados da rede. Porém, como foi explicado no capítulo 5, no AntOBS, as formigas utilizam o canal de controle para se deslocar na rede. Portanto, o impacto gerado pelas formigas deve ser analisado juntamente com o BCP.

Ao término de cada simulação do AntOBS foi gerado o resultado com seis colunas da seguinte forma: número de BCPs enviados; número de BCPs recebidos; número de BCPs perdidos; número de rajadas enviadas; número de rajadas recebidas; número de rajadas perdidas.

Levando em consideração apenas as simulações feitas para gerar os gráficos dessa dissertação, foram feitas 600 simulações. Em nenhuma das simulações citadas, houve o BCP perdido. Tal situação acontece quando o algoritmo tenta reservar o comprimento de onda para o BCP e não existe um disponível. Consequentemente, esse fato leva ao bloqueio da rajada também.

Como não houve perdas de BCP, podemos concluir que o impacto gerado pela utilização das formigas foi irrelevante para os resultados mostrados, apesar de gerar mais pacotes com o acréscimo das formigas.

Capítulo 7

Conclusão e Trabalhos Futuros

O desenvolvimento de novos algoritmos RWA é uma etapa essencial para a evolução das redes ópticas, objetivando diminuir a perda de dados e, conseqüentemente, o desperdício de recursos com as retransmissões. Esse é um dos motivos pelos quais a comunidade científica internacional investe muita atenção nesta área. Uma estratégia de roteamento e atribuição de comprimento de onda bem projetada e implementada pode evitar que recursos importantes da rede óptica sejam desperdiçados.

Essa dissertação abordou esse importante tópico das redes ópticas, através do AntOBS, que é baseado no comportamento adaptativo de uma colônia de formigas. Tal comportamento mostrou-se capaz de satisfazer as necessidades de aplicações sensíveis ao tempo de forma rápida e eficiente.

7.1 Conclusão

A primeira etapa desse trabalho foi definir exatamente o problema. Através de intensas pesquisas, foi definido que o ideal seria trabalhar para a criação de um novo algoritmo de RWA dinâmico e adaptativo. Posteriormente, foi necessário escolher uma meta-heurística para tratar o problema em questão. No caso, as pesquisas feitas apontaram que ACO seria uma boa escolha. A partir disto, novas pesquisas foram feitas para fortificar a base teórica para a solução do problema. Em paralelo, foi feita uma análise dos algoritmos RWA já existentes.

Com a base teórica suficiente, iniciou-se a fase de implementação. O simulador NS-2 mostrou-se o mais adequado para o problema. Foi desenvolvido o algoritmo AntOBS em C++, a linguagem base para os protocolos do NS-2. A partir desta implementação, simulações comparando o AntOBS com o OBS Puro foram geradas.

Os resultados apresentados nos gráficos do capítulo 6 mostram que o trabalho proposto nesta dissertação teve sucesso. Os pontos de destaque são mostrados a seguir:

- O AntOBS teve menor probabilidade de bloqueio;
- O AntOBS é um algoritmo para tratar o RWA adaptativo, ou seja, altera as tabelas de roteamento de acordo com o estado atual da rede;
- O *overhead* gerado pela utilização das formigas não afetou o desempenho da rede;
- O AntOBS mostrou-se mais eficiente que OBS Puro.

7.2 Trabalhos Futuros

Alguns temas podem ser tratados para continuar essa dissertação. Alguns deles estão listados abaixo:

- O AntOBS pode ser aplicado a outras topologias e outros cenários;
- Novas fórmulas de atualização e parâmetros do algoritmo AntOBS podem ser testados;
- Pode-se inserir novos tipos de pacotes, por exemplo, outro tipo de formiga, para fazer novas tarefas na rede óptica, como por exemplo, checar o estado de um enlace periodicamente;
- Novas métricas podem ser testadas;
- Pode-se alterar o comportamento das formigas;
- Pode-se testar novos protocolos de sinalização, como por exemplo, o JIT;
- Pode-se testar novas penalidades aos custos dos enlaces;
- Pode-se aplicar outro algoritmo bio-inspirado no lugar de ACO, como por exemplo, Otimização por Colônia de Abelhas (*Bee Colony Optimization - BCO*).

Referências Bibliográficas

Bastos Filho, C. J. A.; Alcântara, R. S. e Oliveira, A. L. I. **A Novel Approach for a Routing Algorithm Based on a Discrete Time Hopfield Neural Network**. In: The First IEEE Symposium on Foundations of Computational Intelligence (FOCI'07), Honolulu. pp. 363-369, 2007.

Bastos Filho, C. J. A.; *et al.* **Routing Algorithm Based on Swarm Intelligence and Hopfield Neural Network Applied to Communication Networks**. Electronics Letters, v. 44, p. 995, 2008.

Bisbal, D. **Dynamic Routing and Wavelength Assignment in Optical Networks by Means of Genetic Algorithms**. In: Photonic Network Communications, vol. 7, nº 1, pp. 43-58, 2004.

Blum, C.; Li, X. **Swarm Intelligence in Optimization**. In: Blum, C.; Merkle, D.(Ords.). Swarm Intelligence: Introduction and Applications. German: Springer. p. 43-86, 2008.

Bonabeau, E.; Dorigo, M. e Theraulaz, G. **Swarm Intelligence: From Natural to Artificial Systems**. England: Oxford Univerity Press, 1999, 320p.

Carvalho, D. F. Roteamento em Redes Ópticas Transparentes Utilizando Otimização por Colônia de Formigas. Dissertação de Mestrado. Universidade de Pernambuco, 2008.

Chin, T. S.; Abbou, F.M. e Tat, E. H. **Ant-Based Contention-Resolution Schemes for Shared Fiber-Delay Line Optical Packet Switches**. 2006.

Corne D; Dorigo M e Glover F. **The Ant colony Optimization Meta-Heuristic. New Ideas in Optimization**. UK: London, McGraw-Hill, 11-32, 1999.

(Redes de Alta Velocidade) Distribuição de Vídeo em Redes de Alta Velocidade. Disponível em: <http://www.gta.ufrj.br/taquara/atividades/1.htm>, acesso em 30 junho de 2010.

Deneubourg, J.-L.; Aron, S. e Goss, S. **The self-organizing exploratory pattern of the argentine ant**. Journal of Insect Behavior, [S.l.], v.3, p.159-169, 1990.

Dorigo, M.; Birattari, M. e Stutzle, T. **Ant Colony Optimization: Artificial Ants as a Computational Intelligence Technique**. IEEE Computational Intelligence Magazine. Belgium, Université Libre de Bruxelles, nov. 2006.

Dorigo, M.; Caro, G. D. e Gambardella, L. M. **Ant Algorithms for Discrete Optimization**. Artificial Life, [S.l.], v.5, n.2, p.137–172, 1999.

Dorigo, M. e Gambardella, L. M. **Ant Colony System: A cooperative learning approach to the traveling salesman problem**. IEEE Trans. on Evolutionary Computation, [S.l.], v.1, n.1, p.53–66, 1997.

Dorigo, M.; Stutzle, T. **Ant Colony Optimization**. USA: The MIT Press, 2004. 321p.

Dorigo, M. e Caro, G. D. **AntNet: distributed stigmergetic control for communications networks**. Journal of Artificial Intelligence Research. 1998.

Garlick, R.M. e Barr, R.S. **Dynamic wavelength routing in WDM networks via Ant Colony Optimization**. Proceedings of ANTS '2002, pp. 250–255. Springer-Verlag Publishing, Brussels, Belgium, 2002.

Goss, S.; Aron, S.; Deneubour, J.; Pasteels, J. **Self-organized shortcuts in the Argentine ant**. Naturwissenschaften, [S.l.], v.76, n.12, p.579–581, Dezembro 1989.

Hsu C., Liu, T. e Huang, N., **Performance Analysis of Deflection Routing in Optical Burst-Switched Networks**. Proceedings, INFOCOM 2002, IEEE, 2002.

Huo, W.; Assi, C. e Shami, A. **A New Framework for Rapid Restoration in Optical Mesh Networks**. International Conference on Parallel Processing Workshops. Canadá: Montreal, 2004.

INFORMAÇÕES ESTATÍSTICAS [da] Internet. Disponível em: <<http://www.internetworldstats.com/stats.htm>>. Acesso em: 28 jul., 2010.

Mohammed, A. W.; Sahoo, N. C. e Tan, K. G. **Solving Shortest Path Problem Using Particle Swarm Optimization**. In: Applied Soft Computing, jan. 2008.

Mukherjee, B. **Optical WDM networks**. Davis, CA: Springer, 2006.

Ngo, S. H.; Jiang, X.; Horiguchi, S. **Adaptive Routing and Wavelength Assignment Using Ant-Based Algorithm**. In: Proc. 12th IEEE International Conference on Networks, vol. 2, pp. 482–486, nov. 2004.

Ngo, S. H.; Jiang, X. e Horiguchi S. **An ant-based approach for dynamic RWA in optical WDM networks**. Photonic Network Communications, vol. 11, no. 1, pp. 39-48, Jan. 2006.

Oliveira, J.; et al, D. **Integração de Redes IP com Rede Ópticas**. Recife, 2005.

Pavani, G. S.; *et al.* **Distributed Approaches for Impairment-Aware Routing and Wavelength Assignment Algorithms in GMPLS Networks.** In: Computer Networks, vol. 52, n° 10, pp. 1905–1915, jul. 2008.

Pavani, G. S. e Waldam, H. **Traffic engineering and restoration in optical packet switching networks by means of ant colony optimization.** 3rd BROADNETS, pp. 1068-1077, Oct. 2006.

Qiao C. e Yoo M. **Optical burst switching (OBS)—a new paradigm for an optical Internet.** Journal of High Speed Networks, 1999.

Schoonderwoerd, R.; Holland, O. e Bruten, J. **Ant-Like Agents for Load Balancing in Telecommunications Networks.** In: Proc. 1st International Conference on Autonomous Agents, pp. 209–216, California. Nova Iorque: ACM Press, 1997.

Shi, Z.; TinJin, Y. e Bing, Z. **Ant algorithm in OBS RWA.** Proc. of SPIE, Optical Transmission, Switching and Subsystems II, vol. 5625, pp. 705-713, Fev. 2005.

Silveira et al. **Arquitetura, Topologia e Roteamento em Redes Ópticas.** XXI Simpósio Brasileiro de Redes: 279-318, 2004.

Sivarajan, K. N. e Ramaswami, R. **Optical Networks: a practical perspective.** 2 Ed. San Francisco, CA: Morgan Kaufmann Publishers, 2002.

Sousa, D. F.; Esteves, R. P.; Farias, F. N. N. e Abelém, A. J. G. **Cálculo do Tempo de Ajuste em Redes Ópticas Comutadas por Rajadas Baseadas na Sinalização JET,** XXVIII Simpósio Brasileiro de Computação: Belém, 2008.

Strand, J.; Chiu, A. L. e Tkach, R. **Issue for routing in the optical layer.** Communications Magazine, v. 39, n. 2, p. 81-87, fevereiro 2001.

Tanenbaum, A. S. **Redes de Computadores,** 4ed. Ed. Campus, 2003.

Triay, J. e Cervelló-Pastor, C. **An Ant-Based Algorithm for Distributed Routing and Wavelength Assignment in Dynamic Optical Networks.** IEEE Journal on Selected Areas in Communications, vol. 28, No. 4, maio 2010.

Yang, L. e Rouskas, G. N. **Adaptive Path Selection in OBS Networks.** Journal of Lightwave Technology, vol. 24, No. 8, Agosto 2006.

Yoo, M. e Qiao C. **Just-enough-time (JET): A high speed protocol for bursty traffic in optical networks.** IN Proceeding of IEEE/LEOS Conf. on Technologies For a Global Information Infrastructure, 1997.

Wei, J. Y. e McFarland, R. I. **Just-In-Time signaling for WDM optical burst switching networks.** Journal of Lightwave Technology, 2000.

Zang, H.; Jue, J. P. e Mukherjee, B. **A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks.** Optical Networks Magazine, v. 1, n. 1, p. 47-60, Janeiro 2000

Zhang, Z.; Fu, J.; Guo, D. e Zhang, L. **Lightpath routing for intelligent optical networks.** IEEE Networks, v. 15, n. 4, p. 28-35, Julho 2001.