

UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO
EM CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO

Algoritmo de Roteamento para Economia de Energia em
Redes Ópticas

Rafael Motta Coutinho

Itajubá, Agosto de 2013.

Dissertação de Mestrado	Rafael Motta Coutinho	2013
-------------------------	-----------------------	------

UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO
EM CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO

Rafael Motta Coutinho

Algoritmo de Roteamento para Economia de Energia em
Redes Ópticas

Dissertação submetida ao Programa de Pós-Graduação em
Ciência e Tecnologia da Computação como parte dos
requisitos para obtenção do título de Mestre em Ciência e
Tecnologia da Computação.

Área de Concentração: Sistemas de Computação

Orientador: Prof. Dr. Danilo Henrique Spadoti

Agosto de 2013

Itajubá

Ficha catalográfica elaborada pela Biblioteca Mauá –
Bibliotecária Margareth Ribeiro- CRB_6/1700

C871a

Coutinho, Rafael Motta

Algoritmo de roteamento para economia de energia em redes
ópticas / Rafael Motta Coutinho. -- Itajubá, (MG) : [s.n.], 2013.
116 p. : il.

Orientador: Prof. Dr. Danilo Henrique Spadoti.

Dissertação (Mestrado) – Universidade Federal de Itajubá.

1. Economia de energia. 2. Redes ópticas. 3. PA-RWA. 4.
RWA. I. Spadoti, Danilo Henrique, orient. II. Universidade Fede_
ral de Itajubá. III. Título.

UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO
EM CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO

Rafael Motta Coutinho

Algoritmo de Roteamento para Economia de Energia em
Redes Ópticas

Dissertação aprovada por banca examinadora em 09 de agosto de 2013, conferindo ao autor o título de **Mestre em Ciência e Tecnologia da Computação**

Banca examinadora:

Prof. Dr. Danilo Henrique Spadoti (Orientador)

Prof. Dr. Valtemir Emerencio do Nascimento

Prof. Dr. Francisco Martins Portelinha

Prof^a. Dr^a. Thatyana de Faria Piola Seraphim

Itajubá

2013

Dedicatória

*À meus pais,
Luiz Carlos e Rosemara.*

*À meus irmãos,
Filipe e Flávio.*

*À minha avó,
Celina.*

*À minha namorada,
Josiani.*

Agradecimentos

Aos meus pais, Luiz Carlos e Rosemara, que iluminaram os caminhos obscuros com afeto e dedicação para que eu os trilhasse sem medo e cheio de esperança pela realização dos meus sonhos.

Ao meus irmãos, Filipe e Flávio, que sempre confiaram em mim.

A minha avó, por sempre estar presente e me dar todo apoio necessário para seguir em frente.

A minha namorada, Josiani, pelo exemplo de dedicação e paciência em todos os momentos.

Ao meu orientador, Prof. Danilo Henrique Spadoti, por me cobrar nos momentos em que precisava ser cobrado, por estar presente quando era preciso e, acima de tudo, por me ajudar em todos os momentos.

Resumo

O gasto de energia dos equipamentos de uma rede de telecomunicações representa uma parcela significativa de todo o consumo mundial de energia. Um aumento na demanda por energia, além de um alto custo, representa um maior volume de gases poluentes sendo lançados na atmosfera e contribuindo para o aquecimento global. Portanto, este é um problema de caráter político, econômico e social. Este trabalho propõe um novo algoritmo de roteamento e atribuição de comprimentos de onda em redes ópticas *backbone*, chamado Todo-Verde, que tem por objetivo minimizar o consumo de energia neste tipo de rede. Os resultados foram obtidos através de simulações e comparados a outros algoritmos de roteamento. As simulações foram realizadas com base na rede européia GÉANT2. Assim, os resultados puderam ser comparados a de trabalhos importantes já desenvolvidos na área. Os resultados mostram que com este novo algoritmo é possível alcançar uma economia média de 4% no consumo de energia sobre os algoritmos RWA tradicionais. Comparando-o aos algoritmos de roteamento e atribuição de comprimentos de onda tradicionais, o Todo-Verde teve um consumo aproximadamente três vezes menor que os outros.

Palavras chave: Economia de energia, Redes Ópticas, PA-RWA, RWA, WDM.

Abstract

The energy consumption of the network equipment represents a significant portion of the global consumption. Larger energy consumption, besides a high cost, represents a larger volume of greenhouse gases being released at the atmosphere, contributing for the global warming. Therefore, this is a political, economic and social aspect problem. This work proposes a new routing and wavelength attribution algorithm in backbone optical networks, called All-Green, which aims to minimize the energy consumption in this type of network. The results were obtained through simulations and were compared to other routing algorithms with different goals, including energy saving, just like All-Green. The simulations were performed based on the European network GEANT2. Thus, the results could be compared to similar work developed in the same area. The results show that with this new algorithm it is possible to achieve an average of 4% energy saving over the traditional RWA algorithms. Comparing it to the traditional routing and wavelength attribution algorithms, the Todo-Verde had approximately 3 times smaller energy consumption than the others.

Keywords: Energy-Aware, Optical Networks, PA-RWA, RWA, WDM.

Sumário

Dedicatória	v
Agradecimentos.....	vi
Resumo.....	vii
Abstract	viii
Sumário	ix
Lista de Figuras	xi
Lista de Tabelas.....	xiii
Glossário	xiv
1 Introdução	15
1.1 Motivação.....	15
1.2 Objetivos	18
1.3 Estrutura do trabalho	19
2 Revisão Bibliográfica	20
3 Desenvolvimento Teórico.....	26
3.1 Redes de Telecomunicações	26
3.1.1 Classificação de redes por escala	26
3.1.2 Topologias de Rede.....	26
3.1.3 Redes Backbone	27
3.1.4 Grafos.....	29
3.2 Redes Ópticas.....	31
3.2.1 História das Redes Ópticas.....	31
3.2.2 Características das Redes Ópticas	33

3.2.3	Topologia de Redes Ópticas.....	36
3.2.4	Arquitetura das redes ópticas	37
3.2.5	Técnicas de Multiplexação.....	39
3.2.6	Tecnologia WDM.....	41
3.2.7	Roteamento e atribuição de comprimentos de onda	44
4	Algoritmo.....	48
4.1	Ambiente de desenvolvimento.....	48
4.2	Redes Utilizadas.....	50
4.3	O modelo de rede	55
4.4	O algoritmo Todo Verde	56
4.5	Peso dos enlaces	58
4.6	Cálculo do consumo de energia	59
4.7	Fluxograma	63
5	Resultados.....	67
5.1	Parâmetros utilizados	67
5.2	Simulações	72
6	Conclusão	77
6.1	Resultados obtidos	77
6.2	Trabalhos futuros	78
	Trabalhos aceitos e submetidos.....	79
	Referências Bibliográficas	80
	Anexo 1 – Principais classes “ <i>Model</i> ”	85
	Anexo 2 – Principais classes “ <i>Control</i> ”	95
	Anexo 3 – Principais Classes “ <i>View</i> ”.....	111

Lista de Figuras

Figura 3-1 Topologias de Rede (Dante, 2005)	27
Figura 3-2 Rede Backbone Genérica (Texas A&M University, 1996)	28
Figura 3-3 (a) Grafo não direcionado, (b) Grafo direcionado (Easley & Kleinberg, 2010).....	29
Figura 3-4 Arpanet, exemplo de grafos em redes (Heart, McKenzie, McQuillian, & Walden, 1978)	30
Figura 3-5 Arpanet, grafo com abstrações (Heart, McKenzie, McQuillian, & Walden, 1978) ..	30
Figura 3-6 Evolução dos sistemas de comunicação (Dante, 2005)	32
Figura 3-7 Estrutura interna da fibra óptica (Kasper, Wiggenhorn, Resch, & Friess, 2012)	34
Figura 3-8 Visão geral da arquitetura de redes ópticas (Gençata, Singhal, & Mukherjee, 2003)	36
Figura 3-9 Arquitetura de uma rede óptica, com OLTs, OADMs e OXCs (Sasaki, Ramaswami, & Sivarajan, 2009).....	37
Figura 3-10 OXC de tamanho $NM \times NM$ (N = número de comprimentos de onda, M = número de fibras de entrada/saída (Gençata, Singhal, & Mukherjee, 2003)	39
Figura 3-11 Técnicas de Multiplexação. (a) Multiplexação por divisão de tempo (b) Multiplexação	40
Figura 3-12 Representação esquemática de um enlace WDM ponto-a-ponto (Dorf, 2003)	41
Figura 3-13 Enlace ponto-a-ponto em fibra óptica e o uso de OADM (Raja, 2003)	42
Figura 3-14 WDM para três comprimentos de onda (Zhemin & Hamdi, 2003)	43
Figura 3-15 Estrutura hierárquica da arquitetura multicamadas (Zhemin & Hamdi, 2003)	44
Figura 3-16 Conexões em caminhos ópticos de rede WDM (Gençata, Singhal, & Mukherjee, 2003)	45
Figura 4-1 Instituições de ensino com acesso a rede Kyatera (Fapesp, 1998)	50
Figura 4-2 Cidades alcançadas pela rede Kyatera (Fapesp, 1998)	51
Figura 4-3 Infraestrutura da rede Geant2 (European Commission, Europe's national research, & Education networks, 2011)	52
Figura 4-4 Rede GIGA (Centro de Pesquisa e Desenvolvimento - CPqD, 2005).....	53
Figura 4-5 Rede GEANT2 em grafo	54
Figura 4-6 Rede Kyatera em grafo numerado (Fapesp, 1998)	55

Figura 4-7 Equação $\ln(x)$	58
Figura 4-8 Fluxograma do algoritmo Todo-Verde	64
Figura 5-1 Exemplo de saída de uma conexão	69
Figura 5-2 Parâmetros de saída dos resultados finais da simulação.....	71
Figura 5-3 Consumo de energia x Número de conexões na rede GÉANT2.....	73
Figura 5-4 Taxa de Bloqueio x Número de Conexões	75
Figura 5-5 Energia Economizada x Número de Conexões.....	76

Lista de Tabelas

Tabela 4-1 Consumo de energia (em Watts) de acordo com a largura de banda agregada (Ricciardi, et al., 2011)	60
Tabela 5-1 - Levantamento estatístico dos Resultados.....	72

Glossário

ICT:	Information and communication technology
RWA:	Routing and Wavelength Assignment
PA-RWA:	Power-Aware Routing and Wavelength Assignment
WDM:	Wavelength division multiplexing
FDM:	Frequency division multiplexing
TDM:	Time division multiplexing
VTD:	Virtual Topology Design
PTD:	Physical Topology Design
AON:	All-Optical Network
ILP:	Integer Linear Programs
GRiDA:	Green Distributed Algorithm for Backbone Networks
LAN:	Local Area Network
MAN:	Metropolitan Area Network
WAN:	Wide Area Network
EDFA:	Erbium-Doped Fiber Amplifiers
OADM:	Optical Add Drop Multiplexers
WADM:	Wavelength Add Drop Multiplexer
OXC:	Optical cross connect
LCP:	Least congested-path
VWP:	Virtual wavelength path
OLT:	Optical line terminal
SONET:	Synchronous Optical Network
SDH:	Synchronous Digital Hierarchy
CPqD:	Fundação Centro de Pesquisa e Desenvolvimento
WC	Wavelength Conversion
OA:	Optical Amplifier

1 Introdução

1.1 Motivação

Os serviços que necessitam de uma grande quantidade de largura de banda, como, por exemplo, voz sobre IP e vídeo-conferência, estão se tornando cada vez mais populares e, com isso, o tráfego da *Internet* tem crescido de 50 a 60% todo ano (University of Minnesota, 2009). Essa demanda requer um crescente aumento no número de dispositivos necessários para infraestrutura das redes, tais como: amplificadores, *switches* e roteadores, o que acaba gerando um consumo cada vez maior de energia.

Hoje em dia, o consumo de energia relacionado à infraestrutura de informação e comunicações (ICT), representa por volta de 8% de toda a energia consumida no mundo e se estima que essa porcentagem cresça para 14% até o ano de 2020. Já a energia gasta exclusivamente com equipamentos de rede corresponde a quase 2% do total de consumo de energia mundial (Pickavet, et al., 2008).

Segundo (Christensen, Gunaratne, Nordman, & George, 2004), a principal barreira para um crescimento ainda mais acelerado da *Internet* é o alto consumo de energia relacionado aos equipamentos de rede necessários para o seu funcionamento.

O aumento nos gastos com ICT, junto com a escassez de combustíveis fósseis, chama atenção para diferentes aspectos no mundo moderno. O primeiro fator é econômico, pois os gastos com energia tem se tornado cada vez maiores. Além disso, há o fator ambiental, que está cada vez mais em evidência nos dias atuais. Considerando que boa parte da energia utilizada no mundo não é limpa (queima de combustíveis fósseis), esse consumo excessivo de energia aumenta a emissão de gases poluentes, que contribuem para uma maior poluição atmosférica e mudanças climáticas.

Essa questão do consumo de energia é tão relevante, que segundo (Baliga, Ayre, Hinton, Sorin, & Tucker, 2009), nos próximos anos a *Internet* não será restrita por sua capacidade de transporte, mas por seu consumo de energia e impacto ambiental.

A quantidade de energia gasta com infraestrutura de rede no mundo gira em torno de 22 GigaWatts (BONE project, 2009). Considerando o aumento anual de 50 a 60%, hoje temos algo em torno de 111 GigaWatts. Uma economia nesse valor representaria uma quantidade bem menor de gases poluentes sendo emitidos na atmosfera. Portanto, esse problema passa a ter caráter político, econômico e social.

As redes *backbone*, foco da otimização proposta neste trabalho, são responsáveis por aproximadamente 20% do consumo total de energia da *Internet* e a tendência é que essa porcentagem aumente significativamente em um futuro próximo (Bonetto, Chiaraviglio, Cuda, Castillo, & Neri, 2009).

As inovações tecnológicas imaginadas para economia de energia em redes são apresentadas neste trabalho, juntamente com os desafios e oportunidades para criar uma sociedade com crescimento mais sustentável. A infraestrutura de rede é um combinado heterogêneo de diversos equipamentos e para alcançar uma redução no consumo de energia faz-se necessária uma solução bem planejada e de fácil gerenciamento. A seguir serão listadas as principais estratégias que podem ser utilizadas para alcançar esse objetivo, segundo (Ricciardi, et al., 2011):

- **Incluir a eficiência energética como fator chave em avanços tecnológicos:** criação de novos dispositivos mais econômicos e que tenham o mesmo desempenho dos anteriores.
- **Design, construção e manutenção da rede devem considerar o fator energia como uma restrição adicional:** desenvolvimento de algoritmos que levem a economia de energia em consideração e satisfaçam os requisitos das redes em termos de cobertura, robustez e desempenho. De maneira geral, estratégias para um gerenciamento dinâmico de energia podem ser altamente econômicas.
- **Protocolos de controle orientados a economia de energia:** o objetivo é acomodar o tráfego da rede, considerando a eficiência no consumo de energia, consciência ambiental e o uso de fontes renováveis de energia.

Neste trabalho, o foco será a utilização da segunda estratégia citada, procurando estudar os diferentes algoritmos e aplicações existentes na literatura e, por fim, desenvolver um algoritmo que possa contribuir na caminhada para uma possível redução nos quase 8% do total mundial consumido pela infraestrutura de informação e comunicações.

Na literatura, este problema é conhecido como *Power-Aware Routing and Wavelength Assignment* ou, simplesmente, PA-RWA. Seu objetivo é mapear caminhos ópticos em redes com multiplexação por comprimento de onda, minimizando o consumo de energia (Wu, Chiaraviglio, Mellia, & Neri, 2009).

A maioria dos trabalhos na área consegue uma economia de energia através da desativação de elementos de rede. Segundo (Ricciardi, et al., 2011) esta não é uma solução viável na prática, somente em alguns experimentos em que há uma carga muito pequena, pois há vários inconvenientes em se utilizar o modo *Sleep* dos elementos de rede.

- **Tempo de espera:** um roteador leva alguns minutos para iniciar quando retorna do modo *sleep*, o que pode ser um grande inconveniente quando a rede estiver congestionada.
- **Consumo alto:** há um pico de consumo de energia quando o roteador está voltando do modo *sleep*. Dependendo do tempo em que ficar inativo, o consumo de energia pode ser ainda maior do que se estivesse ligado sem interrupções.
- **Financeiro:** não faz sentido ter um equipamento que normalmente custa muito caro, ocioso por questões de economia;
- **Vida útil:** Quando um roteador entra em vários ciclos de ligar e desligar, a sua vida útil se reduz consideravelmente.
- **Configurações manuais:** quando iniciado, o roteador pode exigir algumas configurações manuais para que comece operar adequadamente. Nesse caso, ficará inviável fazê-lo sempre.
- **Balanceamento de carga:** desligar alguns roteadores resulta em um balanceamento de carga reduzido. Dessa forma, o tráfego terá um número menor de caminhos como opção e a tendência é que a rede fique congestionada com muito mais facilidade.

Por conta desses motivos, neste trabalho o modo *sleep* será considerado como indisponível e a economia de energia será obtida exclusivamente de um roteamento otimizado das requisições de conexão que chegarem a rede.

1.2 Objetivos

A proposta inicial deste trabalho era fazer um estudo dos algoritmos de roteamento *Power-Aware RWA* presentes na literatura e a implementação dos mesmos em um ambiente único para que uma comparação pudesse ser feita com relação ao desempenho e economia de energia.

Conforme o estudo foi sendo desenvolvido e os algoritmos implementados, observou-se a necessidade de um novo algoritmo, que tivesse uma taxa de bloqueio reduzida e reduzisse o consumo de energia do sistema significativamente. Partindo desse princípio, o algoritmo Todo Verde foi implementado e testado nas mesmas condições dos algoritmos anteriores.

O algoritmo Todo-Verde foi dividido em duas fases. Na primeira, o objetivo é encontrar os melhores caminhos para que a rede não fique congestionado a medida que as conexões são estabelecidas. Na segunda fase, há um cálculo do consumo de energia por caminho e escolha do mais econômico.

Para obter resultados mais precisos e concretos, o sistema foi preparado para que os algoritmos pudessem ser testados em diferentes redes, bastando que o usuário entrasse com os dados das mesmas.

Durante a bateria de testes no sistema, duas redes foram utilizadas como exemplo: a rede Geant2 e a rede Kyatera. Ambas são redes criadas e utilizadas para fins acadêmicos. A segunda não pôde ser utilizada devido ao número de caminhos extremamente restrito, que não permite otimização por roteamento.

Os resultados foram mensurados em termos de taxa de bloqueio e consumo de energia. Eles foram comparados a dois algoritmos tradicionais da literatura, não voltados para economia de energia e O GreenSpark, que pode ser considerado o estado-da-arte em economia de energia para redes ópticas *backbone*.

Por fim, o objetivo geral do trabalho é buscar uma possível redução no consumo de energia em redes ópticas, sem que isso prejudique o seu desempenho, contribuindo, dessa forma, para um mundo mais sustentável.

1.3 Estrutura do trabalho

No Capítulo 2 é feita uma revisão bibliográfica, com o estado-da-arte nesta área e todo o material que serviu de base para o desenvolvimento do trabalho.

O Capítulo 3 é dividido em duas partes. A primeira aborda alguns conceitos gerais sobre redes de telecomunicações, como sua classificação por escala, as topologias, o que são redes *backbone* e um pouco de teoria dos grafos, que serão aplicadas no desenvolvimento do algoritmo aqui proposto. Na segunda parte é falado especificamente sobre redes ópticas, foco deste trabalho, trazendo uma linha do tempo que resume a sua história, suas principais características, arquitetura, técnicas de multiplexação, WDM, topologias, roteamento e atribuição de comprimentos de onda (RWA).

O Capítulo 4 explica em detalhes o algoritmo aqui proposto. Fala sobre o ambiente em que foi desenvolvido, as redes utilizadas como base para os testes, o modelo de rede, como são calculados o peso nos enlaces e o consumo de energia do sistema. Por fim, traz um fluxograma que resume o funcionamento do algoritmo, explicando passo a passo cada uma de suas etapas.

O Capítulo 5 mostra por que as simulações são a melhor forma de testar este tipo de algoritmo, quais os principais parâmetros utilizados no sistema tanto para entrada quanto para saída, e os resultados obtidos na comparação com algoritmo similares na literatura.

No Capítulo 6 é feita a conclusão, onde são apontados os principais resultados obtidos neste trabalho, mostrando como os objetivos iniciais foram alcançados e com uma sugestão para trabalhos futuros.

2 Revisão Bibliográfica

O tema tratado neste trabalho é relativamente novo e, embora as pesquisas sobre redes WDM (*Wavelength Division Multiplexing*) e o problema RWA (*Routing and Wavelength Assignment*) estejam bastante evidenciadas, ainda não há muitos trabalhos que tratam os algoritmos PA-RWA (*Power-Aware Routing and Wavelength Assignment*). Sendo assim, aqui será feita uma síntese do estado-da-arte nesta linha de pesquisa.

Como o tema do trabalho envolve redes de telecomunicações, fez-se necessário um livro que pudesse servir de base para toda a pesquisa. O livro escolhido foi (Tanenbaum, 2003) que faz uma boa introdução de todos os conceitos que envolvem as redes de computadores, mostrando exemplos do mundo real e explanando conceitos desde a camada física, de *hardware*, até as aplicações mais populares utilizadas nos dias atuais. Uma das desvantagens do livro é ser muito abrangente abordar alguns conceitos que não estão diretamente ligados a redes, tais como linguagem XML, compressão de vídeo, HTML, etc.

Após a leitura do livro de introdução a redes de computadores, foi necessário um maior aprofundamento em redes ópticas, para entender melhor como funciona a arquitetura dessas redes, gerenciamento de recursos, o que é roteamento e atribuição de comprimentos de onda, gerenciamento de conexões, tecnologias, etc. Nesse contexto, (Dorf, 2003) foi o livro escolhido, e se mostrou útil não só no início do trabalho, mas no decorrer de toda a pesquisa com explicações de conceitos e terminologias utilizadas por profissionais da área.

Ainda com relação a redes ópticas, (Neto, 2009) em sua tese de doutorado, propôs uma forma de reduzir a taxa de bloqueio nesse tipo de rede. Esta redução é feita através de um esquema de escolha de comprimentos de onda chamado canal específico e um esquema de monitoramento baseado na intensidade do tráfego de rede.

Outra tese de doutorado, que foi especialmente importante para dar início as pesquisas na área, foi (Dante, 2005), que implementou algoritmos para simulação de roteamento e atribuição de comprimentos de onda na linguagem C++ e obteve bons resultados com relação

a redução da taxa de bloqueio. A ideia neste trabalho é implementar esses algoritmos, de maneira que seja possível levar em consideração o aspecto economia de energia.

Carvalho em (Carvalho, 2002) apresenta uma definição mais completa dos tipos de redes ópticas existentes, como funcionam, quais as topologias utilizadas, o que são e como funcionam os encaminhadores de comprimentos de onda, exemplos de redes puramente ópticas e protocolos de acesso neste tipo de rede. Além disso, o trabalho apresenta uma relação muito bem elaborada de tecnologias de caminhos ópticos e quais os benefícios resultantes do uso desse tipo de tecnologia.

As fontes citadas anteriormente serviram como embasamento teórico para as pesquisas, porém, para prosseguir com o desenvolvimento foi necessário procurar artigos científicos com pesquisas mais específicas do tema tratado neste trabalho. Antes de estudar economia de energia, foi necessário analisar os algoritmos existentes para redes ópticas e, principalmente, aqueles que solucionam o problema de roteamento e atribuição de comprimentos de onda. Nos próximos parágrafos serão citados os principais artigos utilizados como norte para implementação dos algoritmos, que são o objetivo final do trabalho.

Conforme citado no início do Capítulo 1, o número de pessoas com acesso a *Internet* vem crescendo exponencialmente e novas aplicações se tornam cada vez mais comuns, tais como vídeo sob demanda, teleconferências, etc. Neste contexto, tem-se estimulado a pesquisa em redes de transporte para suportar o fluxo de informação gerado por essas aplicações. (Ramaswami R. , 2006) propõe um modelo de infraestrutura óptica inteligente com a utilização da tecnologia de multiplexação por divisão de comprimento de onda (WDM). Além disso, o autor faz uma síntese das tecnologias desenvolvidas nas últimas décadas para essa área e comenta porque algumas delas ficaram somente no papel, enquanto outras se tornaram úteis na prática.

Banerjee em (Banerjee & Mukherjee, 1996), propõe uma solução prática para o problema RWA, dividindo-o em vários subproblemas menores e solucionando cada um de maneira independente. Além disso, os autores mostram porque durante o projeto dos caminhos ópticos de uma rede WDM é necessário minimizar a quantidade de comprimentos de onda, portas, etc. Os principais pontos citados pelos autores são:

- Preservação dos recursos para demandas futuras;
- Imprevistos;
- Necessidade de reconfiguração em caso de falhas.

Murthy e Gurusamy em (Murthy & Gurusamy, 2002), fazem uma abordagem *top-down* do problema RWA sobre as redes WDM, onde utiliza o tráfego da rede para definir caminhos e comprimentos de onda como solução. O grande problema dessa abordagem é a limitação de recursos que o projetista de rede pode ter que seguir, como por exemplo, supor um número muito grande de comprimentos de onda disponíveis na rede óptica, o que torna o gerenciamento caro e complexo. Outro fator complicado nessa abordagem é um possível bloqueio em requisições futuras, devido ao número muito grande de comprimentos de onda alocados pelo algoritmo utilizado.

Em (Assis, Santos, & Giozza, Hybrid Algorithms for Routing and Assignment, 2010), também é pesquisada a solução do problema Min-RWA (minimização do número de comprimentos de onda). Nas soluções para o problema Min-RWA busca-se minimizar o número de comprimentos de onda para solucionar o problema de roteamento e alocação em redes WDM. A estratégia utilizada é contrária a anterior, procurando, em primeiro lugar, disponibilizar os caminhos físicos e comprimentos de onda, e, somente após isso, fazer o roteamento da demanda de tráfego. Os testes foram feitos utilizando duas meta-heurísticas (*Tabu Search* e *Simulated Annealing*) e os resultados ficaram bem próximos da solução ótima.

Ramaswami em (Ramaswami & Sivarajan, 1996), mostra a separação do projeto de uma rede toda óptica (AON – *All-Optical Network*) entre:

- **Projeto da Topologia Virtual (VTD – *Virtual Topology Design*):** onde são determinados os caminhos ópticos da rede baseada na demanda de tráfego.
- **Projeto da Topologia Física (PTD – *Physical Topology Design*) ou Roteamento e Alocação de Comprimentos de Onda (RWA– *Routing and Wavelength Assignment*):** roteamento na topologia física e alocação dos comprimentos de onda estabelecidos na fase anterior.

Em (Assis, Savasini, & Waldman, How Many Lightpaths we need Today and How Many Lightpaths we will need Tomorrow, 2009), foi criada uma solução para cálculo do número de comprimentos de onda atualmente e de como ele ficará no futuro. De maneira geral, o objetivo do artigo é maximizar o número de rotas remanescentes, fazendo uso eficiente de recursos, tais como: comprimentos de onda e *transceivers*. No trabalho é

proposto um algoritmo que economiza recursos de acordo com a demanda atual, para que os mesmos não faltem no futuro.

Diferentes estratégias e formulações matemáticas foram propostas como solução do problema de roteamento e atribuição de comprimentos de onda ao longo do tempo. Em (Pavon-Marino, Aparicio-Pardo, Garcia-Manrubia, & Skorin-Kapov, 2009), por exemplo, na pesquisa com o objetivo de reduzir o número de *transceivers* necessários para o projeto de rede, resolveram problemas de diferentes níveis utilizando programação linear inteira (ILPs - *Integer Linear Programs*) e matrizes de tráfego. Problemas de Programação Linear (ILPs) são problemas de otimização nos quais a função objetivo e as restrições são todas lineares. Além disso, todas as variáveis do problema devem pertencer ao conjunto dos números inteiros. As matrizes de tráfego contêm informações a respeito do fluxo de tráfego entre todos os possíveis pares de nós em uma rede. Segundo os autores, essa estratégia fornece o embasamento teórico e técnicas necessárias para reduzir a complexidade do problema e resolvê-lo com maior facilidade.

Alguns trabalhos abordam o mesmo tema utilizando programação linear inteira como solução para o problema (Jaumard, Meyer, & Thiongane, Comparison of ILP Formulations for the RWA, 2002). Nele os autores fazem uma síntese de várias soluções já existentes e propõe algumas melhorias nas formulações apresentadas, como eliminação de possíveis ciclos de caminhos ópticos e acréscimo do número de *hops* como uma restrição adicional na solução do problema RWA. Já em (Jaumard, Meyer, & Thiongane, ILP Formulations for the RWA Problem - Symmetric Systems, 2004) é feito um comparativo experimental das formulações ILP para vários exemplos de rede e tráfego.

Os artigos citados anteriormente foram estudados com o intuito de conhecer as últimas pesquisas sobre as redes ópticas e o problema de roteamento e atribuição de comprimentos de onda. Eles consideram grau virtual, quantidade de comprimentos de onda, taxa de bloqueio, etc. Porém, todos eles têm a desvantagem de não considerar o consumo de energia como uma possível restrição no sistema. Conforme argumentos apresentados no início deste Capítulo, essa é uma variável que tende a ficar a cada dia mais importante e, portanto, deve ser considerada.

Por ser um assunto que só ficou em evidência nos últimos anos, ainda não há muitas pesquisas específicas para tratar desse tema. Porém, nos próximos parágrafos serão destacados os principais trabalhos existentes até o momento para solucionar o problema de economia de energia em redes ópticas.

Ricciardi em (Ricciardi, et al., 2011), criou um modelo orientado a energia, combinando planejamento, protocolos e algoritmos voltados para este fim com o objetivo de solucionar o problema de roteamento e atribuição de comprimentos de onda. Outra ótima contribuição dos autores é analisar e quantificar através de formulações ILP a quantidade de energia que pode ser economizada nas próximas gerações de redes ópticas.

Para o desenvolvimento do algoritmo Todo Verde, foi utilizado como base o trabalho de (Pascoal & Martins, 2003), que resolve o clássico problema de selecionar *K-Shortest Paths*. Ou seja, para um determinado par de nós, ou elementos de rede, é preciso selecionar os K menores caminhos.

No artigo (Ricciardi, et al., 2012) é proposto um *framework* de roteamento e atribuição de comprimentos de onda chamado *GreenSpark*, que resolve os problemas tradicionais de gerenciamento de rede e acrescenta dois novos objetivos (em algoritmos separados): reduzir o consumo de energia e minimizar a emissão de gases poluentes na atmosfera. Boa parte da pesquisa realizada neste trabalho, se baseou no *framework GreenSpark*. As mudanças propostas diferem-se do *GreenSpark* em alguns aspectos:

- **Cálculo de caminhos:** ao invés de definir um número fixo de iterações, elas são feitas com base na capacidade da rede em questão.
- **Consumo:** por se basear em um número de iterações proporcional a capacidade da rede, o Todo Verde obteve um desempenho melhor que o *GreenSpark* no consumo de energia.
- **Objetivos:** Não tem como objetivo direto a redução de gases poluentes, somente a economia de energia.
- **Redes:** o Todo Verde foi testado em duas redes de pesquisa e preparado para ser utilizado em qualquer uma, bastando apenas um arquivo de entrada com dados básicos sobre a mesma.

Como referências para os testes nos algoritmos foram utilizadas duas redes. A primeira foi a Kyatera (Fapesp, 1998) e a segunda foi a Geant2 (European Commission, Europe's national research, & Education networks, 2011), ambas criadas para servir de base para pesquisas na área de redes, sendo a primeira brasileira e a segunda européia. As redes serão mostradas em detalhes nos próximos capítulos.

No projeto do algoritmo GRiDA, (Bianzino, Chiaraviglio, & Mellia, 2011) os autores propõe uma solução para reduzir o consumo de energia em redes *backbone*, através do desligamento dos enlaces em uma rede baseada em IP de um provedor de *Internet*. Embora

isso possa causar um aumento considerável na taxa de bloqueio da rede, o artigo apresenta uma técnica de aprendizagem dinâmica onde as decisões de desligamento dos enlaces são descentralizadas para cada nó, levando em conta decisões locais em intervalos aleatórios, sem uma coordenação entre os nós.

Em (Ricciardi, et al., 2011), os autores fazem uma formulação ILP para resolver dois problemas de uma só vez. O primeiro é reduzir o consumo de energia, problema chamado de *MinPower-RWA*, uma função cujo objetivo é obter o consumo mínimo de energia. O segundo é reduzir a emissão de gases poluentes na atmosfera, chamado de *MinGas-RWA*, uma função para obter o mínimo de emissão de gases poluentes. O sistema proposto possui duas fontes de energia, uma delas limpa (energia eólica, solar, etc) e a outra com energia suja. Embora os autores não citem uma solução que funcione bem nos dois casos, os resultados mostraram que cada uma delas obteve um ótimo desempenho no seu objetivo principal.

Outra solução para o roteamento e atribuição de comprimentos de onda *Power-Aware* (com economia de energia), também chamado de PA-RWA, é mostrada em (Coiro, Listanti, Valenti, & Matera, 2011). Nela o problema é solucionado através de ILP para o estabelecimento de caminhos ópticos em casos de roteamento estático e algumas heurísticas são propostas como solução para o estabelecimento de caminhos ópticos no caso de um sistema dinâmico. A solução é apresentada através de um algoritmo com custo baseado em carregamento, ou seja, o peso dos enlaces é atribuído conforme o carregamento na fibra óptica correspondente. Além disso, são propostos dois novos algoritmos, o primeiro deles utiliza a estratégia *First-Fit* e o segundo é associado a um custo de comprimento de onda em cada enlace no caminho, posteriormente o caminho é escolhido considerando aquele que possui o menor custo.

Com relação a comparação de resultados com outros trabalhos, em (Palmieri, Fiore, & Ricciardi, 2009) desenvolveram um *framework* chamado *SimulNet* para testar e validar protocolos de controle e algoritmos para redes ópticas. É um ambiente aberto para que pesquisadores possam testar e comparar os resultados de suas pesquisas. Seria uma excelente oportunidade para testar o algoritmo desenvolvido nesse trabalho e comparar com outras soluções, porém o *framework* não está disponível na *Internet* e nenhum dos autores respondeu ao e-mail de solicitação enviado. Sendo assim, o ambiente de testes deste trabalho teve que ser criado por conta própria e os algoritmos implementados um a um para fins de comparação de resultados.

3 Desenvolvimento Teórico

Neste Capítulo serão abordados os conceitos que serviram de base para o entendimento e desenvolvimento deste trabalho.

3.1 Redes de Telecomunicações

3.1.1 Classificação de redes por escala

Aqui as redes serão classificadas de acordo com seu tamanho. Embora já sejam conceitos bem difundidos na literatura, é importante trazer sua definição, pois elas serão citadas várias vezes mais adiante no trabalho.

As redes locais (LANs – *Local Area Networks*) correspondem a redes com alguns quilômetros de extensão, podendo ser um prédio ou campus universitário, por exemplo. Sua taxa de transmissão varia entre 10Mbps e 100Mbps , podendo chegar a até 10Gbps em redes mais modernas (Tanenbaum, 2003).

As redes metropolitanas (MANs – *Metropolitan Area Networks*) abrangem dezenas de quilômetros, geralmente cidades ou regiões metropolitanas, como uma rede de TV a cabo, por exemplo. Suas taxas de transmissão chegam a 40Gbps.

As redes geograficamente distribuídas (WANs – *Wide Area Networks*) abrangem uma grande área geográfica, normalmente milhares de quilômetros como um país ou continente. Sua taxa de transmissão varia de dezenas de Gbps a Tbps.

3.1.2 Topologias de Rede

Nos primeiros sistemas de comunicação, as informações eram transmitidas de uma estação diretamente a outra, que podia estar de alguns metros até alguns quilômetros. Esta foi a primeira topologia de rede, denominada ponto-a-ponto. Com o desenvolvimento dos

sistemas de comunicação, novas topologias foram surgindo ao longo do tempo. Na Figura 3-1 são mostradas as topologias de rede existentes nos dias atuais.

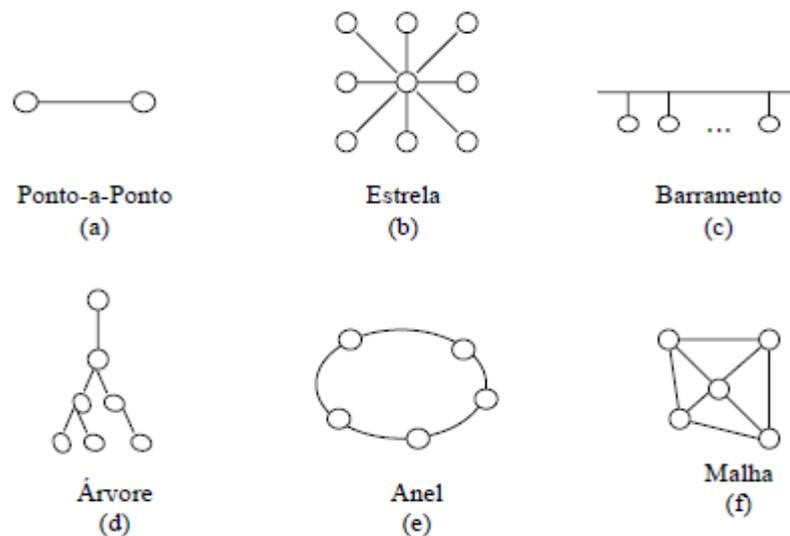


Figura 3-1 Topologias de Rede (Dante, 2005)

Das topologias de rede apresentadas, as topologias ponto-a-ponto, estrela, anel e malha são utilizadas também em redes ópticas. Aplicando as topologias anel e malha às redes ópticas, surgiram as seguintes arquiteturas: Redes ópticas de acesso (*Access Optical Networks* ou *Enterprise Optical Networks*), redes ópticas metropolitanas (*Metro Optical Networks*) e redes ópticas de longa distância (*Long-Haul Optical Networks* ou *Backbone Optical Networks*). Nas redes ópticas, a topologia tipo malha é um conjunto de anéis conectados por *cross connects* (OXC's).

3.1.3 Redes Backbone

Redes *backbone* são aquelas que conectam duas ou mais LANs, provendo um canal para que elas possam trocar pacotes (Texas A&M University, 1996). Após se conectar a uma rede *backbone*, a LAN pode manter-se distinta ou agrupada com outra.

Na Figura 3-2 é mostrada uma estrutura genérica de uma rede *backbone*, responsável por conectar a LAN A com a LAN B, através de um determinado dispositivo que pode variar entre um *hub*, repetidor, *bridge*, roteador ou *gateway*. Essa interconexão entre as redes pode ser feita, inclusive, utilizando padrões diferentes de ligação de dados no *backbone*.

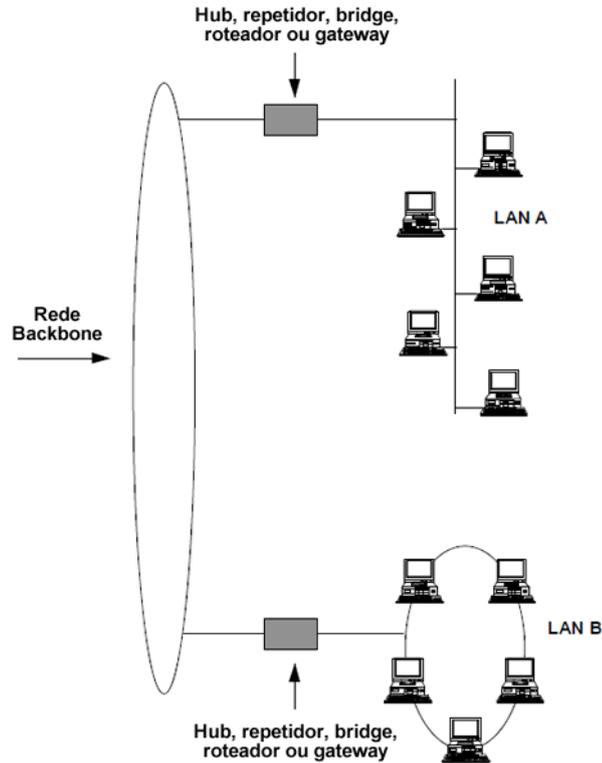


Figura 3-2 Rede Backbone Genérica (Texas A&M University, 1996)

Redes *backbone* podem ser utilizadas em uma variedade de ambientes distintos. Ela pode ser utilizada, por exemplo, para conectar duas ou mais LANs em prédios vizinhos ou até para prover acesso de LANs para WANs. Nelas pode ser utilizada uma grande variedade de configurações e tecnologias diferentes, de acordo com cinco critérios (Texas A&M University, 1996):

- Meio de transmissão.
- Topologia.
- Controle de acesso.
- Técnica de Transmissão.
- Velocidade.

Para não fugir ao escopo do trabalho, aqui não serão detalhados cada um desses critérios, pois o foco será o desenvolvimento de um algoritmo que opere sobre uma rede *backbone*.

3.1.4 Grafos

Aqui será mostrado um pouco da teoria dos grafos, aplicada às redes de telecomunicações para facilitar o entendimento dos próximos capítulos e dos algoritmos implementados, em que a teoria dos grafos está presente.

Um **grafo** é um modo de demonstrar visualmente uma coleção de objetos. Estes objetos são chamados de **nós** ou **vértices** e os nós são ligados por **arestas**. A Figura 3-3 mostra dois exemplos de grafo (Easley & Kleinberg, 2010). (a) é um grafo não direcionado, composto por um conjunto de 4 vértices: A, B, C e D, com A conectado a B por uma aresta, B conectado a C e assim por diante. Quando dois nós estão conectados por uma aresta, eles são chamados de nós vizinhos.

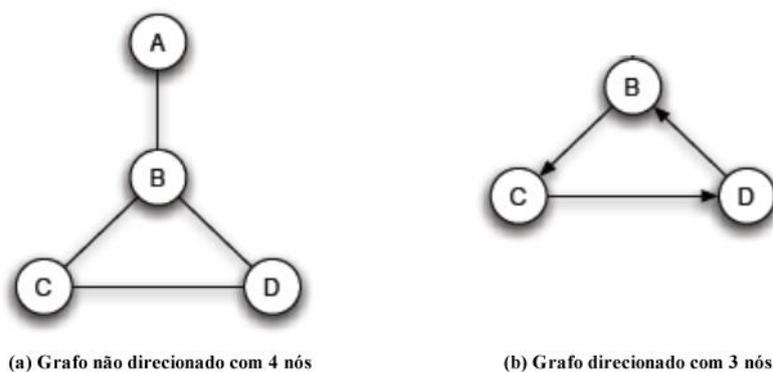


Figura 3-3 (a) Grafo não direcionado, (b) Grafo direcionado (Easley & Kleinberg, 2010)

As relações presentes entre os nós e arestas de (a) são denominadas simétricas, pois a aresta simplesmente conecta um nó a outro. Essa relação pode também ser assimétrica, com as arestas direcionadas para um lado ou outro, conforme o exemplo demonstrado no grafo (b). Nesse caso, B aponta para C, por exemplo, mas o contrário não é verdadeiro. Este tipo de grafo é conhecido como grafo direcionado, enquanto (a) pode ser chamado de grafo não direcionado.

A Figura 3-4 (Heart, McKenzie, McQuillian, & Walden, 1978) demonstra um exemplo real de uso da teoria dos grafos em redes. MIT, Harvard e Utah, por exemplo, são vértices do grafo ligado por arestas dentro do mapa dos Estados Unidos em Arpanet. Na prática, cada nó da figura representa um dos 13 *hosts* presentes na época, em Dezembro de 1970, ligados por arestas que representam os enlaces de comunicação entre eles. Os grafos

podem ser úteis em modelos de rede, como no exemplo citado, modelos matemáticos e várias outras aplicações.

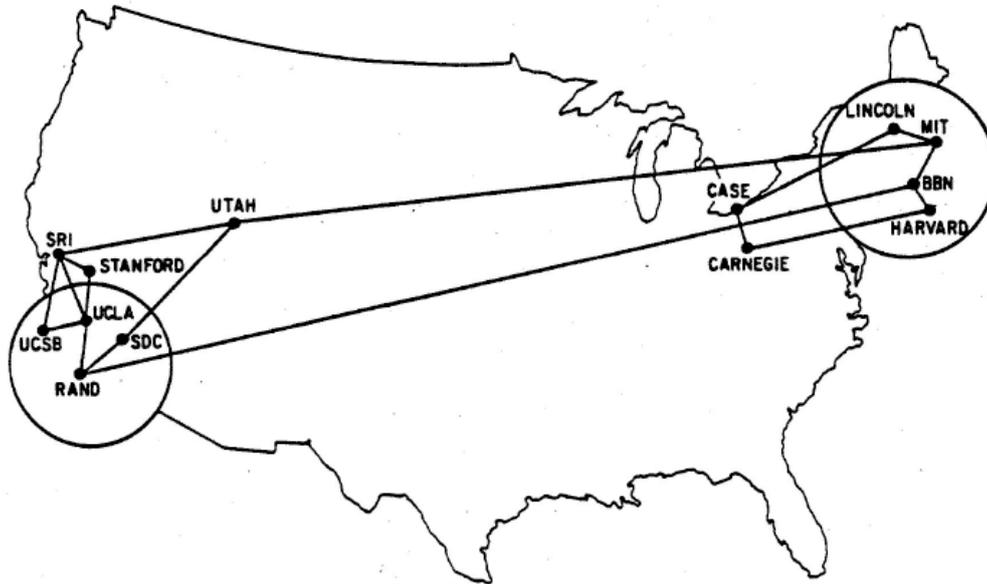


Figura 3-4 Arpanet, exemplo de grafos em redes (Heart, McKenzie, McQuillian, & Walden, 1978)

Abstraindo o mapa americano de fundo e os *hosts* destacados no mapa, chegamos a algo como a Figura 3-5, uma representação bem mais próxima do primeiro exemplo apresentado nesta seção.

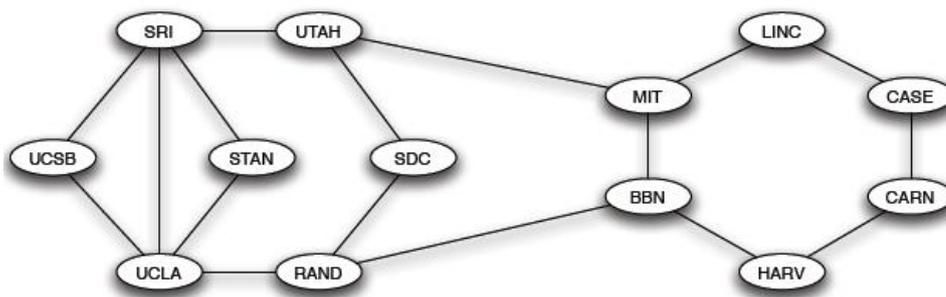


Figura 3-5 Arpanet, grafo com abstrações (Heart, McKenzie, McQuillian, & Walden, 1978)

Caminhos em um grafo são um passeio por uma sequência de vértices. Por exemplo, um viajante indo de cidade a cidade ou mesmo uma sequência de cliques em vários perfis de uma rede social. Caminhos podem também ser definidos como uma sequência de nós onde cada par de nós sequenciais esteja ligado por uma determinada aresta. Vale lembrar que os

caminhos não são apenas os nós, mas também as arestas responsáveis por fazer a ligação deles. Na teoria dos grafos, um nó pode ser repetido em um determinado caminho, porém, neste trabalho só serão considerados aqueles caminhos onde os nós não se repetem, denominados caminhos simples.

Tomando por base a Figura 3-5, um possível caminho simples seria HARV-BBN-MIT-UTAH. Um possível caminho com repetição de nós seria: LINC-MIT-UTAH-MIT-BBN.

3.2 Redes Ópticas

3.2.1 História das Redes Ópticas

Com a invenção do telefone, as técnicas de comunicação elétrica e analógica começaram a se desenvolver e, a partir delas, um sistema de telecomunicações também foi criado. Com o emprego de cabos coaxiais, já no século XX, a capacidade dos sistemas telefônicos aumentou para 300 canais de voz, porém as perdas de frequência desses cabos eram maiores do que 10MHz (Dante, 2005).

Dois parâmetros são utilizados para medir a capacidade de um sistema de comunicação: distância (L) e taxa de transmissão (B_{total}) em bits/s [Km] entre dois elementos quaisquer da rede (Wei, 2002). A Figura 3-6 mostra como foi a evolução dos sistemas de comunicação de 1850, desde a invenção do telégrafo, até o ano 2000, utilizando o parâmetro taxa de transmissão como comparativo.

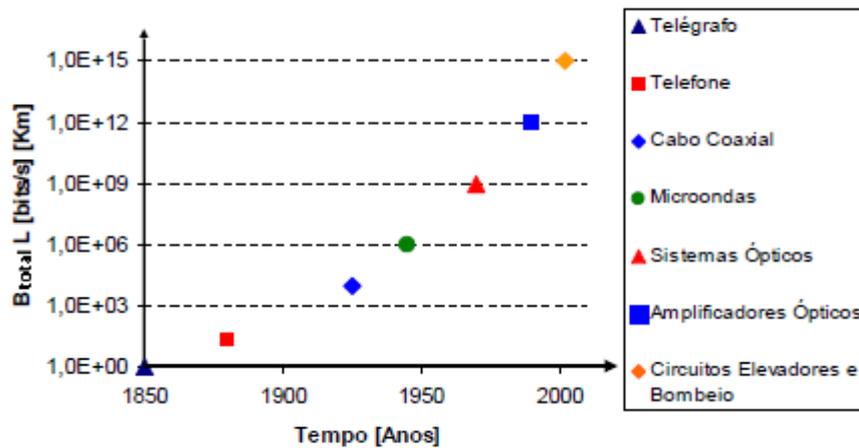


Figura 3-6 Evolução dos sistemas de comunicação (Dante, 2005)

O primeiro sistema óptico foi proposto em 1966, porém, a perda era tão grande (cerca de 1000dB/km) que inviabilizava a sua adoção. Em 1978, com a descoberta dos *lasers* e a fabricação de novos detectores, o primeiro sistema óptico foi ativado. Ele operava com taxas entre 50Mbps e 100Mbps a distâncias de até 10km. Este sistema foi chamado de primeira geração de sistemas ópticos e é demonstrado em (T. C. Cannon, 1978).

No final da década de 80, foi atingida uma distância de 50Km, com uma taxa de transmissão de 1,7Gbps e perdas na fibra da ordem de 0,5 dB/km, que ficou conhecida como segunda geração de sistemas ópticos comerciais. Já em 1990, entrou em operação a terceira geração de sistemas ópticos, com taxas de transmissão de até 2,4Gbps (Dante, 2005).

Também nesta época, houve o surgimento do Amplificador a Fibra Dopada com Érbio (EDFA) que teve grande impacto nas comunicações ópticas por acabar com a necessidade de converter os sinais ópticos para elétricos antes de amplificá-los. Dentre as vantagens trazidas pelo uso do EDFA em redes ópticas estão a maior cobertura das regiões de comunicação e a redução dos custos com a rede, visto que o custo com equipamentos necessários para seu bom funcionamento reduziu-se drasticamente.

Os avanços trazidos pelo EDFA, junto com o uso de novas técnicas de transmissão permitiu a criação de uma rede com 32 canais a uma taxa de transmissão de 5,2Gbps e com quase 10Km de extensão, apresentada em (N. Shimojoh, 1997).

Conforme já mencionado, a análise da capacidade de uma rede é o produto de dois parâmetros: distância e taxa de transmissão. Com essa relação em mente é possível citar 3 redes recentes em que houve um grande avanço tecnológico (Dante, 2005).

1. Sistema óptico terrestre com 200 canais de 10Gbps e distância de 400Km, com total de 800[Tbps][Km];

2. Outro sistema óptico terrestre com 200 canais de 40Gbps a uma distância de 50Km, total 400[Tbps][Km];
3. Sistema óptico submarino com 1 canal de 40Gbps a uma distância de 10000Km, total de 400[Tbps][Km].

O avanço tecnológico acontece nos 3 sistemas, cada um com suas características. Em 1, por exemplo, um grande número de canais é multiplexado a uma distância consideravelmente alta. Já no sistema 2, a distância não é tão alta, mas a taxa de transmissão por canal óptico quadruplicou. No sistema 3, só há um canal disponível, porém sua taxa de transmissão é alta e a distância atingida supera todos os outros exemplos, por ser um sistema submarino e precisar superar distâncias intercontinentais.

Portanto, os desafios tecnológicos em redes ópticas estão em aumentar a taxa de transmissão por canal óptico, multiplexar uma quantidade de canais ópticos cada vez maior, e transmitir os sinais a distâncias cada vez mais longas com o objetivo de reduzir custos com equipamentos e conseguir uma região de cobertura cada vez maior.

A seguir será apresentada uma linha do tempo, criada por (Júnior, 2012) e aprimorada neste trabalho, que resume boa parte do que foi falado nesta seção:

- 1860 - Invenção do telefone
- 1880 - Invenção do cabo coaxial
- 1960 - Invenção do *Laser*
- 1962 - *Laser* Semicondutor (77K)
- 1966 - Proposto o primeiro sistema óptico
- 1970 - *Laser* Semicondutor Tamb (300K)
- 1970 - Fibra Óptica (20 dB/km)
- 1973-75 - *Laser*, Fibras Ópticas, Sistemas (no Brasil)
- 1985 - Amplificador a Fibra Dopada com Érbio (EDFA)
- 1992-94 - Sistemas WDM de longa distância
- 1998 - Sistemas WDM metropolitanos
- 2000-04 - Redes Ópticas WDM e redes fotônicas

3.2.2 Características das Redes Ópticas

As fibras ópticas podem suportar o aumento exponencial do tráfego da *Internet* citado no início do Capítulo 1, devido a algumas características específicas, segundo (Dorf, 2003):

- Enorme largura de banda: próxima a 50 Terabits por segundo, para fibras monomodo (*single-mode fiber*). Potencial aproximadamente três vezes maior do que taxa de dados em meio eletrônico.
- Baixa atenuação do sinal: menor do que 0,2 dB/km
- Baixo consumo de energia
- Pouco uso de material
- Pouco espaço necessário
- Baixo custo

Normalmente a fibra é acessada por vários usuários ou sub-redes com uma taxa bem menor, o que pode acabar reduzindo consideravelmente essas taxas de transmissão. Para explorar esse imenso potencial das redes ópticas, são necessários meios para lidar com a concorrência de múltiplos usuários como, por exemplo, a tecnologia de multiplexação por comprimento de onda WDM explicada mais adiante.

As fibras ópticas são utilizadas para transportar sinais digitais na forma de pulsos de luz modulados. Ela é composta por um cilindro de vidro, envolvida por uma camada de vidro concêntrica de maior índice de refração para garantir o mecanismo de reflexão interna total. A Figura 3-7 mostra de maneira esquemática como é a estrutura interna de uma fibra óptica.

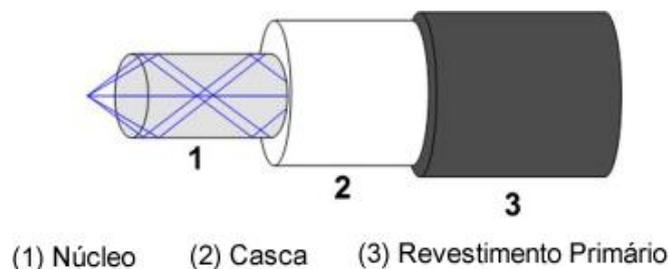


Figura 3-7 Estrutura interna da fibra óptica (Kasper, Wiggernhorn, Resch, & Friess, 2012)

Hoje em dia, as fibras ópticas são adotadas em praticamente todos os tipos de redes de telecomunicação, exceto (na maioria dos países) em redes de acesso residenciais, devido ao custo envolvido na instalação e ao questionável prazo de retorno do investimento dos provedores de serviço desse tipo.

Há duas gerações distintas de redes ópticas. A primeira foi usada para prover uma maior capacidade de transmissão, com uma baixa taxa de erro de *bits*. Todas as outras

funcionalidades da rede eram oferecidas por dispositivos eletrônicos. Os exemplos mais comuns de redes da primeira geração são a SONET (*Synchronous optical network*) e a SDH (*Synchronous digital hierarchy*), que formam o núcleo da infraestrutura de telecomunicações da América do Norte, Europa e Ásia (Sasaki, Ramaswami, & Sivarajan, 2009).

SONET é o padrão atual de transmissão e multiplexação para sinais de alta velocidade na infraestrutura da América do Norte. SDH é um padrão semelhante adotado na Europa e Japão. A estrutura síncrona desses padrões oferece uma redução significativa do custo de multiplexação e demultiplexação. Nelas um único sinal pode ser extraído localizando-se os respectivos *bits* no sinal multiplexado.

São vários os benefícios oferecidos pelos padrões SONET e SDH. Entre eles, pode-se citar: simplificam a multiplexação com relação à multiplexação assíncrona, seu gerenciamento é mais fácil, definem padrões de interfaces ópticas para interoperabilidade de equipamentos de diferentes fabricantes. Além disso, possuem um tratamento de falhas muito mais eficaz do que os sistemas anteriores, com topologias de redes específicas e protocolos que oferecem serviços mais eficientes no tratamento de falhas.

Com relação às redes da primeira geração, sua principal desvantagem era a necessidade de um processamento eletrônico não só no nó fonte e destino de uma determinada conexão, mas em todos os nós pelos quais os dados passavam. Por isso, pesquisas levaram à criação das redes de segunda geração, que eliminavam essa grande desvantagem.

Atualmente, na segunda geração das redes ópticas, o roteamento, comutação e a inteligência foram passadas para camada óptica. Esta evolução acontece, pois a taxa de transmissão vem se tornando cada vez maior, a ponto de os dispositivos eletrônicos não conseguirem processar dados no mesmo ritmo. Esse tipo de rede é também chamado de rede roteada por comprimento de onda e oferecem caminhos ópticos para os seus usuários. Caminhos ópticos são conexões entre dois nós que oferecem um comprimento de onda em cada enlace entre eles. Pode ou não haver conversão de comprimentos de onda entre um enlace e outro. O mesmo comprimento de onda pode ser utilizado em várias partes da rede, contanto que não seja no mesmo enlace (Sasaki, Ramaswami, & Sivarajan, 2009).

3.2.3 Topologia de Redes Ópticas

O enlace de transporte óptico ponto-a-ponto é a unidade básica para todos os tipos de redes ópticas (Feiste, et al., 2001). As topologias de redes ópticas e interconexões são compostas por vários enlaces ópticos ponto-a-ponto.

A topologia de rede depende da extensão dos enlaces, dos multiplexadores e do *hardware* em geral, tais como: *switches* e roteadores. Pode ser classificada como anel, árvore, malha e suas combinações.

A classificação da rede óptica quanto a sua arquitetura, varia de acordo com a área geográfica ocupada, sub-redes e interconexões entre elas. Conforme (Gençata, Singhal, & Mukherjee, 2003), podem ser classificadas como:

- Redes ópticas *Long-haul* ou *backbone*
- Redes ópticas *Regional/metro*
- Redes ópticas de acesso

A Figura 3-8 apresenta uma visão geral da arquitetura das redes ópticas, para um melhor entendimento.

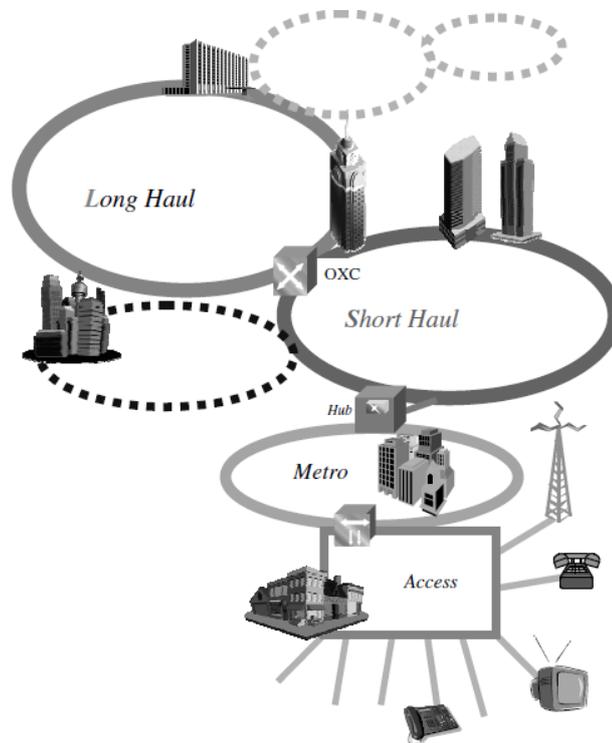


Figura 3-8 Visão geral da arquitetura de redes ópticas (Gençata, Singhal, & Mukherjee, 2003)

As redes ópticas de acesso usam uma enorme variedade de tecnologias e protocolos para fornecer conectividade dos clientes com a central. Normalmente possuem de 1 a 10km

de extensão e são chamadas de a “primeira milha” ou “a última milha” das redes de comunicação.

Redes ópticas *Long-haul* ou *backbone* têm escala global ou, ao menos, maior do que regional, com tamanho mínimo de 500km. Oferecem conectividade a várias redes metro ou sub redes.

Já as redes ópticas *Metro* ou *Regional* estão situadas entre as duas anteriores, com extensão entre 10km e 500km (Raja, 2003).

3.2.4 Arquitetura das redes ópticas

A Figura 3-9 apresentada originalmente em (Sasaki, Ramaswami, & Sivarajan, 2009), mostra a arquitetura de uma rede óptica WDM da segunda geração. Essa rede provê caminhos ópticos aos seus usuários, que geralmente são roteadores IP ou terminais SONET. Nela estão presentes 6 caminhos ópticos diferentes: A-F, B-C, D-E, E-F (1), E-F (2), E-F (3). Por exemplo, o caminho óptico A-F utiliza λ_2 (maior linha pontilhada da figura).

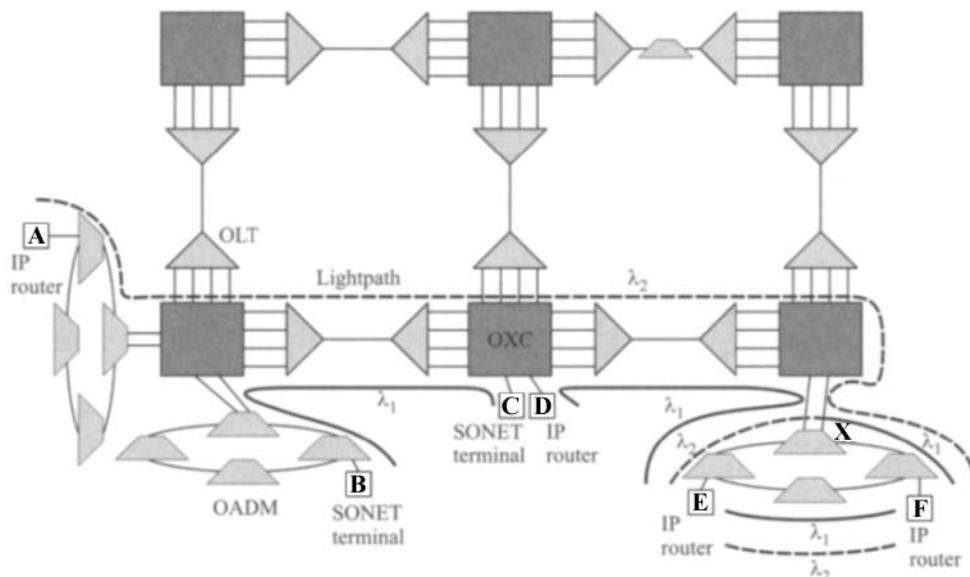


Figura 3-9 Arquitetura de uma rede óptica, com OLTs, OADMs e OXCs (Sasaki, Ramaswami, & Sivarajan, 2009)

B-C, D-E e E-F(1) não dividem nenhum enlace, portanto podem usar o mesmo comprimento de onda λ_1 , conforme explicado na seção 3.2.2. Como A-F e B-C dividem o mesmo enlace, devem utilizar comprimentos de onda diferentes. Quando há dois ou mais caminhos ópticos em um mesmo enlace, como ocorre no segmento E-F, eles devem usar

comprimentos de onda diferentes. No exemplo citado, todos os caminhos ópticos usam o mesmo comprimento de onda do nó fonte ao nó destino por não haver conversores de comprimento de onda na rede em questão. Se somente dois comprimentos de onda estivessem disponíveis e houvesse a necessidade de utilizar o terceiro caminho óptico E-F, isso não seria possível sem conversão de comprimento de onda. Porém, com um conversor de comprimento de onda X no enlace E-F é possível ter esse terceiro caminho utilizando λ_2 em E-X e λ_1 em X-F.

Os elementos chave das redes ópticas, mostrados na Figura 3-9, são os terminais de linha óptica (OLT – *Optical line terminal*), multiplexadores ópticos *add/drop* (OADM – *Optical add/drop multiplexers*) e o *Optical crossconnect* (OXC).

Um OLT multiplexa vários comprimentos de onda em uma única fibra e demultiplexa um conjunto de comprimentos de onda de uma fibra em várias outras fibras (Sasaki, Ramaswami, & Sivarajan, 2009). As OLTs são utilizadas como terminais de uma conexão ponto a ponto em um enlace WDM. O OADM tem por objetivo retirar ou adicionar informações em um comprimento de onda específico nos sinais que passam por ele.

Quando os nós são conectados por fibras de maneira a formar um grafo arbitrário é necessário um dispositivo para rotear os sinais de entrada para a saída desejada. Esse dispositivo é denominado OXC (*Optical crossconnect*) e, assim como os OADM, podem ou não fazer a conversão de comprimento de onda. A Figura 3-10 mostra o funcionamento de um OXC, onde os sinais de um determinado comprimento de onda (λ_1 , por exemplo), chegam em uma fibra M e saem por outra (Gençata, Singhal, & Mukherjee, 2003). O OXC tem uma função parecida com o OADM, mas em tamanhos maiores por ter de dezenas a milhares de portas, enquanto o OADM tem somente duas. Além disso, o OXC tem a capacidade de trocar comprimentos de onda de uma porta para outra.

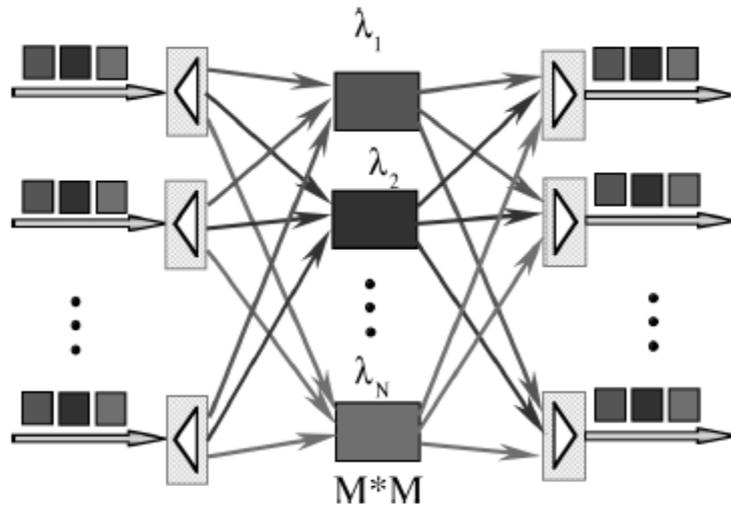


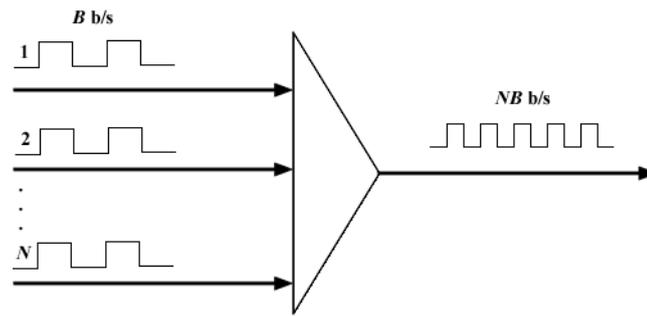
Figura 3-10 OXC de tamanho $N \times N$ (N = número de comprimentos de onda, M = número de fibras de entrada/saída (Gençata, Singhal, & Mukherjee, 2003)

3.2.5 Técnicas de Multiplexação

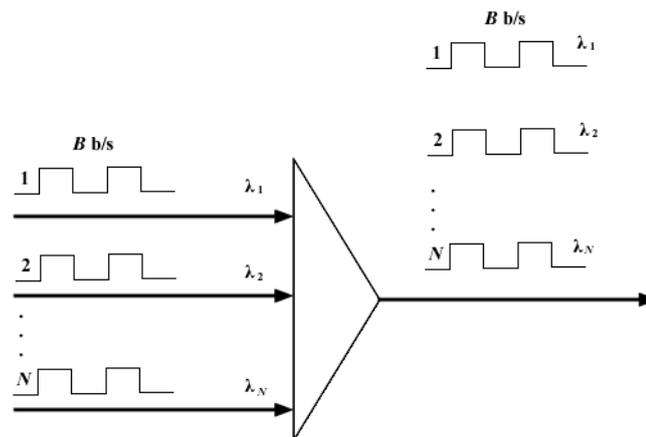
É muito mais econômico transmitir dados a taxas de *bits* altas em uma única fibra óptica, do que transmitir a uma taxa menor em várias fibras. Por isso há a necessidade da multiplexação dos sinais.

A Figura 3-11, mostra duas formas possíveis de fazer a multiplexação dos sinais. Ambas técnicas mostradas na figura utilizam N sinais de entrada, cada um com B [bps] e multiplexam em uma única fibra, com uma taxa de NB b/s.

A primeira técnica (a) é conhecida como Multiplexação por Divisão de Tempo (TDM – *Time Division Multiplexing*), e nela muitos sinais de baixa velocidade são intercalados pelo multiplexador para formar um único sinal de alta velocidade. Essa intercalação pode ser o primeiro *byte* do sinal 1, com o primeiro *byte* do sinal 2 e assim sucessivamente. Como um exemplo dessa técnica, um multiplexador usando a técnica TDM poderia multiplexar 64 sinais de 155Mb/s em um único sinal de 10Gb/s. Há algumas formas de fazer essa multiplexação com sinal óptico, sem conversão eletrônica e elas são chamadas de Multiplexação por divisão de tempo óptica (OTDM - *Optical time division multiplexing*) (Ellis, et al., 1995).



(a) Multiplexação TDM ou OTDM



(b) Multiplexação WDM

Figura 3-11 Técnicas de Multiplexação. (a) Multiplexação por divisão de tempo (b) Multiplexação por divisão de comprimentos de onda (Sasaki, Ramaswami, & Sivarajan, 2009)

O segundo modo de aumentar a capacidade da fibra, proposto em (b), é chamado de Multiplexação por divisão de comprimento de onda (WDM - *Wavelength division multiplexing*). WDM usa a mesma ideia da Multiplexação por divisão de frequência (FDM - *frequency division multiplexing*), que vem sendo utilizada em sistemas de rádio há mais de um século, no contexto das redes ópticas. Nela os sinais são transmitidos simultaneamente em comprimentos de onda diferentes na fibra óptica. Na prática, é como se houvessem várias fibras “virtuais” em uma mesma fibra, cada uma carregando uma informação diferente. Como esta técnica será utilizada neste trabalho, posteriormente ela será descrita de maneira mais detalhada.

As técnicas TDM e WDM são diferentes maneiras de aumentar a capacidade de transmissão, e, nas redes atuais, são utilizadas em conjunto para atingir um desempenho ainda melhor. Combinando essas técnicas é possível chegar até 1Tb/s em uma única fibra, o que permite a criação de redes com maior capacidade e que alcancem distâncias maiores.

3.2.6 Tecnologia WDM

WDM é uma tecnologia para multiplexação de comprimentos de onda em redes ópticas. Ela provê a concorrência necessária para explorar a alta largura de banda das redes ópticas, permitindo transmitir várias informações simultaneamente, em diferentes comprimentos de onda (Jean-Pierre, 1993).

Cada comprimento de onda opera com uma velocidade diferente de processamento eletrônico e, assim, permite um fluxo de informação bem maior, multiplicando a quantidade de canais disponíveis pela velocidade de cada uma deles. Por exemplo, se uma fibra suporta 160 canais de comunicação (comprimentos de onda), a 40Gbps cada um, iria resultar em uma capacidade total de:

$$160 \text{ canais} * 40\text{Gbps} = 6,4\text{Tbps}$$

Na Figura 3-12, é feita uma representação esquemática de um enlace WDM ponto-a-ponto, que permitem a interfaces ópticas de um único comprimento de onda (SONET), ou várias se comunicarem através dessa arquitetura aberta. Um conjunto desses enlaces compõe a infraestrutura de uma rede óptica (Dorf, 2003).

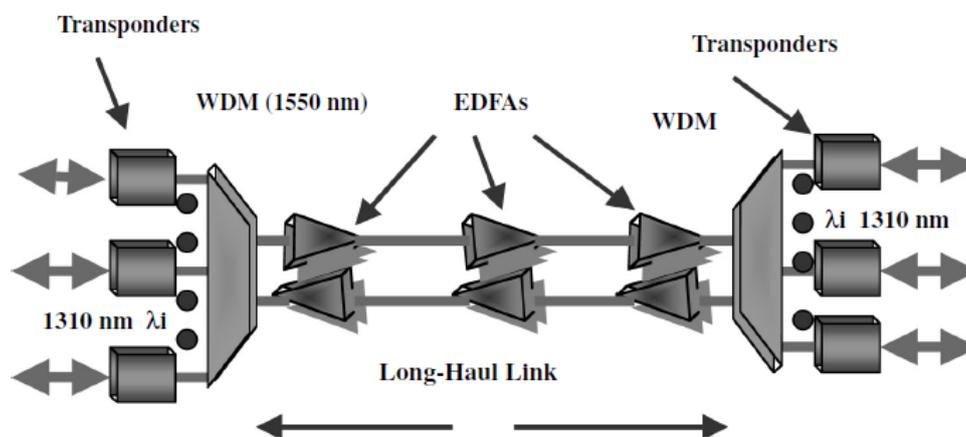


Figura 3-12 Representação esquemática de um enlace WDM ponto-a-ponto (Dorf, 2003)

Nesse esquema, o enlace de um ou mais comprimentos de onda é, então, atribuído, multiplexado, amplificado e transportado através de duas fibras para o fluxo do tráfego, uma para subida de dados (*upstream*) e outra para descida (*downstream*). Os *transponders* são responsáveis por preparar o sinal que chega dos enlaces para que sejam multiplexados em comprimentos de onda diferentes. O EDFA (*Erbium-Doped Fiber Amplifiers*) é responsável por amplificar o sinal óptico antes de lançá-lo na fibra para transporte. Normalmente é

necessário um EDFA para cada 120km de seção da fibra (Raja, 2003). Na outra ponta acontece o mesmo processo, com a conversão feita de maneira inversa pelos *transponders*.

O uso do comprimento de onda para rotear dados é conhecido como *roteamento de comprimentos de onda*, e a rede que utiliza tal técnica é chamada de rede com roteamento de comprimentos de onda. Essa rede é composta de *switches* ou roteadores de comprimentos de onda interconectados utilizando fibras ópticas. Alguns desses nós de rede (*switches* ou roteadores) têm conexão com uma rede de acesso, que também faz a conversão de sinal óptico para eletrônico e vice versa para os demais equipamentos. As redes que carregam dados de um nó a outro, sem essa conversão eletro-óptica de comprimentos de onda, são denominadas redes completamente ópticas com roteamento de comprimentos de onda (Potharlanka, Antani, Ramamurthy, Sahasrabudde, & Mukherjee, 2003).

Outro termo bastante utilizado na literatura é o OADM (*Optical Add Drop Multiplexer*), também conhecidos como WADM (*Wavelength Add Drop Multiplexer*). Esses componentes são responsáveis por colocar (*add*) ou remover (*drop*) comprimentos de onda individuais de uma fibra óptica, sem a conversão do sinal óptico para sinal digital (Gençata, Singhal, & Mukherjee, 2003). Na Figura 3-13 é demonstrado o uso desses componentes em um enlace de comunicação óptico ponto-a-ponto, semelhante a Figura 3-12.

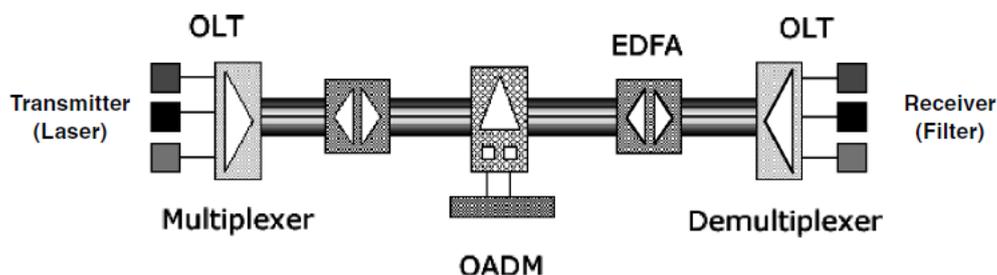


Figura 3-13 Enlace ponto-a-ponto em fibra óptica e o uso de OADM (Raja, 2003)

O OLT (Optical line terminator) apresentado na Figura 3-13 é responsável por converter sinais eletrônicos em sinais ópticos e vice-versa (Gençata, Singhal, & Mukherjee, 2003). A grande vantagem das redes WDM, além da flexibilidade permitida pelo roteamento de comprimentos de onda nos caminhos ópticos, é permitir o uso de conexões cruzadas (*cross-connects* - OXCs), que roteiam os caminhos ópticos da fonte ao destino, simplificando o gerenciamento e processamento da rede.

Já na Figura 3-14 é demonstrado o funcionamento de um WADM, com uma visão detalhada da sua estrutura interna. Os receptores não precisam demultiplexar, pois somente um comprimento de onda alcança cada um deles (Zhemini & Hamdi, 2003).

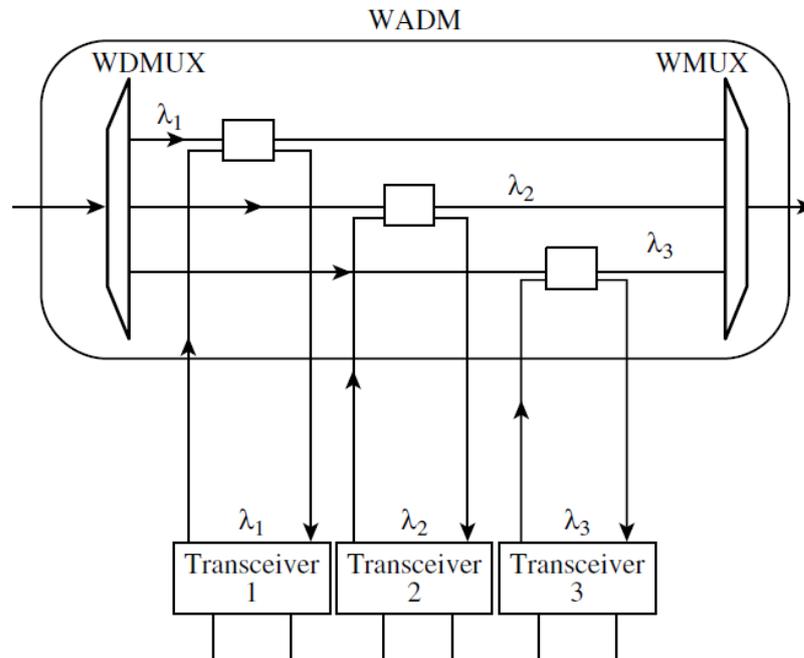


Figura 3-14 WDM para três comprimentos de onda (Zhemin & Hamdi, 2003)

Uma ação bastante comum neste tipo de rede é combinar a imensa largura de banda permitida por ela, com a flexibilidade de outras formas de rede, combinando diferentes camadas lógicas com vários serviços de usuário. Na Figura 3-15 é mostrada uma estrutura hierárquica da arquitetura multicamada das redes WDM. Nela a topologia virtual pode assumir várias formas para uma mesma camada física (Zhemin & Hamdi, 2003).

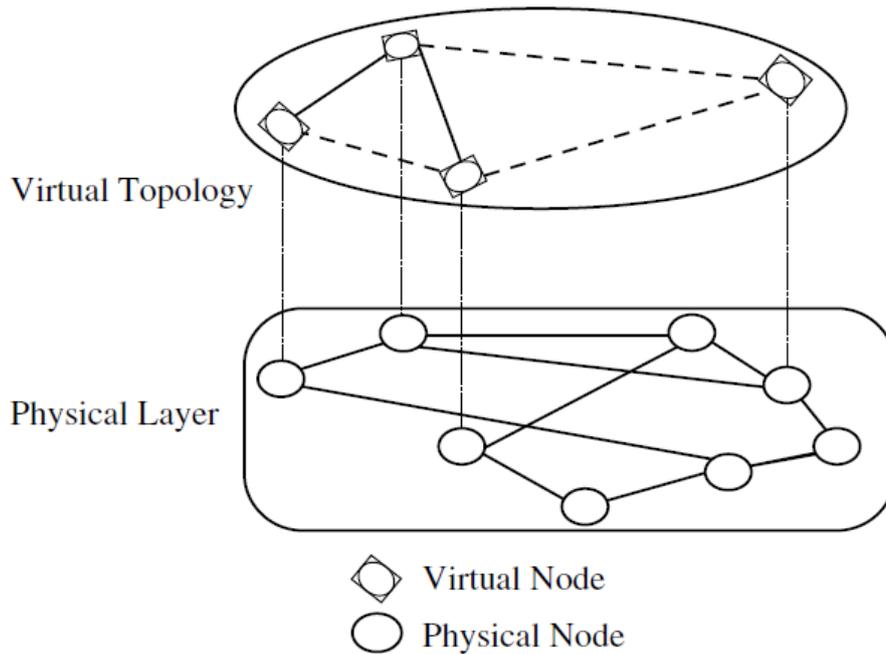


Figura 3-15 Estrutura hierárquica da arquitetura multicamadas (Zhemin & Hamdi, 2003)

3.2.7 Roteamento e atribuição de comprimentos de onda

O roteamento e atribuição de comprimentos de onda (RWA – *Routing and Wavelength Assignment*), em uma rede roteada por comprimentos de onda, os usuários podem se comunicar via canais WDM, que também são chamados de caminhos ópticos. Esses canais são utilizados para suportar uma determinada conexão e podem alcançar vários enlaces de fibra.

A Figura 3-16 mostra um exemplo de como são feitas as conexões através dos caminhos ópticos. Por exemplo, uma conexão do nó CO para o nó NJ atravessa os seguintes enlaces físicos: CO-TX, TX-GA, GA-PA, PA-NJ.

Caso não haja um conversor de comprimentos de onda, um caminho óptico ocupará o mesmo comprimento de onda na fibra seguinte do caminho que foi estabelecido. Esta propriedade é conhecida como restrição de continuidade de comprimento de onda.

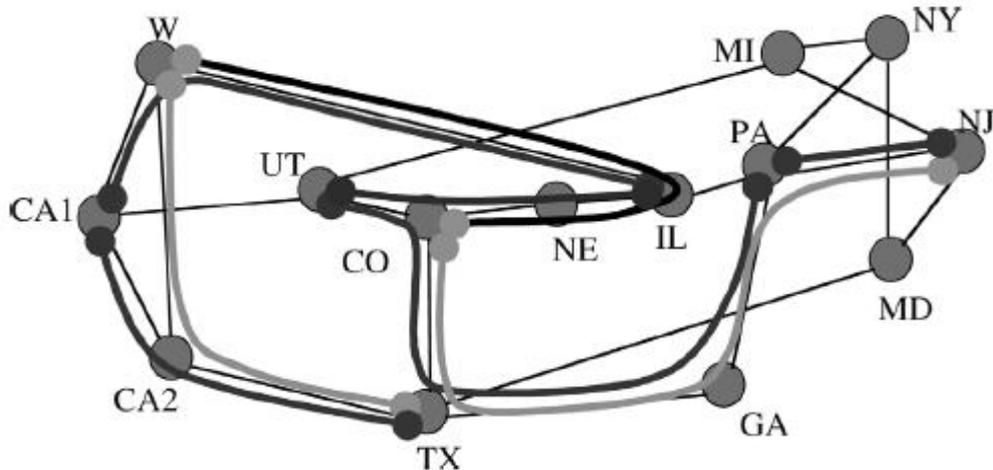


Figura 3-16 Conexões em caminhos ópticos de rede WDM (Gençata, Singhal, & Mukherjee, 2003)

Para um conjunto de conexões, o ato de decidir quais serão os caminhos ópticos roteados para conexão é chamado de problema RWA (Ramaswami & Sivarajan, Routing and Wavelength Assignment in All-Optical Networks, 1995). Esses caminhos podem, inclusive, ser bidirecionais.

O tráfego utilizado nas pesquisas da área pode ser:

- **Estático:** quando todas as requisições de conexão são previamente conhecidas.
- **Dinâmico:** as conexões chegam em uma determinada ordem e os caminhos ópticos devem ser estabelecidos de acordo com a sua chegada.

A rede pode ainda ser classificada com relação a um dos dois modelos principais de tráfego, de acordo com (Jaumard, Meyer, & Thiongane, ILP Formulations for the RWA Problem - Symmetric Systems, 2004): sistemas simétricos e assimétricos.

Sistemas assimétricos são associados a suposições de tráfego assimétricas e podem ser usados em arquiteturas de rede onde a capacidade de transporte pode ser adaptada a requisitos assimétricos de largura de banda, com uma diferença considerável entre subida e descida de dados. Por exemplo, redes WDM bidirecionais com maior quantidade de comprimentos de onda em uma determinada direção, ou mesmo sistemas unidirecionais com um maior número de comprimentos de onda em uma das direções.

Por outro lado, os sistemas simétricos são associados a suposições simétricas de tráfego. Em outras palavras, os requisitos de largura de banda são similares no sentido subida e descida de dados. Em sistemas simétricos com fibras “bidirecionais”, elas normalmente são implementadas com duas fibras unidirecionais, uma em cada sentido (Jaumard, Meyer, & Thiongane, Comparison of ILP Formulations for the RWA, 2002).

O problema RWA normalmente é dividido em outros dois subproblemas: seleção do caminho e atribuição de comprimentos de onda.

A **seleção do caminho** pode ser feita através de um roteamento estático (inclusive com rotas alternativas) e dinâmico. Para a escolha do caminho a ser seguido, existem vários algoritmos propostos na literatura. A seguir serão citadas e comentadas algumas possíveis soluções:

1. **Menor caminho (*Shortest Path*):** esses algoritmos, como o próprio nome diz, encontram o menor caminho entre a fonte e o destino de uma determinada conexão.
2. **K Menores Caminhos (*K-Shortest Paths*):** são selecionadas K possíveis rotas como solução para o problema.
3. **Roteamento adaptativo (*Adaptive routing*):** nessa família de algoritmos, a rota é escolhida de acordo com o estado atual da rede, dinamicamente. Uma possível solução é o algoritmo do caminho menos congestionado (*least congested-path* - LCP), por exemplo.

A **atribuição de comprimento de onda** tem como objetivo selecionar um dos comprimentos de onda disponíveis na rede de acordo com objetivos específicos. As políticas de atribuição de comprimentos de onda são:

1. **Aleatório (*Random wavelenght assignment*):** o comprimento de onda é definido aleatoriamente dentre todos os possíveis;
2. **Mais utilizado (*Most-used assignment*):** atribui ao comprimento de onda mais utilizado na rede, com o objetivo de deixar o restante reservado para demandas futuras;
3. **Menos Utilizado (*Least-used assignment*):** escolhe o comprimento de onda que tem maior disponibilidade na rede. É indicado para dividir o tráfego em todos os comprimentos de onda da rede;
4. **Primeira atribuição (*First-fit assignment*):** comprimentos de onda são definidos em ordem numérica e o primeiro que estiver disponível, de acordo com a ordem previamente estabelecida, é escolhido.

A **função objetivo** do problema é, tradicionalmente, a minimização da taxa de bloqueio ou maximização do número de conexões aceitas. A minimização do número de comprimentos de onda utilizados é solucionada, por exemplo, em (Lee, Kang, Lee, & Park, 2002) e (Margara & Simon, 2000). Neste caso, assume-se que todas as conexões serão

suportadas pela rede e o objetivo é usar o menor número possível de comprimentos de onda, preservando-os para futuras demandas.

Novas soluções são apresentadas em (Wauters & Demeester, 1996) e (Zang, Jue, & Mukherjee, 2000). Nesse caso, o objetivo é conseguir balancear a carga na rede, distribuindo o tráfego de maneira proporcional entre as fibras presentes.

Outra forma de classificar as redes, de extrema importância para solução do problema RWA é entre *single-hop* ou *multi-hop*, dependendo da capacidade de fazer conversão de comprimentos de onda e dos equipamentos eletrônicos presentes em cada nó. Nas redes ópticas *multi-hop* o caminho óptico é decomposto em vários segmentos, cada um associado a um comprimento de onda que pode ou não ser diferente dos anteriores, processo denominado caminhos virtuais de comprimentos de onda (*Virtual wavelength paths* - VWP). Nas redes ópticas *single-hop* uma determinada conexão entre fonte e destino é associada a um comprimento de onda e este se mantém até chegar ao destino.

4 Algoritmo

4.1 Ambiente de desenvolvimento

Segundo (Palmieri, Fiore, & Ricciardi, 2009) criar um ambiente de simulação é um método barato e efetivo de testar modelos de rede. Este método pode ajudar pesquisadores em geral a validar e mensurar o desempenho de novos protocolos e algoritmos sem a necessidade de instalar e manter infraestruturas complexas de testes para tanto.

Mesmo quando há toda uma estrutura de testes para o modelo, nem todos os algoritmos, mecanismos e protocolos podem ser realmente implementados, testados e mensurados. Isso ocorre porque a maioria dos equipamentos disponíveis para redes ópticas são de diferentes fabricantes, que normalmente não permitem muitas alterações nos equipamentos. A implementação desse tipo de modelo requer acesso, controle e programação em baixo nível, funcionalidades que dificilmente estão disponíveis para os usuários ou pesquisadores.

Além disso, há um pequeno número de ferramentas de simulação disponíveis para roteamento e atribuição de comprimentos de onda, gerenciamento de topologias, reotimização, etc. As soluções existentes hoje em dia, nesse sentido, tem escopo limitado, são difíceis de usar ou não são totalmente abertas para uso do público em geral.

Devido aos motivos citados acima foi desenvolvido um ambiente próprio, apenas no escopo deste trabalho, para fazer as simulações necessárias e poder comparar os resultados dos algoritmos sugeridos. Sua arquitetura é facilmente expansível para acrescentar novos algoritmos ou novas redes para simulação dos algoritmos testados.

Os algoritmos foram implementados em linguagem Java, atualmente uma das linguagens de mais ampla difusão e de mais abrangente uso. Além disso, traz algumas características decisivas para sua escolha no escopo deste trabalho (PET Computação - UFRGS, 2010):

- Orientação a Objetos (OO): além de simplificar o código, permite maior reuso do que já foi escrito.
- Segurança: oferece várias camadas de controle de segurança que protegem contra código malicioso.
- Otimização: otimizam a execução de blocos de código repetitivo, como laços;
- Simples: ampla verificação de erros e checagem de tipos em tempo de compilação.
- Otimizada: Gerência automática da memória (*garbage collection*).

Para ter um controle sobre as diferentes versões lançadas do sistema, foi utilizado o Git. Ele é um *software* livre de gerenciamento de código fonte que mantém um repositório com um histórico completo do desenvolvimento e habilidade total de acompanhamento das versões. Duas de suas características principais foram decisivas para a escolha neste projeto: facilidade de aprendizado e um ótimo desempenho (Torvalds, 2005).

O ambiente foi desenvolvido utilizando o padrão de projeto (*Design Pattern*) MVC, um modelo de engenharia de *software* que separa o desenvolvimento em três componentes: modelo (*model*), controlador (*controller*) e visão (*view*). Desta forma, é possível obter uma melhor visibilidade da camada de negócios do sistema, um maior reaproveitamento de código e uma manutenção facilitada. As principais classes são mostradas nos anexos [1, 2 e 3], separados de acordo com cada componente do modelo MVC.

Para acrescentar um algoritmo novo no ambiente, há uma classe mãe chamada *Algoritmo* com os atributos necessários para o bom funcionamento de qualquer algoritmo que não fuja ao escopo do trabalho. Com a ajuda do conceito de herança de programação orientada a objetos é possível criar uma classe filha e fazer uso dessa classe mãe sem maiores dificuldades. Além disso, há uma interface com métodos pré-estabelecidos para que o novo algoritmo funcione dentro do ambiente sem alterações nas demais classes.

Acrescentar uma nova rede no ambiente é ainda mais fácil. Para isso, basta criar um arquivo com o número de nós do grafo que a representa e a largura de banda de cada enlace da rede. Há dois padrões já criados no ambiente, que são carregados assim que ele se inicia, a fim de facilitar os testes e a criação de novos exemplos de rede. Os arquivos devem ser inseridos no diretório *data* do projeto, que já contém os modelos para outras redes e alguns outros arquivos de testes.

4.2 Redes Utilizadas

Conforme foi citado na seção 4.1, o usuário pode acrescentar uma rede de sua escolha ao diretório padrão das redes e, dessa forma, conseguir testar os algoritmos em uma rede qualquer. As redes inseridas não precisam existir de fato, porém, a redes configuradas inicialmente existem e serão descritas a seguir. A Rede Kyatera (Fapesp, 1998) é brasileira e foi criada com o objetivo de conectar instituições de São Paulo para fazer pesquisas em conjunto, procurando reunir competência e recursos para desenvolver a ciência, tecnologia e aplicações para a Internet no futuro. A Figura 4-1, mostra as instituições de ensino com acesso a rede atualmente.

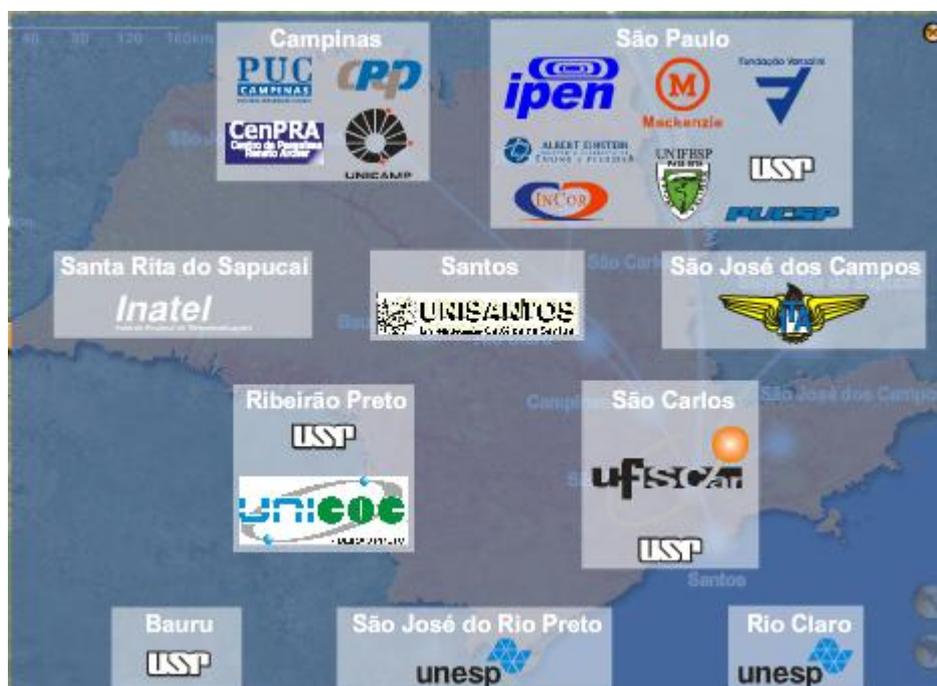


Figura 4-1 Instituições de ensino com acesso a rede Kyatera (Fapesp, 1998)

Hoje em dia, ela reúne mais de 600 pesquisadores brasileiros de diferentes áreas para executar experimentos e testar novas tecnologias, inclusive se estendendo ao estado de Minas Gerais. Além de instituições de ensino, empresas, institutos de pesquisas e agências de financiamento também tem acesso para criar um ambiente de trabalho colaborativo, geração de conhecimento e inovações tecnológicas. A Figura 4-2 mostra um mapa de abrangência da rede, que foi transformada em um grafo e inserida no sistema para testar os algoritmos implementados.

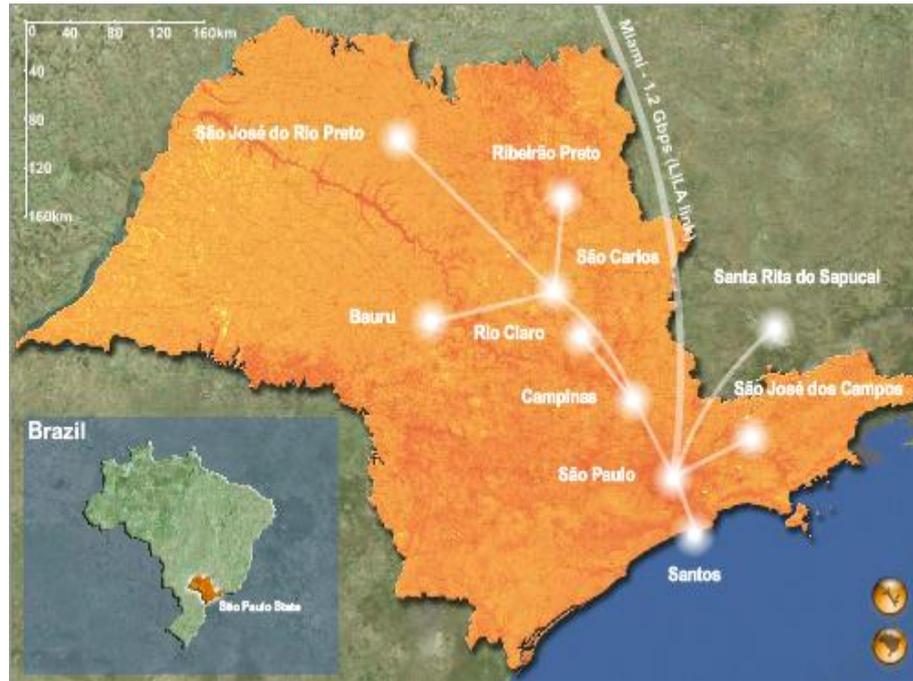


Figura 4-2 Cidades alcançadas pela rede Kyatera (Fapesp, 1998)

A outra rede utilizada foi a GÉANT2 (European Commission, Europe's national research, & Education networks, 2011), a qual é uma rede europeia de alta velocidade, que conecta 34 países, através de 30 redes de pesquisa nacionais e educacionais, com múltiplos comprimentos de onda de 10Gb/s. Assim como a rede Kyatera, procura servir a comunidade educacional e de pesquisa. Segundo seus idealizadores, sua proposta é ser o coração das pesquisas mundiais na área. A Figura 4-3 mostra a infraestrutura da rede GÉANT2, os países que ela conecta e a largura de banda dos enlaces entre os seus nós. Assim como na rede Kyatera, a rede GÉANT2 foi transformada em um grafo e inserida como um arquivo no sistema.

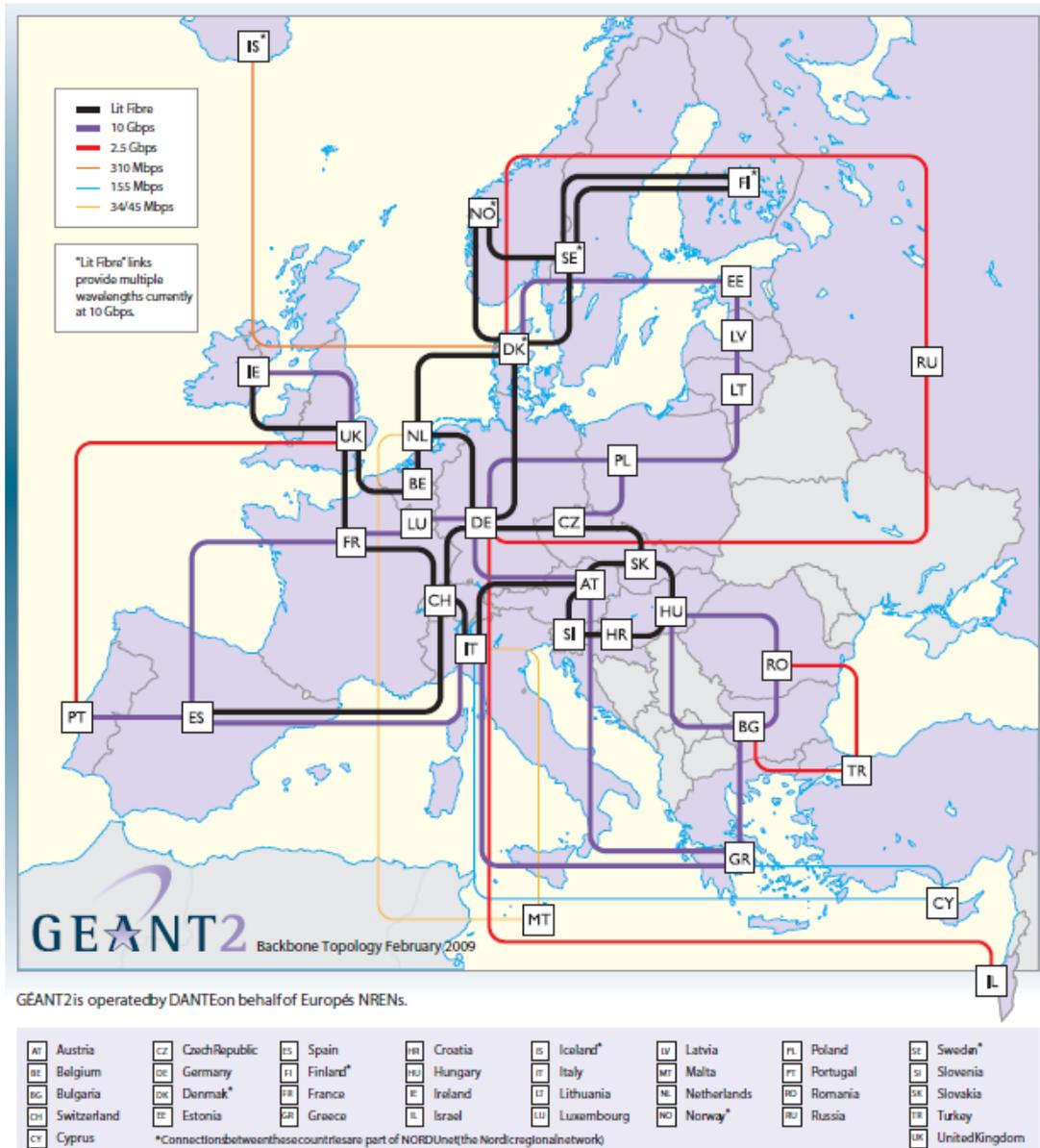


Figura 4-3 Infraestrutura da rede Geant2 (European Commission, Europe's national research, & Education networks, 2011)

Ambas são redes experimentais de alta velocidade, criadas com o intuito de promover a inovação tecnológica em redes. Dentro deste contexto, foram estudadas duas outras redes: a rede GIGA e a rede NSFNET.

A rede GIGA (Centro de Pesquisa e Desenvolvimento - CPqD, 2005) é uma rede brasileira do projeto do Centro de Pesquisa e Desenvolvimento (CPqD) de pesquisa aplicada para o desenvolvimento tecnológico de redes e serviços de telecomunicações. A rede GIGA não pôde ser utilizada como base para as simulações por ter poucos nós e ser descentralizada, atributos que mesmo isoladamente inviabilizariam a utilização da rede como base de testes para solucionar o problema proposto neste trabalho. Se o problema fosse apenas a baixa

quantidade de nós, as simulações poderiam ter sido feitas. Porém, com a descentralização dos nós, sua utilização acabou sendo inviável. Essa descentralização dos nós é mostrada na Figura 4-4, retirada do site do projeto.

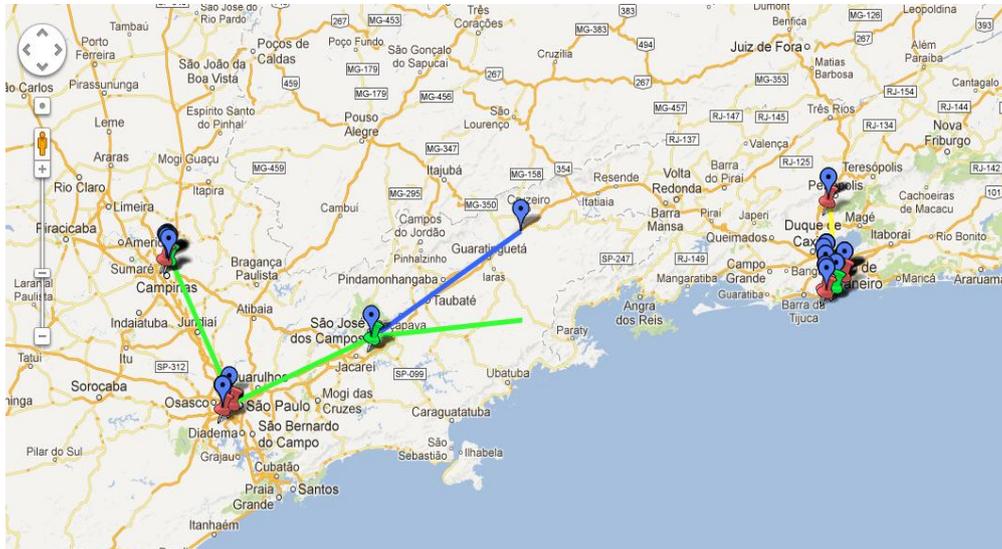


Figura 4-4 Rede GIGA (Centro de Pesquisa e Desenvolvimento - CPqD, 2005)

A rede NSFNET (National Science Foundation, 1985) é uma rede americana criada pela Fundação nacional de ciência (*National Science Foundation* – NSF) para promover pesquisas nacionais e de instituições de ensino. Assim como a rede GIGA, a rede NSFNET não pôde ser utilizada nas simulações. No caso desta rede, ela não era descentralizada e era possível calcular os tamanhos aproximados dos enlaces de rede, pois ela foi baseada no mapa americano. Porém, o problema foi encontrar dados concretos sobre a largura de banda de cada enlace. Esse dado é fundamental para que o sistema consiga calcular a taxa de bloqueio e o consumo de energia nos enlaces de maneira correta.

Para facilitar a padronização do ambiente e desenvolvimento, as redes foram transformadas em grafos, e os grafos foram numerados de cima para baixo, e da esquerda para direita, sequencialmente, de 0 a N, onde N é o número de nós do grafo. A Figura 4-5 mostra o grafo que representa a rede GÉANT2.

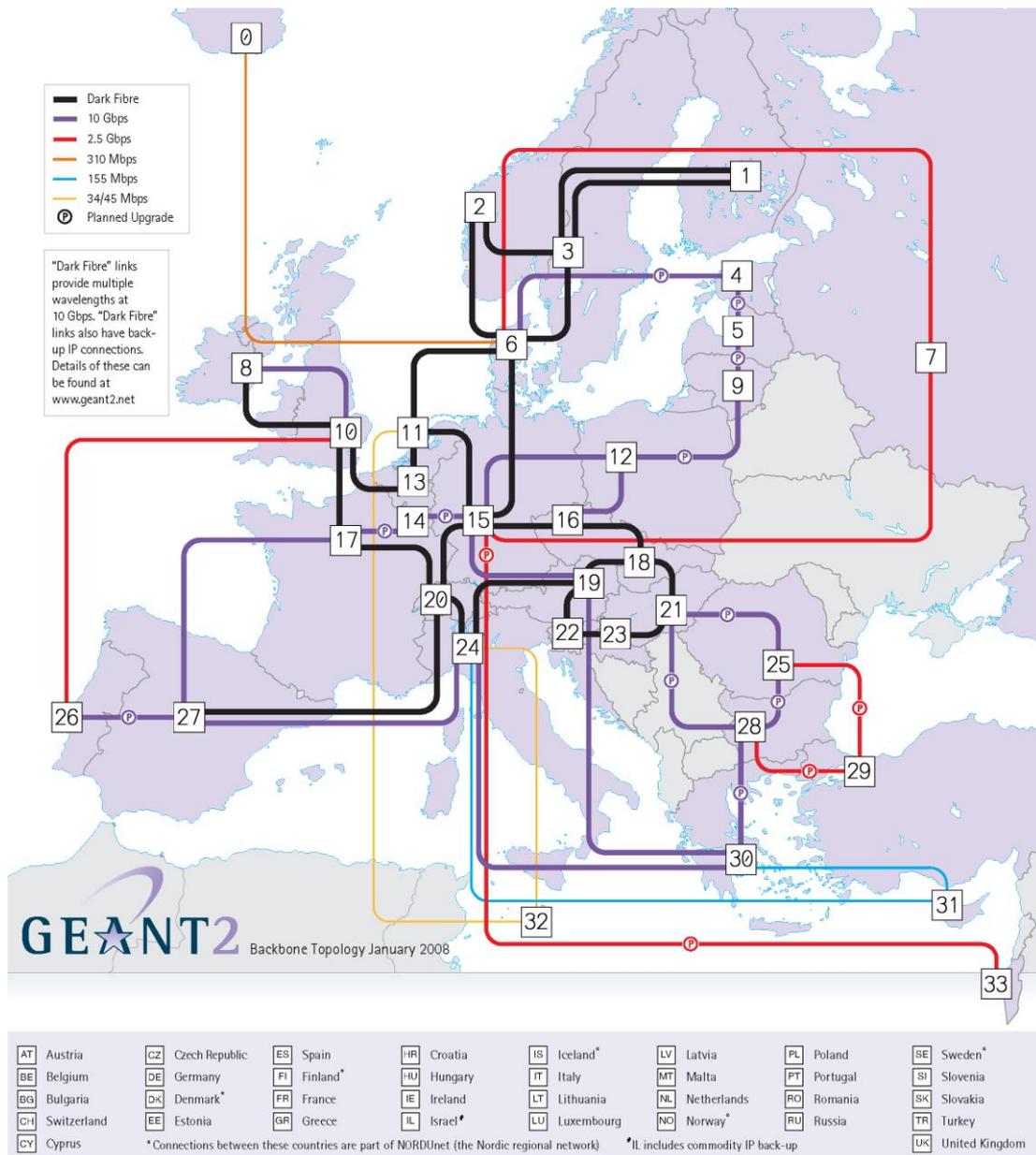


Figura 4-5 Rede GEANT2 em grafo

Assim como na Rede GEANT 2, a rede Kyatera também foi transformada em um grafo, conforme a Figura 4-6.

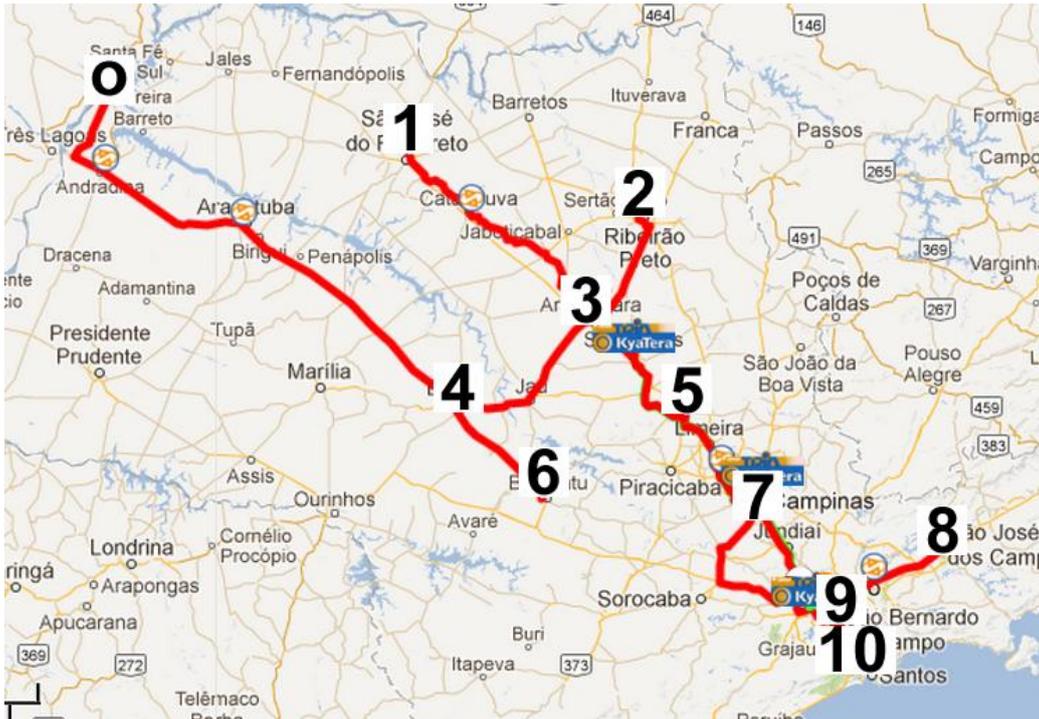


Figura 4-6 Rede Kyatera em grafo numerado (Fapesp, 1998)

4.3 O modelo de rede

Há uma enorme variedade de elementos de rede que contribuem para um alto consumo de energia em uma rede: regeneradores, amplificadores, roteadores óptico-eletrônicos, roteadores ópticos, *switches*, entre outros. Cada um deles consome energia de uma forma específica, dependendo de sua estrutura, componentes internos e volume de tráfego. As informações relacionadas a energia e conceitos associados aos dispositivos e enlaces devem ser definidos de maneira formal, com os detalhes desnecessários removidos.

Desta forma, será desenvolvido um modelo de rede de uma perspectiva de alto nível para manter o cenário o mais geral possível e conseguir manter o foco na economia de energia. Problemas como técnicas de modulação, alocação de espectro em redes elásticas, entre outras, fogem ao escopo do trabalho, que é propor um esquema para economia de energia em redes ópticas.

As estratégias RWA tem potencial para manterem-se efetivas mesmo com futuras evoluções na área de redes, em termos de otimização de recursos e utilização de energia. Portanto, serão consideradas também no modelo deste trabalho.

O modelo de rede terá como base um grafo $G = (V, E)$, onde V é o conjunto de vértices deste grafo G e E é o conjunto de arestas, letra que vem da palavra *edge* em inglês, e o número de cada um pode ser representado da seguinte maneira: $|V| = n$ e $|E| = m$.

Com relação ao número de comprimentos de onda por fibra, nenhuma suposição será feita. O número de fibras em cada enlace pode ser maior ou igual a um. Para simplificação do modelo, será considerado que todas as fibras utilizadas são do mesmo tipo, por exemplo, NZ-DSF ITU-T G.655/656. Este modelo é baseado no trabalho de (Ricciardi, et al., 2012), (Baliga, Ayre, Hinton, Sorin, & Tucker, 2009), (Shen & Tucker, 2009) e (Chabarek, et al., 2008).

A regeneração e amplificação do sinal são necessárias a cada Λ unidades, dependendo do tipo. No caso da amplificação óptica, será considerado $\Lambda_{oa} = 80\text{km}$ para sistemas com amplificação óptica. Para dispositivos 3R com regeneração eletrônica, será considerado $\Lambda_{3R} = 1000\text{km}$. Estes valores serão aplicados nas fórmulas para cálculo do consumo de energia na rede, mostradas e explicadas nos próximos capítulos.

Sendo (u,v) um par de nós fonte e destino do enlace e λ é um comprimento de onda, em cada (u,v) pode haver vários enlaces de comprimentos de onda $(u,v)_\lambda$. Onde λ pode ser qualquer um dos comprimentos de onda disponíveis no circuito físico. Cada comprimento de onda $(u,v)_\lambda$ tem sua capacidade global, $a_{(u,v)}$, e sua capacidade residual, $r_{(u,v)}$. Para calcular a carga atual no comprimento de onda, basta fazer $a_{(u,v)} - r_{(u,v)}$.

O tráfego de rede pode ser multiplexado por divisão de tempo (TDM), com processamento eletrônico, ou multiplexado por divisão de comprimentos de onda (WDM), óptico com ou sem conversão de comprimentos de onda. Roteadores eletrônicos têm a capacidade de adicionar ou remover tráfego da rede, fazer conversão de comprimentos de onda e regenerar o sinal em domínio eletrônico. Os roteadores ópticos só aceitam o tráfego no domínio óptico e podem ou não ter conversão de comprimentos de onda. O tipo de tráfego na rede também exerce influência sobre o consumo de energia e também será explicado mais a frente.

4.4 O algoritmo Todo Verde

O algoritmo aqui implementado, denominado “Todo Verde” tem por objetivo conseguir uma economia de energia em redes ópticas, através do roteamento otimizado das conexões. Para mensurar os resultados e poder compará-los a outra solução foi

implementado, também, o algoritmo *GreenSpark* (Ricciardi, et al., 2012). Ademais, os valores de conexões foram padronizados de acordo com os resultados de outros trabalhos para tornar possível, ainda, a comparação com outros algoritmos, como: Mira e MHA.

Tanto o algoritmo Todo Verde, quanto o *GreenSpark* utilizam o algoritmo *K-Shortest Paths* (Pascoal & Martins, 2003). Sua ideia é resolver o problema de encontrar K caminhos entre um nó fonte e um nó destino, buscando primeiro o menor caminho, depois o segundo menor e assim sucessivamente.

O algoritmo *GreenSpark* é voltado para economia de energia e redução da emissão de gases poluentes. Como neste trabalho o intuito é focar na economia de energia, o segundo objetivo foi removido também do *GreenSpark*. Ele é dividido em duas fases, a primeira é chamada de pré-seleção e consiste em selecionar K menores caminhos que satisfaçam as restrições de conexão utilizando o algoritmo *K-Shortest Paths*. Na segunda fase, um dos caminhos selecionados na fase anterior é escolhido, de acordo com o seu consumo total de energia e volume de gases poluentes emitidos por ele.

O algoritmo Todo Verde, proposto neste trabalho, também é dividido em duas fases. Na primeira fase são selecionados um número de caminhos equivalentes a equação 4.1:

$$K = \ln(BW_{\max}) \quad \text{Eq. 4.1}$$

Este parâmetro K é utilizado para rodar o algoritmo *K-Shortest Paths*, selecionando um número de caminhos de acordo com a largura de banda dos enlaces da rede e, dessa forma, tornando-se mais escalável. O parâmetro BW_{\max} corresponde a largura de banda máxima dos enlaces da rede em Gbps. Se uma determinada rede possui enlaces de 1Gbps, 2Gbps e 10Gbps, por exemplo, o valor do parâmetro BW_{\max} será 10.

A equação acima é uma das principais contribuições deste trabalho. Através dela é possível chegar a um resultado K adequado ao porte da rede. Isto porque, se o valor do K for igual a um, ou muito pequeno para o tamanho da rede, haverão poucos caminhos possíveis para otimizar o roteamento e economizar energia. Por outro lado, se o valor de K for muito alto, poderá ser exigido um poder de processamento muito grande e o *hardware* ficará sobrecarregado.

Levando os requisitos acima em consideração chegou-se a conclusão de que a equação ideal pra cálculo do número de caminhos era representada pela curva inversa da exponencial. Ela foi escolhida por que apresenta uma curva com crescimento acelerado no

início (necessário para obter uma melhoria no consumo, mesmo em redes menores) e um crescimento mais lento para valores maiores (para que não exija muito poder de processamento dos roteadores). Esta curva é demonstrada na Figura 4-7:

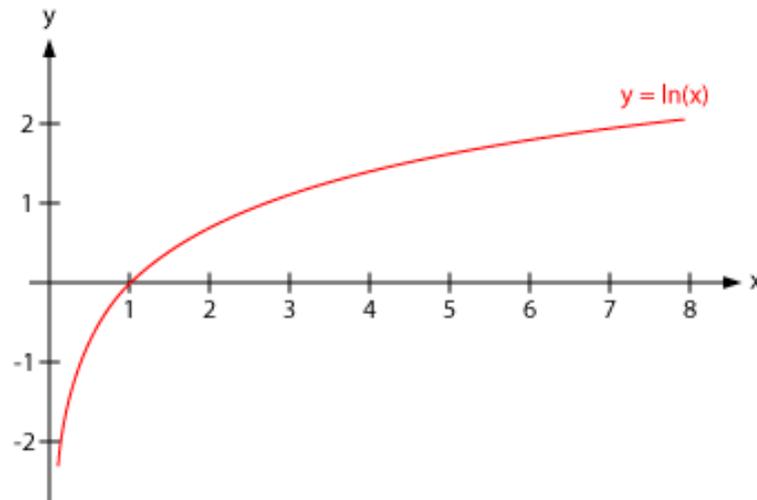


Figura 4-7 Equação $\ln(x)$

A segunda fase consiste em selecionar o caminho mais econômico, dentre aqueles que foram selecionados na primeira fase. Portanto, com o algoritmo Todo Verde é possível selecionar o caminho mais econômico sem precisar definir um parâmetro K, que já é escolhido com base na largura de banda dos enlaces da rede em questão.

Além do consumo de energia dos algoritmos, outro parâmetro importante, que será utilizado para mensurar a eficiência do Todo Verde em comparação com outros é a taxa de bloqueio. Um bloqueio acontece quando não há um caminho entre o nó fonte e o nó destino, ou não há um caminho com capacidade residual suficiente para suportar essa conexão. A probabilidade de bloqueio ou taxa de bloqueio nada mais é do que o número de bloqueios dividido pelo número de conexões, multiplicado por cem, por ser em porcentagem. A equação 4.2 é utilizada para fazer este cálculo (Ramaswami & Sivarajan, Routing and Wavelength Assignment in All-Optical Networks, 1995):

$$\text{Taxa de bloqueio} = ((\text{número de bloqueios}) / (\text{número de conexões})) \quad \text{Eq. 4.2}$$

4.5 Peso dos enlaces

O parâmetro peso dos enlaces é utilizado para definir por qual caminho é mais interessante estabelecer uma determinada conexão na fase 1 do algoritmo Todo Verde, segundo os parâmetros tradicionais nas redes de telecomunicações. Este peso nos enlaces é calculado de acordo com a equação 4.3, apresentada originalmente em (Palmieri, Fiore, & Ricciardi, 2007):

$$\text{Peso do Enlace} = \text{Capacidade Residual} * \log (\text{Capacidade Global}^{-1}) \quad \text{Eq. 4.3}$$

A capacidade global dos enlaces é fixa, enquanto a capacidade residual se atualiza conforme as conexões são estabelecidas no sistema. Portanto, este algoritmo é dinâmico (Zang, Jue, & Mukherjee, 2000).

Caso um dos enlaces calculados pelo algoritmo não tenha largura de banda suficiente para suportar a conexão estabelecida, um novo caminho é calculado no lugar deste. Ao fim da fase 1 do algoritmo Todo Verde é exibido o número de caminhos encontrados entre o nó fonte e o destino da conexão.

4.6 Cálculo do consumo de energia

Para quantificar o consumo de energia em um determinado caminho óptico de maneira realista é necessário considerar o consumo de todos os elementos de rede utilizados ao longo deste caminho, dos nós aos enlaces de transmissão como uma função envolvendo o tipo de tráfego. O consumo de energia em um caminho óptico depende de três fatores, segundo (Aleksic, 2009):

1. O tipo dos equipamentos atravessados ao longo do caminho. Por exemplo: roteador, *switch* e amplificador.
2. A classe do dispositivo, em termos de *hardware* e desempenho de acordo com o tráfego atual. Segundo (Vereecken, Heddeghem, Colle, Pickavet, & Demeester, 2010), a tendência é que dispositivos maiores consumam um número menor de energia por bit que trafega por ele.
3. O tipo de tráfego: eletrônico, óptico sem conversão de comprimentos de onda e óptico com conversão de comprimento de onda.

O consumo de energia em domínio eletrônico é 150 vezes maior do que em domínio óptico sem conversão de comprimento de onda e 50 vezes maior que em domínio óptico com conversão de comprimento de onda (Ricciardi, et al., 2011).

Nos dispositivos atuais, comercialmente disponíveis, apenas de 2 a 3% do consumo de energia varia de acordo com o tráfego de rede. Dessa forma, nestes dispositivos uma otimização no consumo de energia resultaria em uma mudança mínima no consumo em números gerais.

Em (Chabarek, et al., 2008), entretanto, é mostrado que em dispositivos da nova geração com consciência energética (*energy-aware*) tendo os equipamentos de rede ociosos (sem tráfego), aproximadamente metade da energia é consumida. Enquanto a outra metade é consumida de maneira proporcional ao tráfego de rede. Ou seja, nestes dispositivos é possível fazer um ajuste dinâmico do consumo que seja realmente eficiente em termos de resultados alcançados.

Portanto, neste trabalho será considerado que metade do consumo de energia é fixo, para manter os equipamentos de rede ligados, e a segunda metade é variável. A otimização proposta aqui tem por objetivo reduzir essa segunda metade, já que o desligamento/*sleep mode* não será considerado, por causa dos motivos citados no Capítulo 1.

Para calcular o gasto de energia em cada nó, foi utilizada a equação proposta em (Ricciardi, et al., 2011), que calcula o consumo com base na largura de banda presente no nó em questão. A Tabela 4-1, apresentada originalmente no mesmo artigo da equação, mostra os valores máximos de consumo de energia (em Watts) para cada tipo de nó, como função da sua largura de banda (em Gbps).

Tipo do nó	Consumo de energia (y) como função da largura de banda agregada (x)	Consumo de energia (y) como função da largura de banda agregada (x), assumindo metade fixa (Φ)
Eletrônico	$y = 3x$	$y = 1,5x + \Phi$
Óptico com WC	$y = 0,062x$	$y = 0,031x + \Phi$
Óptico sem WC	$y = 0,02x$	$y = 0,01x + \Phi$

Tabela 4-1 Consumo de energia (em Watts) de acordo com a largura de banda agregada (Ricciardi, et al., 2011)

Em outras palavras, de acordo com a tabela acima, m unidades de tráfego em (Gbps), representa m unidades de consumo de energia em (W). Por exemplo, um roteador eletrônico com 10Tbps de largura de banda agregada, possui um consumo máximo de 30kW, utilizando a equação $y=3x$ na primeira linha da tabela. Com o intuito de validar este modelo, os valores foram comparados com os trabalhos (B. Project, 2009) e (Chabarek, et al., 2008) e os resultados foram praticamente idênticos.

Na equação 4.4 é definida a função de consumo de energia $\Psi_n(x)$ em um nó n , apresentada em (Ricciardi, et al., 2012). Os demais parâmetros serão definidos nas próximas equações.

$$\Psi_n(x) = \alpha_n(x) \cdot \theta_n(x) + (1 - \alpha_n(x)) \cdot \vartheta_n(x) \quad \text{Eq. 4.4}$$

Na equação 4.5 é calculada $\theta_n(x)$, uma função logarítmica que modela a energia consumida por bit no melhor caso, onde a primeira parte representa o consumo variável de energia e a segunda os 50% necessários para manter o nó funcionando. O parâmetro B_n representa a largura de banda agregada no nó n .

$$\theta_n(x) = \left[\phi_n - \ln \left(\frac{e^{\phi_n}}{B_n} \cdot (B_n - x) + \frac{x}{B_n} \right) \right] + \phi_n \quad \text{Eq. 4.5}$$

A equação 4.6 representa $\vartheta_n(x)$ uma função linha que modela a energia consumida por bit no pior caso e assim como na equação 3.5, a primeira parte representa o consumo variável de energia e a segundo o consumo fixo.

$$\vartheta_n(x) = \frac{\phi_n}{B_n} + \phi_n \quad \text{Eq. 4.6}$$

O parâmetro de peso entre $\theta_n(x)$ e $\vartheta_n(x)$ é calculado na equação 4.7, definido por $\alpha_n(x)$, $\theta_n(x)$, $\vartheta_n(x)$ depende da classe e desempenho do roteador n . O parâmetro $\beta(x)$ é o que considera o tipo de tráfego que passa pelo nó n .

$$\alpha_n(x) = \frac{B_n}{\max\{B_n, \forall n \in V\}} \cdot \beta(x), \quad 0 \leq \alpha \leq 1 \quad \text{Eq. 4.7}$$

Na equação 4.8 são mostrados os valores para o parâmetro $\beta(x)$, de acordo com o tipo de tráfego agregado no nó. Esses valores foram retirados da equação 4.4, que mostra valores médios de consumo de energia em roteadores.

$$\beta(x) = \begin{cases} 1, & \text{se o tráfego é óptico sem WC} \\ 0,323 & \text{se o tráfego é óptico com WC} \\ 0,006 & \text{se o tráfego é eletrônico} \end{cases} \quad \text{Eq. 4.8}$$

$$0 \leq \beta \in \{(1), (0,323), (0,006)\} \leq 1$$

O consumo de energia fixo foi obtido com base em (Aleksic, 2009), (B. Project, 2009). A largura de banda em cada nó foi calculada com base nas conexões, que são estabelecidas dinamicamente.

Utilizando os cálculos apresentados acima, foi possível determinar a quantidade de energia consumida pelos nós da rede. O consumo de energia em cada enlace da rede, no entanto, deve ser calculado separadamente, levando em conta alguns parâmetros adicionais. Para isso, o *hardware* dos pontos finais, de amplificação e regeneração dos sinais deve ser considerado. Estes valores foram calculados de acordo com a equação 4.9, cujos parâmetros são listados a seguir (Ricciardi, et al., 2012):

- λ é um comprimento de onda.
- (u,v) é o par de nós fonte e destino do enlace.
- $\Psi_{(u,v)\lambda}(x)$ representa o consumo de energia do enlace, no comprimento de onda λ , entre o par de nós (u,v) .
- $w_{u,v}$ é o número de comprimentos de onda na fibra.
- $I_{u,v}$ é o tamanho da fibra.
- ϕ_n consumo de energia fixo.
- Λ_{OA} tamanho máximo do enlace sem necessidade de amplificadores ópticos.
- Λ_{3R} tamanho máximo do enlace sem necessidade de processamento eletrônico.
- $\eta_{(u,v)\lambda}^u(x)$ é o consumo de energia na interface do nó u , associado com o comprimento de onda λ na fibra (u,v) trabalhando com velocidade mínima.
- $Q_{u,v}$ é o consumo associado ao dispositivo de amplificação no enlace (u,v) . Para simplificar o modelo, será considerado que todos os amplificadores do enlace consomem a mesma quantidade de energia.

- $R(x)$ é o consumo de energia relacionado com o tráfego eletrônico no enlace, considerando que haja dispositivos eletrônicos para regeneração do sinal.

$$\Psi_{(u,v)\lambda}(x) = \eta_{(u,v)\lambda}^u(x) + \eta_{(u,v)\lambda}^v(x) + \frac{I_{u,v}}{\Lambda_{OA}} \cdot \frac{Q_{u,v}}{w_{u,v}} + \frac{I_{u,v}}{\Lambda_{3R}} \cdot R(x) \quad \text{Eq. 4.9}$$

A equação 4.9 é utilizada para calcular o consumo de energia em um determinado enlace da rede. Como o objetivo deste trabalho é apontar uma solução para o problema PA-RWA, é necessário ter o consumo de energia por cada caminho óptico estabelecido, e este sim será usado como comparativo nos resultados finais. O consumo por caminho óptico é calculado somando-se o consumo de energia dos nós, dos enlaces da rede, dos amplificadores e regeneradores, através da equação 4.10. Nesta equação estão presentes os parâmetros listados a seguir (Ricciardi, et al., 2012):

- π é um caminho óptico;
- x é a demanda de tráfego;
- $\Psi_{\pi}(x)$ é o consumo de energia do caminho óptico π ;
- I_{π} é o comprimento total do caminho óptico.

$$\Psi_{\pi}(x) = \sum_{n \in \pi} \Psi_n(x) + \sum_{(u,v)\lambda \in \pi} \Psi_{(u,v)\lambda}(x) + \frac{I_{\pi}}{\Lambda_{3R}} \cdot R(x) \quad \text{Eq. 4.10}$$

4.7 Fluxograma

Na Figura 4-8 é mostrado um fluxograma do algoritmo Todo-Verde, que resume o seu funcionamento desde o estabelecimento da função peso nos enlaces até o cálculo do consumo total de energia da rede. Para simplificar a explicação do algoritmo, seus passos foram numerados de 1 a 9 e cada um deles será detalhado a seguir.

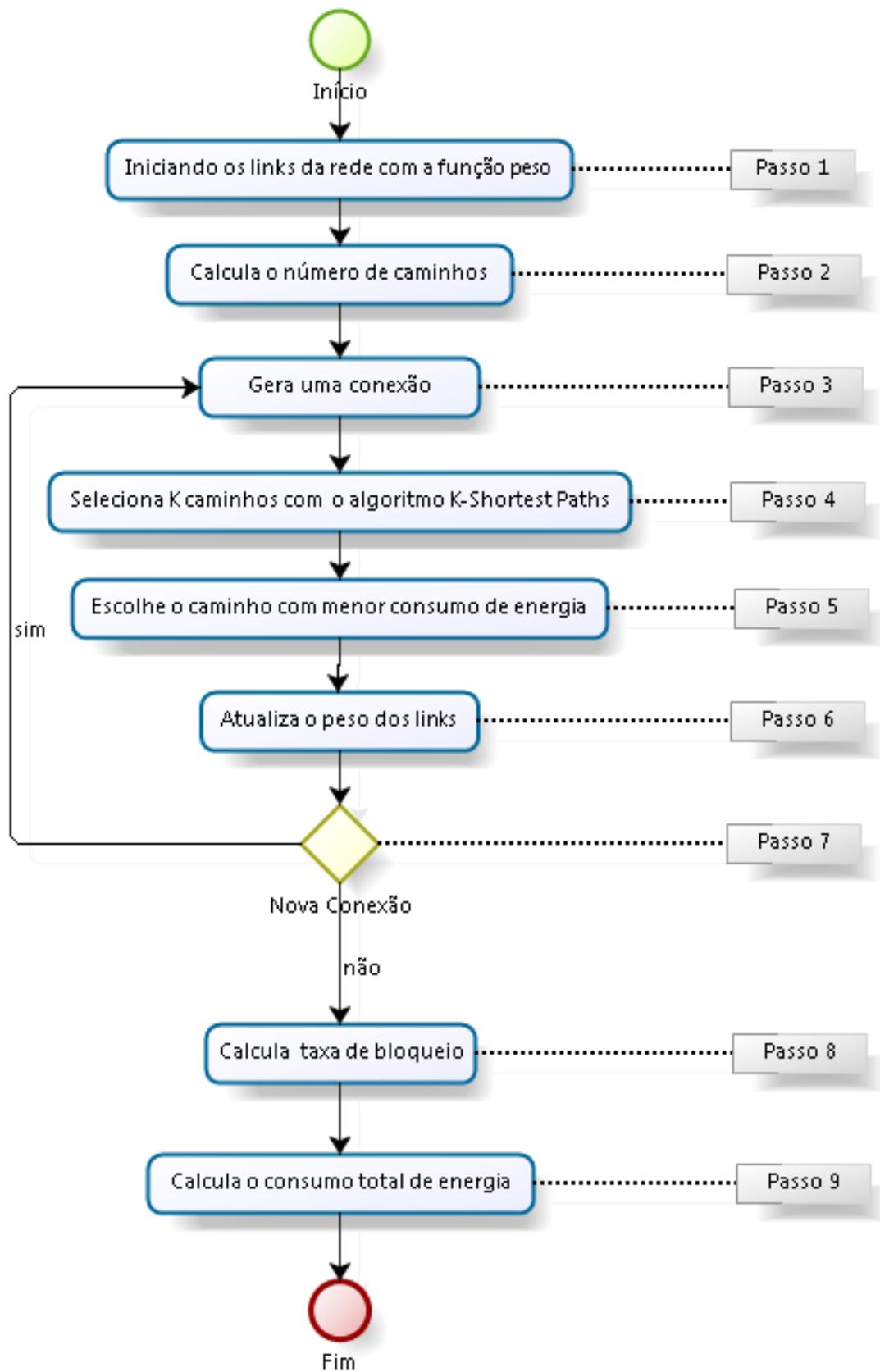


Figura 4-8 Fluxograma do algoritmo Todo-Verde

- Passo 1: Os enlaces são inicializados com os dados fornecidos pelo usuário no arquivo de configuração da rede. Inicialmente, o enlace possui sua capacidade residual igual ao valor de sua capacidade global. Os valores dos pesos dos enlaces são atualizados de acordo com a função “Peso dos enlaces”, explicada anteriormente.
- Passo 2: calcula o valor K, ou seja, o número de caminhos que serão selecionados na fase 1 do algoritmo Todo Verde. É este mesmo valor K que será utilizado posteriormente pelo algoritmo *K-Shortest Paths*. O valor de K é calculado automaticamente pelo algoritmo, de acordo com os dados da rede. Essa é uma das principais melhorias propostas neste trabalho e a equação 4.1, que realiza este cálculo.
- Passo 3: uma conexão é gerada, com um nó fonte e um nó destino sorteados aleatoriamente entre todos os vértices do grafo da rede. Vale lembrar que o sistema coloca uma restrição para que não haja uma conexão de um nó para ele mesmo, ou seja, o nó fonte deve necessariamente ser diferente do nó destino. Além disso, deve ser sorteada uma largura de banda para esta conexão, dentre aquelas que foram predefinidas para esta rede.
- Passo 4: utilizando o algoritmo *K-Shortest Paths* e o valor de K calculado no passo 2, são selecionados K caminhos do nó fonte ao nó destino, levando em conta o peso dos enlaces da rede. Este passo é a etapa fundamental da fase 1 do algoritmo Todo Verde e onde esta fase termina.
- Passo 5: este passo é a principal etapa da fase 2 do algoritmo Todo Verde. Nele é selecionado um dos caminhos selecionados na fase anterior. O critério de seleção utilizado pelo algoritmo é o consumo de energia dos caminhos, escolhendo aquele que proporciona maior economia de energia.
- Passo 6: atualiza o peso e a capacidade residual dos enlaces da rede, de acordo com a equação 4.3.
- Passo 7: testa se o sistema atingiu o limite de conexões estipulado inicialmente ou para o par (algoritmo, rede) que está sendo executado. Se não houver atingido esse limite ainda, volta ao passo 3. Caso contrário vai para o passo 8.
- Passo 8: contabiliza todas as conexões em que foi gerado um bloqueio e calcula a probabilidade de isto acontecer para o par (algoritmo, rede) atual. A equação 4.2 foi utilizada para realizar este cálculo.

- Passo 9: aqui é feita uma somatória do consumo de energia dos caminhos em todas as conexões. Ou seja, neste passo é calculado o consumo total de energia para o par (algoritmo, rede) atual. O cálculo do consumo de energia em cada caminho é a somatória do gasto com os nós e os enlaces pelo qual ele passa. A equação 4.9 foi utilizada para calcular o peso nos enlaces e a equação 4.4 para cálculo do peso dos nós.

5 Resultados

5.1 Parâmetros utilizados

O ambiente criado permite que sejam simulados vários algoritmos em diversas redes simultaneamente. Para isso, devem ser criados pares do tipo (algoritmo, rede) na execução do sistema. Portanto, o usuário tem opção de entrar com os algoritmos e redes desejadas para realizar as simulações. Obviamente, os algoritmos devem ser previamente implementados, conforme explicado anteriormente e o arquivo de configuração da rede inserido no diretório destinado a isso no sistema.

Para cada par (algoritmo, rede) criado é possível estabelecer um limite de consumo de energia no sistema. Além dos limites individuais, é possível também criar um limite global de consumo de energia para cada (algoritmo, rede) criados. Dessa forma, quando houver energia limitada em uma determinada região, é possível limitar o consumo de energia da rede a um valor específico para um dado número de conexões.

Após a criação de um determinado par de (algoritmo, rede) é possível estabelecer parâmetros específicos para os algoritmos. Por exemplo, no caso do algoritmo *GreenSpark*, é possível configurar um parâmetro K para cada instância criada do mesmo. Isso permite uma maior flexibilidade do ambiente em relação a simulações mais variadas, podendo comparar, por exemplo, o desempenho de uma instância do *GreenSpark* com $K=3$ e outra com $K=5$, além de facilitar a criação de diferentes algoritmos, que podem ter parâmetros específicos assim como há no *GreenSpark*.

Assim como o consumo de energia, o estabelecimento do número de conexões pode ser feito para cada par (algoritmo, rede) do sistema. É claro que para poder comparar o desempenho de algoritmos diferentes na mesma rede é necessário utilizar o mesmo número de conexões, mas isso da liberdade para, por exemplo, testar os mesmos algoritmos em duas redes diferentes com um número de conexões para cada uma. Em redes menores, como a

Kyatera, por exemplo, pode ser de interesse do usuário testar com um número menor de conexões.

Com relação a largura de banda de cada enlace e taxas de transmissão de cada conexão, foi utilizado o padrão *OC-Units*. OC significa *Optical carrier transmission rates*, e é um conjunto de especificações padronizadas para sinais digitais transportados em fibras ópticas SONET (*Synchronous Optical Networking*). As classificações *Optical Carrier* tem o prefixo OC, seguidas de um número N que é múltiplo de 51,84 Mbit/s, conforme a equação 5.1:

$$N \times 51.84 \text{ Mbit/s} \Rightarrow \text{OC-n} \quad \text{Eq. 5.1}$$

Por exemplo, OC-3 representa uma taxa de transmissão de aproximadamente 155 Mbit/s.

Para cada rede inserida no sistema, o usuário deve especificar a largura de banda de cada enlace na rede e alguns valores padrões de taxa de transmissão. Esses valores serão utilizados para gerar as conexões, calcular o consumo de energia, taxa de bloqueio dos pares (algoritmo, rede) e dos dados de consumo de energia, descritos no Capítulo 4. A taxa de transmissão de cada conexão é calculada da maneira aleatória, dentre os valores especificados pelo usuário ao inserir os dados da rede.

Como se trata de um ambiente de simulação, na saída é mostrado o resultado de cada conexão criada no sistema. A Figura 5-1 mostra um exemplo do resultado de uma determinada conexão no algoritmo Todo Verde sobre a rede GÉANT 2.

```

1 -----Conexao 662-----
2 Conexao do vertice 12 ao vertice 14 com largura de banda 3
3
4 Fase 1 - Caminhos encontrados:
5 :[[12, 15, 14]
6 , [12, 16, 15, 14]
7 , [12, 15, 11, 13, 10, 17, 14]
8 , [12, 15, 20, 17, 14]
9 , [12, 15, 6, 11, 13, 10, 17, 14]
10 ]
11 Numero de solucoes encontradas: 5
12
13 Fase 2 - Escolhendo o melhor caminho:
14 Consumo de energia do caminho 1: 25
15 Consumo de energia do caminho 2: 40
16 Consumo de energia do caminho 3: 48
17 Consumo de energia do caminho 4: 28
18 Consumo de energia do caminho 5: 73
19 O caminho escolhido foi [12, 15, 14]

```

Figura 5-1 Exemplo de saída de uma conexão

No exemplo acima foram geradas 1000 conexões e para cada um delas é mostrado um bloco como o da Figura 5-1. Na primeira linha da figura é mostrado o número da conexão, que neste caso é 662.

Cada conexão é feita de um vértice fonte a um vértice destino do grafo. A escolha do vértice fonte e destino de cada conexão são feitas de maneira aleatória pelo sistema, com uma única limitação de que não pode haver uma conexão de um determinado vértice para ele mesmo.

Na segunda linha é mostrado qual o nó fonte e o nó destino para esta conexão, além da sua largura de banda. Neste caso, o nó fonte é o 12, o nó destino é o 14 e a largura de banda da conexão é 3 OC-Units, que representa uma taxa de transmissão de aproximadamente 155Mbit/s. Cada rede utilizada no sistema deve ter seus nós numerados como um grafo para que sejam utilizadas no sistema, conforme explicado anteriormente.

Na quarta linha é iniciada a fase 1, com a busca pelos K melhores caminhos do grafo através do algoritmo *K-Shortest Paths*. O número K de caminhos é calculado conforme a equação 4.1.

Nas linhas 5 a 9, são mostrados os caminhos do grafo que foram selecionados como solução para a fase 1. Vale observar que todos os caminhos selecionados devem, obrigatoriamente, ir do nó fonte ao nó destino da conexão.

Neste caso, o “número de soluções encontradas: ”, na linha 11, é igual ao valor de K. Porém, este parâmetro pode assumir os seguintes valores:

- Número de soluções = 0 : não há um caminho com capacidade residual suficiente para atender a largura de banda desta conexão. Neste caso, é gerado um bloqueio para esta conexão, que será utilizado para calcular a taxa de bloqueio no fim da execução.
- $0 < \text{Número de soluções} < K$: não foram encontrados caminhos suficientes ou com capacidade residual suficiente para atingir o valor K. Mesmo assim, os caminhos encontrados serão utilizados na fase 2.
- Número de soluções = K : foram encontrados os K caminhos possíveis na fase 1.

Não há uma forma do número de caminhos encontrados ser maior do que K, pois o valor de K é calculado antes da geração de conexões e serve como condição de parada do algoritmo *K-Shortest Paths*.

Se o número de caminhos encontrados for maior ou igual a um, o algoritmo parte para a segunda fase, que tem início na linha 13 da Figura 5-1. Na segunda fase, é calculado o consumo de energia para cada caminho que foi selecionado na fase anterior, nas linhas 14 a 18, ainda da Figura 5-1.

Por fim, na linha 19 é escolhido o caminho que teve um menor consumo de energia, conforme calculado na fase 2 do algoritmo. No exemplo acima, o primeiro caminho 12-15-14 foi aquele que teve um menor consumo de energia e, portanto, o caminho selecionado.

A Figura 5-2 mostra os parâmetros de saída no fim da execução do sistema.

```

1 -----Resultados Finais-----
2 Numero de conexoes: 1000
3
4 Algoritmo GreenSpark na rede Geant2
5 Numero de bloqueios: 107
6 Probabilidade de bloqueio: 10.7%
7 Gasto de energia: 40460 Watts
8
9 Algoritmo Shortest Path na rede Geant2
10 Numero de bloqueios: 118
11 Probabilidade de bloqueio: 11.799999999999999%
12 Gasto de energia: 44744 Watts
13
14 Algoritmo TodoVerde na rede Geant2
15 Numero de bloqueios: 105
16 Probabilidade de bloqueio: 10.5%
17 Gasto de energia: 38998 Watts
18
19 Algoritmo GreenSpark na rede Kyatera
20 Numero de bloqueios: 462
21 Probabilidade de bloqueio: 46.2%
22 Gasto de energia: 17694 Watts
23
24 Algoritmo Shortest Path na rede Kyatera
25 Numero de bloqueios: 473
26 Probabilidade de bloqueio: 47.3%
27 Gasto de energia: 22826 Watts
28
29 Algoritmo TodoVerde na rede Kyatera
30 Numero de bloqueios: 472
31 Probabilidade de bloqueio: 47.199999999999996%
32 Gasto de energia: 17512 Watts

```

Figura 5-2 Parâmetros de saída dos resultados finais da simulação

Na linha 2 da Figura 5-2 é mostrado o número total de conexões do sistema, quando há um valor padrão para todos os pares (algoritmo, rede) como neste caso. O resultado de cada par (algoritmo, rede) é mostrado individualmente em um bloco separado. Por exemplo, as linhas 4-7, mostram o par (*GreenSpark*, GEANT2).

O primeiro parâmetro de cada bloco é o número de bloqueios, mostrado nas linhas 5, 10, 15, 20, 25 e 30. Seu valor representa o número de conexões que não puderam ser concretizadas, de acordo com a disponibilidade ou não de enlaces para um caminho entre o nó fonte e destino. O fato de haver um caminho de um nó para qualquer outro no grafo, como acontece na rede testada, não significa que não há bloqueios, pois a capacidade residual dos enlaces é levada em conta na atribuição dos caminhos, através da equação para cálculo dos pesos dos enlaces detalhada anteriormente.

O segundo parâmetro, mostrado nas linhas 6, 11, 16, 21, 26 e 31, é a taxa de bloqueio ou probabilidade de bloqueio, explicada no Capítulo 4.

O terceiro e último parâmetro é o consumo de energia em cada par (algoritmo, rede), mostrado nas linhas 7, 12, 17, 22, 27 e 32. Este parâmetro representa o consumo total de energia do sistema, somando-se os consumos individuais de cada caminho escolhido na fase 2 do algoritmo.

5.2 Simulações

Os algoritmos utilizados para fazer as simulações foram os seguintes:

- *Mira (Minimum Interference Routing Algorithm)*: um algoritmo dinâmico utilizado para causar o mínimo possível de interferência nos caminhos ópticos roteados.
- *MHA (Min-Hop Algorithm)*: algoritmo criado com o intuito de minimizar o número de *hops* (saltos) entre nós da rede pelos quais os caminhos serão estabelecidos.
- *GreenSpark*: tem dois objetivos principais. O primeiro é economizar energia e o segundo é reduzir a emissão de gases poluentes na atmosfera. Este é o estado-da-arte da literatura, que também soluciona o problema PA-RWA sem utilizar o modo *sleep* dos elementos de rede.
- *TodoVerde*: algoritmo proposto neste trabalho, descrito no capítulo 4.

Para mensurar os resultados do algoritmo TodoVerde foram realizadas baterias de testes com 10 execuções no programa para cada número de conexões, que variaram entre 50 e 3000. A Tabela 5-1 mostra um levantamento estatístico dos valores obtidos de média, mediana, variância e desvio padrão para os algoritmos *Mira (Minimum Interference Routing Algorithm)*, *MHA (Min-Hop Algorithm)*, *GreenSpark* e *Todo Verde* com relação a consumo total de energia.

	Mira	MHA	GreenSpark (k=3)	Todo-Verde
Média	118750	87000	34231,95	33052,81
Mediana	120500	94500	27448,5	25724,65
Variância	4799357143	2264285714	972625291	930383550,5
Desvio Padrão	69277,39	47584,51	31186,94	30502,19

Tabela 5-1 - Levantamento estatístico dos Resultados

Segundo os dados de média da tabela, o algoritmo Todo-Verde teve um desempenho 259% melhor que o Mira, 163% em relação ao MHA e 3,57% para o *GreenSpark*. Considerando os valores de mediana, a economia foi de 368% em relação ao Mira, 267% para o MHA e 6,70% melhor que o *GreenSpark*.

Esta melhora em relação aos algoritmos Mira e MHA ocorre principalmente por que eles não são voltados para o mesmo objetivo, que é economizar energia. Em comparação com o *GreenSpark*, o algoritmo TodoVerde teve um melhor desempenho por que o número de caminhos selecionados na primeira não é fixo, ele é calculado com base no tamanho da rede utilizada. Desta forma, normalmente há um maior número de caminhos selecionados e consequentemente uma maior probabilidade de otimização no consumo de energia.

Todas as simulações foram realizadas em um HP Pavilion dv7 Notebook PC, com um processador Intel Core 2 Quad CPU Q9000 2,00 Ghz, 6,00 Gb de memória RAM sobre o sistema operacional Windows Vista de 64 bits.

O gráfico apresentado na Figura 5-3 mostra o consumo de energia de cada algoritmo, quando se varia o número de conexões do sistema.

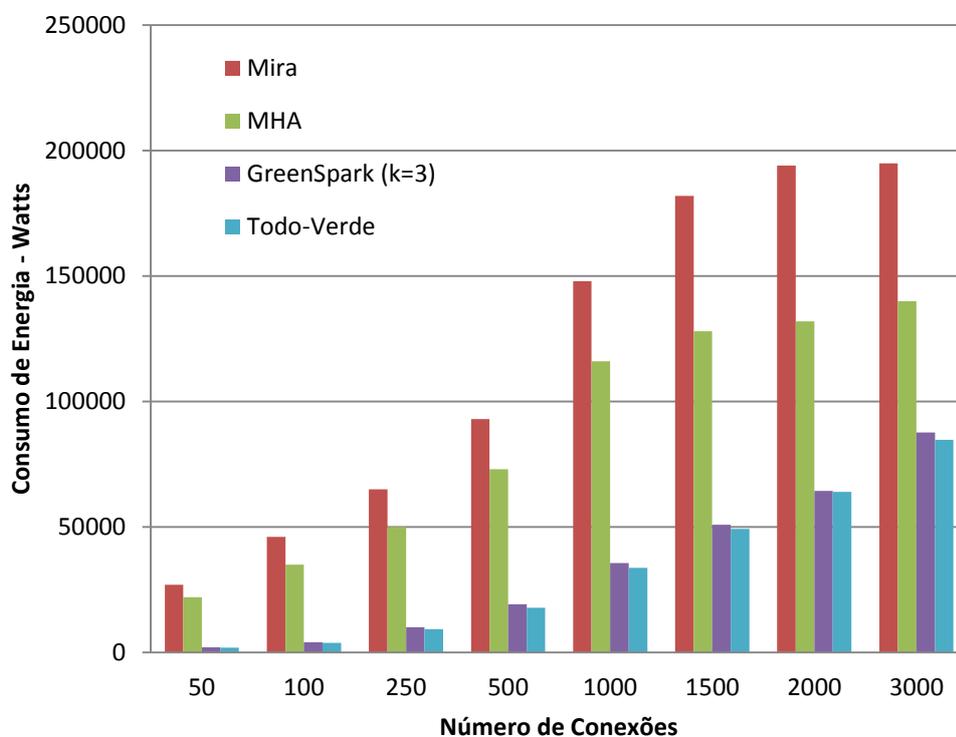


Figura 5-3 Consumo de energia x Número de conexões na rede GÉANT2

O desempenho de cada algoritmo, comparado aos outros é parecido em todos os números de conexões. Por exemplo, o algoritmo Todo-Verde manteve um desempenho melhor em todos os números de conexões, enquanto Mira apresentou um resultado menos favorável.

Os algoritmos Mira e MHA tiveram o pior desempenho para todos os valores de conexões por não serem algoritmos voltados, especificamente, para a economia de energia, enquanto o desempenho do *GreenSpark* e Todo-Verde foram bem superiores na economia de energia.

Se comparados os dois melhores desempenhos, o algoritmo Todo-Verde foi em média 4% melhor do que o *GreenSpark* com relação ao consumo de energia.

O segundo parâmetro investigado foi a taxa de bloqueio em função do consumo de energia. Diferente do consumo de energia, na taxa de bloqueio os algoritmos oscilaram entre as primeiras e últimas posições em desempenho, conforme indicado na Figura 5-4. Com um número baixo de conexões (50 e 100) os algoritmos Mira e MHA se mantiveram melhores, enquanto em um número maior de conexões os algoritmos *GreenSpark* e Todo-Verde tiveram um melhor desempenho. Considerando que um número alto de conexões é o que acontece na prática, é possível considerar que o desempenho dos últimos dois algoritmos foi melhor. O algoritmo Todo-Verde foi em média 1% melhor do que o *GreenSpark* na taxa de bloqueio.

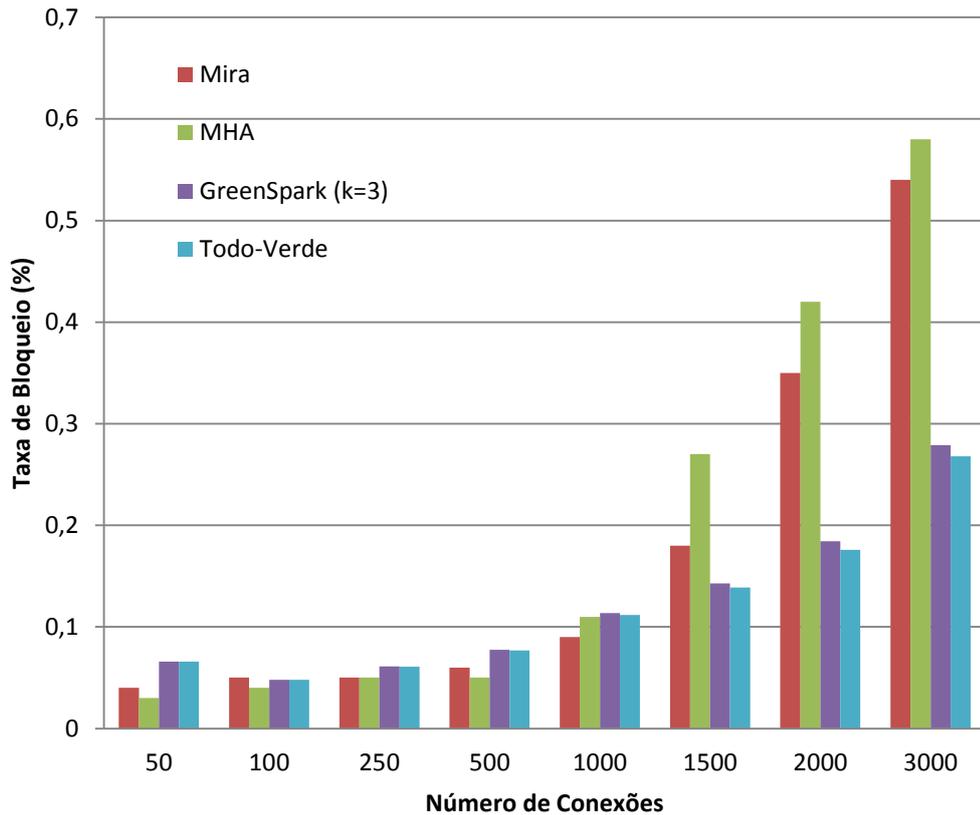


Figura 5-4 Taxa de Bloqueio x Número de Conexões

O terceiro e último parâmetro verificado nas simulações foi o valor total de energia economizado pelo algoritmo Todo Verde em relação aos outros 3 comparados neste trabalho.

Pelos resultados da Figura 5-5 foi mostrado que houve uma economia média de 1371,54 *Watts* em relação ao *GreenSpark*, que representa algo em torno de 4%. Em comparação com o algoritmo MHA, a economia média foi de 53947,17 *Watts*. Comparando o Todo Verde com o Mira, a média foi de 85797,20 *Watts*.

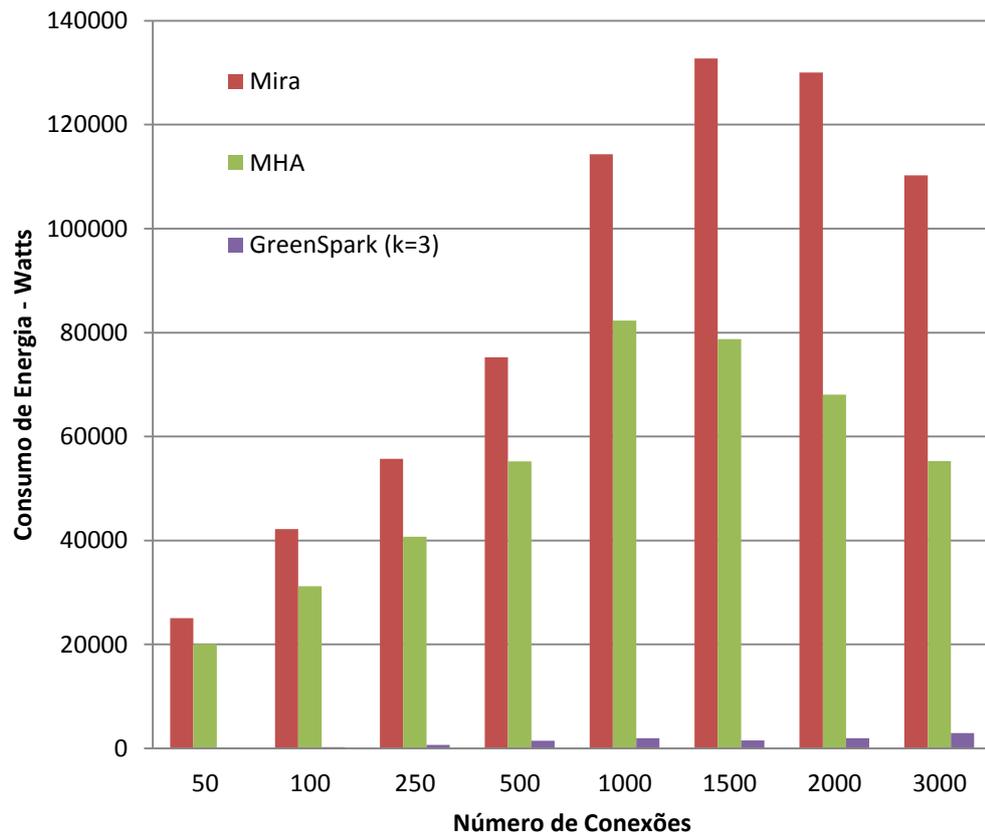


Figura 5-5 Energia Economizada x Número de Conexões

Não foi possível obter resultados satisfatórios nas simulações da rede Kyatera, pois de acordo com o grafo criado a partir da rede, há apenas um caminho a seguir de um nó fonte para qualquer outro nó destino dessa rede. Desta forma, não há espaço para uma otimização no consumo de energia, já que o algoritmo Todo-Verde economiza energia através do roteamento das conexões pelos caminhos mais econômicos.

6 Conclusão

Neste trabalho foi implementado o algoritmo Todo-Verde, voltado para economia de energia em redes ópticas *backbone*, como solução para o problema de economia de energia no roteamento e atribuição de comprimentos de onda, conhecido na literatura como PA-RWA.

6.1 Resultados obtidos

A comparação do algoritmo com soluções já implementadas e utilizadas comercialmente, como Mira e MHA mostra a grande economia de energia proporcionada pelos algoritmos PA-RWA em relação a soluções tradicionais do mercado. Ao mesmo tempo, as simulações demonstram que não há uma perda de desempenho em relação a técnicas tradicionais, conforme demonstrado nos resultados da taxa de bloqueio.

O algoritmo Todo-Verde teve um desempenho aproximadamente 3 vezes superior aos algoritmos RWA tradicionais no consumo de energia, e 4% melhor do que o *GreenSpark*, estado da arte na literatura. Com relação a taxa de bloqueio, o desempenho foi melhor para um número maior de conexões, situação mais próxima ao que acontece na prática. Comparando os valores gastos em cada algoritmo, foi possível obter uma economia média de 85 *KWatts* do Todo Verde em relação ao algoritmo Mira.

Conforme citado na introdução, o consumo de energia da *Internet* gira em torno de 22 *GigaWatts*. O consumo em redes *backbone*, gira em torno de 20% deste total, ou seja, consomem por volta de 4,4 *Gigawatts*. Com a economia proposta neste trabalho, em relação a Mira e MHA, este consumo passa a ser de aproximadamente 1,47 *Gigawatts*, com uma economia direta de 2,93 *Gigawatts*.

Portanto, o algoritmo Todo-Verde traz uma melhora considerável no consumo de energia, bem como na taxa de bloqueio em relação aos algoritmos tradicionais e, com isso, mostrou-se uma ótima opção de algoritmo PA-RWA.

6.2 Trabalhos futuros

Como trabalhos futuros é possível transformar o simulador criado em um *framework* para simulação. Uma das melhorias interessantes neste ambiente será a criação de uma interface mais intuitiva para que o usuário possa entrar com novas redes que sirvam de base para os testes e modificá-las facilmente. Além disso, o *framework* pode validar e mensurar novos protocolos de controle de rede e algoritmos para redes WDM.

Em um *framework* específico é possível trazer todas as informações relevantes em um ambiente de simulação, enquanto detalhes irrelevantes podem ser abstraídos. Desta forma, é possível simplificar a simulação e a análise das pesquisas desenvolvidas.

Esta abordagem normalmente permite às redes simuladas crescerem em tamanho e complexidade, podendo chegar a milhares de elementos de rede e comprimentos de onda. Através de um *framework* específico, é possível representar tudo de maneira simplificada, reduzindo significativamente a probabilidade de erros de programação.

Com o *framework* é criado um modelo efetivo do sistema, que garante precisão e uma maior confiabilidade nos resultados. Além disso, poderá ser aberto ao público, permitindo que várias pesquisas na área sejam testadas no mesmo ambiente, minimizando significativamente o trabalho dos pesquisadores e a possibilidade de erros por simularem em ambientes diferentes.

Trabalhos aceitos e submetidos

Congressos:

HPCS 2013 - The 2013 International Conference on High Performance Computing & Simulation

- Título: Power-Aware Routing Algorithm in Optical Networks
- Situação: Aprovado

IPC-IEEE – International Photonic Conference

- Título: An Algorithm to Reduce the Energy Consumption in Optical Networks
- Situação: Submetido

Periódicos:

IEEE Communications Letters

- Título: All-Green Algorithm as Solution for Power-Aware RWA in Optical Networks
- Situação: Submetido

Referências Bibliográficas

- Feiste, U., Ludwig, R., Schubert, C., Berger, J., Schmidt, C., Weber, H., . . . Rumpf, F. (2001). 160 Gbit/s transmission over 116 km field-installed fiber using 160 Gbit/s OTDM and 40 Gbit/s ETDM. *OFC 2001, Optical Fiber Communication Conference and Exhibit. Technical digest*, ThF3-1-3.
- When Will the Layoufss End? (1993). *Fortune*, 40.
- Ahmad, A., Bianco, A., Bonetto, E., Cuda, D., Castillo, G. G., & Neri, F. (2011). Power-Aware Logical Topology Design Heuristics. *Optical Network Design and Modeling (ONDM) - 15th International Conference on. IEEE*.
- Aleksic, S. (2009). Analysis of power consumption in future high-capacity network nodes. *IEEE/OSA Journal of Optical Communications and Networking*, 245-258.
- Assis, K., Santos, A., & Giozza, W. (2010). Hybrid Algorithms for Routing and Assignment. *IEEE LATIN AMERICA TRANSACTIONS, VOL. 8, NO. 3*, 214-220.
- Assis, K., Savasini, M., & Waldman, H. (2009). How Many Lightpaths we need Today and How Many Lightpaths we will need Tomorrow. *J. Opt. Commun.* 30, 176-179.
- B. Project. (2009). WP 21 TP green optical networks, D21.2b. *Y1 and updated plan for activities*.
- Baliga, J., Ayre, R., Hinton, K., Sorin, W., & Tucker, R. (2009). Energy Consumption in Optical IP Networks. *Journal of Lightwave Technology*, vol. 27, issue 13, 2391-2403.
- Banerjee, D., & Mukherjee, B. (1996). A Practical Approach for Routing and Wavelength. *JSAC*, vol. 14, n.5, 903-908.
- Bianzino, A. P., Chiaraviglio, L., & Mellia, M. (2011). GRiDA: a Green Distributed Algorithm for Backbone Networks. *Online Conference on Green Communications (GreenCom), 2011 IEEE*, 113-119.
- BONE project. (2009). WP 21 Topical Project Green Optical Networks: Report. *NoE, FP7-ICT-2007- 216863*.
- Bonetto, E., Chiaraviglio, L., Cuda, D., Castillo, G. G., & Neri, F. (2009). Optical Technologies Can Improve the Energy Efficiency of Networks. *ECOC, Vienna*.

- Carvalho, J. P. (2002). Redes Ópticas WDM. *Universidade do Porto, Faculdade de Engenharia - FEUP*, 1-33.
- Centro de Pesquisa e Desenvolvimento - CPqD. (2005). *Website do Projeto GIGA*. Fonte: Projeto GIGA: <http://www.giga.org.br/>
- Chabarek, J., Sommers, J., Barford, P., Estan, C., Tsiang, D., & Wright, S. (2008). Power awareness in network design and routing. *INFOCOM 2008, The 27th Conference on Computer Communications, IEEE*, 457-465.
- Christensen, K., Gunaratne, C., Nordman, B., & George, A. (2004). The next frontier for communications networks: power management. *Comput. Commun., vol. 27*, 1758–1770.
- Coiro, A., Listanti, M., Valenti, A., & Matera, F. (2011). Power-Aware Routing and Wavelength Assignment in Multi-Fiber Optical Networks. *J. OPT. COMMUN. NETW./VOL. 3, NO. 11*, 816-829.
- Dante, R. G. (2005). Algoritmos de roteamento e atribuição de comprimentos de onda para as redes ópticas inteligentes e transparentes. *Tese de doutorado, Unicamp*.
- Dorf, R. C. (2003). *The Handbook of Optical Communication Networks*. University of California, CRC Press LLC.
- Easley, D., & Kleinberg, J. (2010). *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*. Cambridge University Press, 2010.
- Ellis, A. D., Patrick, D. M., Flannery, D., Manning, R. J., Davies, D. A., & Spirit, D. M. (1995). Ultra-high-speed OTDM networks using semiconductor amplifier-based processing nodes. *Lightwave Technology, Journal of*, 13(5), 761-760.
- European Commission, Europe's national research, & Education networks. (2011). *GÉANT2 Website*. Fonte: <http://www.geant2.net/>
- Fapesp. (1998). *Rede Kyatera*. Fonte: <http://kyatera.ifi.unicamp.br/>
- Gençata, A., Singhal, N., & Mukherjee, B. (2003). Overview of optical communication networks: Current and future trends. *CRC Press 2003*, 14-38.
- Heart, F., McKenzie, A., McQuillian, J., & Walden, D. (1978). *ARPANET Completion Report*. Fonte: College of Business Administration & Public Policy: <http://som.csudh.edu/cis/lpress/history/arpamaps/>
- <http://www.dtc.umn.edu/mints/2002-2008/analysis-2002-2008.html>. (s.d.). Minnesota internet traffic studies.

- Idzikowski, F., Orlowski, S., Raack, C., Woesner, H., & Wolisz, A. (2010). Saving energy in IP-over-WDM networks by switching off line cards in low demand scenarios, in: Optical Network Design and Modeling (ONDM). *2010 14th Conference on*, 1-6.
- Jaumard, B., Meyer, C., & Thiongane, B. (2002). Comparison of ILP Formulations for the RWA. *JOURNAL OF LATEX CLASS FILES, VOL. 1, NO. 11*, 1-13.
- Jaumard, B., Meyer, C., & Thiongane, B. (2004). ILP Formulations for the RWA Problem - Symmetric Systems. 1-45.
- Jean-Pierre, L. (1993). *Wavelength division multiplexing*. Prentice-Hall.
- Júnior, L. S. (2012). *Redes de Alta Velocidade*. Fonte: Universidade Federal Rural do Semi-Árido - UFERSA: <http://www2.ufersa.edu.br/portal/view/uploads/setores/150/ROpticas.pdf>
- Kasper, J. C., Wiggenhorn, M., Resch, M., & Friess, W. (2012). Implementation and evaluation of an optical fiber system as novel process monitoring tool during lyophilization. *European Journal of Pharmaceutics and Biopharmaceutics*, pp. 449–459.
- Lee, K., Kang, K., Lee, T., & Park, S. (2002). An optimization approach to routing and wavelength assignment in WDM all-optical mesh networks without wavelength conversion. *ETRI Journal*, 24(2), 131-141.
- Margara, L., & Simon, J. (2000). Wavelength assignment problem on all-optical networks with k fibres per link. *ICALP 2000, volume 1853 of Lectures Notes in Computer Science*, 768-779.
- Murthy, C., & Gurusamy, M. (2002). WDM Optical Networks – Concepts, Design and Algorithms. *Prentice Hall PTR*.
- N. Shimojoh, e. a. (1997). *Electronnic Letters*, 33, 10, 877.
- National Science Foundation. (1985). *NSFNET Network*. Fonte: NSFNET: <http://www.nsfnet-legacy.org/>
- Neto, H. M. (2009). Arquitetura de nós e engenharia de tráfego em redes ópticas. *Tese de doutorado, Escola de Engenharia de São Carlos da Universidade de São Paulo*.
- Networking, C. (2012). *Tecnologia Fibra Óptica*. Fonte: CIANET Networking: <http://www.cianet.ind.br/pt/produtos/tecnologias/tecnologia-fibra-optica/>
- Palmieri, F., Fiore, U., & Ricciardi, S. (2007). SPARK: a smart parametric online RWA algorithm. *Journal of Communications and Networks* 9 (4), 368-376.

- Palmieri, F., Fiore, U., & Ricciardi, S. (2009). SimulNet: a Wavelength-Routed Optical Network Simulation Framework. *Computers and Communications, 2009. ISCC 2009. IEEE Symposium*, 281-286.
- Pascoal, M., & Martins, E. (2003). A new implementation of Yen's ranking loopless paths algorithm. *4OR – Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, 121-133.
- Pavon-Marino, P., Aparicio-Pardo, R., Garcia-Manrubia, B., & Skorin-Kapov, N. (2009). Virtual topology design and flow routing in optical networks under multi-hour traffic demand. *Photonic Network Communications, August. Springer Netherlands*.
- PET Computação - UFRGS. (2010). *Orientação a Objetos e Linguagem Java*. Fonte: Instituto de Informática - UFRGS: <http://www.inf.ufrgs.br/pet/cursos/Java/cursojavatotal.pdf>
- Pickavet, M., Vereecken, W., Demeyer, S., Audenaert, P., Vermeulen, B., Develder, C., & Demeester, P. (2008). Worldwide energy needs for ICT: the rise of power-aware networking. *Proc. IEEE ANTS 2008*.
- Pickavet, M., Vereecken, W., Demeyer, S., Audenaert, P., Vermeulen, B., Develder, C., . . . Demeester, P. (2008). Worldwide energy needs for ICT: the rise of power-aware networking. *Proc. IEEE ANTS 2008, Bombay*, 1–3.
- Potharlanka, K., Antani, T., Ramamurthy, B., Sahasrabudhe, L., & Mukherjee, B. (2003). Design aspects of optical communication networks. *CRC Press 2003*, 58-78.
- Raja, M. .. (2003). Evolution of optical networks architecture. *CRC Press 2003*, 42-56.
- Ramaswami, R. (2006). Optical Networking Technologies: What Worked and What Didn't. *IEEE Communications Magazine*, 132-139.
- Ramaswami, R., & Sivarajan, K. (1995). Routing and Wavelength Assignment in All-Optical Networks. *IEEE/ACM Trans. Networking, vol.3, no.5*, 489-500.
- Ramaswami, R., & Sivarajan, K. (1996). Design of Logical Topologies for Wavelength-Routed All Optical Networks. *IEEE/JSAC, vol. 14*, 840-851.
- Ricciardi, S., Careglio, D., Palmieri, F., Fiore, U., Santos-Boada, G., & Solé-Pareta, J. (2011). Energy-Aware RWA for WDM Networks with Dual Power Sources. *IEEE ICC 2011*.
- Ricciardi, S., Careglio, D., Santos-Boada, G., Solé-Pareta, J., Fiore, U., & Palmieri, F. (2011). Towards an energy-aware Internet: modeling a cross-layer optimization approach. *Springer Science+Business Media*.

- Ricciardi, S., Palmieri, F., Fiore, U., Careglio, D., Santos-Boada, G., & Solé-Pareta, J. (2012). An energy-aware dynamic RWA framework for next-generation wavelength-routed networks. *Computer Networks* 56 , 2420-2442.
- Sasaki, G., Ramaswami, R., & Sivarajan, K. (2009). *Optical Networks: A Practical Perspective*. ELSEVIER SCIENCE.
- Shen, G., & Tucker, R. (2009). Energy-minimized design for IP over WDM networks. *IEEE/OSA Journal of Optical Communications and Networking* 1, 176-186.
- T. C. Cannon, e. a. (1978). *IEEE Trans. Commun. COM-26*, 1045.
- Tanenbaum, A. S. (2003). *Computer Networks*. Fourth Edition, Prentice Hall.
- Texas A&M University. (1996). Chapter 8 - Backbone Design Overview. In: *BICSI LAN Design Manual - CD-ROM* (pp. 1-55).
- Torvalds, L. (2005). *About Git*. Fonte: Git: <http://git-scm.com/>
- University of Minnesota. (2009). *Minnesota internet traffic studies*. Fonte: University of Minnesota: <http://www.dtc.umn.edu/mints/2002-2008/analysis-2002-2008.html>
- University of Minnesota. (2009). *Minnesota internet traffic studies*. Acesso em Janeiro de 2013, disponível em University of Minnessota: <http://www.dtc.umn.edu/mints/2002-2008/analysis-2002-2008.html>
- Vereecken, W., Heddeghem, W. V., Colle, D., Pickavet, M., & Demeester, P. (2010). Overall ICT footprint and green communication technologies. *2010 4th International Symposium on Communications*, 1-6.
- Wauters, N., & Demeester, P. (1996). Design of the optical path layer in multiwavelength cross-connected networks. *IEEE Journal on Selected Areas in Communications*, 0733-8716/96, Vol. 14, Nr. 5, June, 881-892.
- Wei, L. (2002). China's optical-network evolution. *SPIE's OEMagazine*, 22-25.
- Wu, Y., Chiaraviglio, L., Mellia, M., & Neri, F. (2009). Power-Aware Routing and Wavelength Assignment in Optical Networks. *BONE research project*, 1-2.
- Zang, H., Jue, J., & Mukherjee, B. (2000). A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks. *Optical Networks Magazine*, 47-60.
- Zhemin, D., & Hamdi, M. (2003). Optical network resource management and allocation. *CRC Press 2003*, 160-181.

Anexo 1 – Principais classes “*Model*”

Anexo 1.1 - Classe Rede

```
package br.edu.unifei.Model;

import edu.asu.emit.qyan.alg.model.VariableGraph;

/**
 *
 * @author Rafael Coutinho
 */
public class Rede extends VariableGraph {
    String nome;
    int limiteDeEnergia;
    int[] larguraDeBandaPermitida;

    /**
     *
     * @param string
     * @param nome
     * @param larguraDeBandaPermitida
     */
    public Rede(String string, String nome, int[] larguraDeBandaPermitida) {
        super(string);
        this.nome = nome;
        this.larguraDeBandaPermitida = larguraDeBandaPermitida;
    }
}
```

```
public String getNome() {
    return nome;
}

public void setNome(String nome) {
    this.nome = nome;
}

public int getLimiteDeEnergia() {
    return limiteDeEnergia;
}

public void setLimiteDeEnergia(int limiteDeEnergia) {
    this.limiteDeEnergia = limiteDeEnergia;
}

public int[] getLarguraDeBandaPermitida() {
    return larguraDeBandaPermitida;
}

public void setLarguraDeBandaPermitida(int[] larguraDeBandaPermitida) {
    this.larguraDeBandaPermitida = larguraDeBandaPermitida;
}
}
```

Anexo 1.2 – Classe Link

```
package br.edu.unifei.Model;

import edu.asu.emit.qyan.alg.model.Graph;
import edu.asu.emit.qyan.alg.model.Pair;
import edu.asu.emit.qyan.alg.model.VariableGraph;

/**
 *
 * @author Rafael Coutinho
 */
public class Link{
    int fonte, destino;
    int capacidadeGlobal, capacidadeResidual, consumoDeEnergia;
    double peso;
    Graph graph;
    Pair borda;

    public Link(int fonte, int destino, double peso, VariableGraph graph){
        //this.graph = graph;
        //graph.remove_edge(new Pair(fonte, destino));
        this.fonte = fonte;
        this.destino = destino;
        this.peso = peso;
        this.capacidadeGlobal = this.capacidadeResidual = (int) peso;
    }

    public int getCapacidadeGlobal() {
        return capacidadeGlobal;
    }
}
```

```

}

public void setCapacidadeGlobal(int capacidadeGlobal) {
    this.capacidadeGlobal = capacidadeGlobal;
}

public int getCapacidadeResidual() {
    return capacidadeResidual;
}

public void setCapacidadeResidual(int capacidadeResidual) {
    this.capacidadeResidual = capacidadeResidual;
}

public double getPeso() {
    return peso;
}

public void setPeso(double peso) {
    this.peso = peso;
}

public int getConsumoDeEnergia() {
    return consumoDeEnergia;
}

public void setConsumoDeEnergia(int consumoDeEnergia) {
    this.consumoDeEnergia = consumoDeEnergia;
}

public int getDestino() {
    return destino;
}

```

```

public void setDestino(int destino) {
    this.destino = destino;
}

public int getFonte() {
    return fonte;
}

public void setFonte(int fonte) {
    this.fonte = fonte;
}

public Graph getGraph() {
    return graph;
}

public void setGraph(Graph graph) {
    this.graph = graph;
}

@Override
public String toString(){
    String str="";
    str+="[" + this.getFonte() + ", " + this.getDestino() + "] ";
    str+= " | Peso: " + (double)Math.round(this.peso * 100) / 100;
    str+= " | Capacidade Global: " + this.capacidadeGlobal;
    str+= " | Capacidade Residual: " + this.capacidadeResidual;
    str+= " | Consumo de Energia: " + this.consumoDeEnergia;
    return str;
}
}

```

Anexo 1.3 – Classe Conexão

```
package br.edu.unifei.Model;

/**
 *
 * @author Rafael Coutinho
 */
public class Conexao{
    int fonte, destino, larguraDeBandaNecessaria;

    public Conexao(int fonte, int destino, int larguraDeBandaNecessaria) {
        this.fonte = fonte;
        this.destino = destino;
        this.larguraDeBandaNecessaria = larguraDeBandaNecessaria;
    }

    public int getDestino() {
        return destino;
    }

    public void setDestino(int destino) {
        this.destino = destino;
    }

    public int getFonte() {
        return fonte;
    }

    public void setFonte(int fonte) {
        this.fonte = fonte;
    }
}
```

```
}  
  
public int getLarguraDeBandaNecessaria() {  
    return larguraDeBandaNecessaria;  
}  
  
public void setLarguraDeBandaNecessaria(int larguraDeBandaNecessaria) {  
    this.larguraDeBandaNecessaria = larguraDeBandaNecessaria;  
}  
  
}
```

Anexo 1.4 – Classe Caminho Óptico

```
package br.edu.unifei.Model;

import edu.asu.emit.qyan.alg.model.Path;
import edu.asu.emit.qyan.alg.model.abstracts.BaseVertex;
import java.util.List;

/**
 *
 * @author Rafael Coutinho
 */
public class CaminhoOptico extends Path {
    int consumoDeEnergia;

    public CaminhoOptico(List<BaseVertex> _vertex_list, double _weight) {
        super(_vertex_list, _weight);
        consumoDeEnergia = 0;
    }

    public int getConsumoDeEnergia() {
        return consumoDeEnergia;
    }

    public void setConsumoDeEnergia(int consumoDeEnergia) {
        this.consumoDeEnergia = consumoDeEnergia;
    }
}
```

Anexo 1.5 – Classe Algoritmo

```
package br.edu.unifei.Model;

/**
 *
 * @author Rafael Coutinho
 */
public class Algoritmo {
    int numeroDeBloqueios;
    int gastoDeEnergia;
    String nome;

    public Algoritmo(String nome){
        numeroDeBloqueios = gastoDeEnergia = 0;
        this.nome = nome;
    }

    public int getGastoDeEnergia() {
        return gastoDeEnergia;
    }

    public void setGastoDeEnergia(int gastoDeEnergia) {
        this.gastoDeEnergia = gastoDeEnergia;
    }

    public int getNumeroDeBloqueios() {
        return numeroDeBloqueios;
    }

    public void setNumeroDeBloqueios(int numeroDeBloqueios) {
        this.numeroDeBloqueios = numeroDeBloqueios;
    }
}
```

```
}

public void addBloqueio(){
    this.numeroDeBloqueios++;
}

public String getNome() {
    return nome;
}

public void setNome(String nome) {
    this.nome = nome;
}
}
```

Anexo 2 – Principais classes “Control”

Anexo 2.1 – Classe MainControl

```
package br.edu.unifei.Control;

/**
 *
 * @author Rafael Coutinho
 */
public class MainControl {
    int limiteDeEnergia;
    RedeControl redeControl;
    AlgoritmoControl algoritmoControl;
    ConexaoControl conexaoControl;

    public MainControl(RedeControl redeControl, ConexaoControl conexaoControl,int
algoritmo, int limiteDeEnergia, int numeroConexoes){
        this.limiteDeEnergia = limiteDeEnergia;
        this.redeControl = redeControl;
        this.conexaoControl = conexaoControl;
        algoritmoControl = new AlgoritmoControl(this, algoritmo);
        //this.conexaoControl = new ConexaoControl(generalControl, numeroConexoes,
this.larguraDeBanda, rede);
    }
    public void executar() {
        algoritmoControl.executar();
    }
}
```

```

}

public int getLimiteDeEnergia() {
    return limiteDeEnergia;
}

public void setLimiteDeEnergia(int limiteDeEnergia) {
    this.limiteDeEnergia = limiteDeEnergia;
}

public RedeControl getRedeControl() {
    return redeControl;
}

public void setRedeControl(RedeControl redeControl) {
    this.redeControl = redeControl;
}

public AlgoritmoControl getAlgoritmoControl() {
    return algoritmoControl;
}

public void setAlgoritmoControl(AlgoritmoControl algoritmoControl) {
    this.algoritmoControl = algoritmoControl;
}

public ConexaoControl getConexaoControl() {
    return conexaoControl;
}

public void setConexaoControl(ConexaoControl conexaoControl) {
    this.conexaoControl = conexaoControl;
}

}

```

Anexo 2.2 – Classe AlgoritmoControl

```
package br.edu.unifei.Control;

import br.edu.unifei.Model.Algoritmo;
import br.edu.unifei.Model.AlgoritmoGreenSpark;

/**
 *
 * @author Rafael Coutinho
 */
public class AlgoritmoControl {
    MainControl mainControl;
    Algoritmo algoritmo;
    int nome;

    public static final int K_SHORTEST_PATHS = 1;
    public static final int GREEN_SPARK = 2;
    public static final int TODO_VERDE = 3;

    public AlgoritmoControl(MainControl mainControl, int nome){
        this.mainControl = mainControl;
        this.nome = nome;
        this.algoritmo = inicializarAlgoritmo();
    }

    private Algoritmo inicializarAlgoritmo() {
        if(nome == K_SHORTEST_PATHS) { return new Algoritmo("k-ShortestPaths"); }
        if(nome == TODO_VERDE) { return new Algoritmo("TodoVerde"); }
        if(nome == GREEN_SPARK) { return new AlgoritmoGreenSpark(5, "GreenSpark"); }
    }
}
```

```

        return null;
    }

    void ejecutar() {
        if(nome == K_SHORTEST_PATHS){
            AlgoritmoKSPControl ksp = new AlgoritmoKSPControl (mainControl,
algoritmo);
            ksp.ejecutar();
        }
        if(nome == TODO_VERDE){
            AlgoritmoTodoVerdeControl atvc = new
AlgoritmoTodoVerdeControl(mainControl, algoritmo);
            atvc.ejecutar();
        }
        if(nome == GREEN_SPARK){
            AlgoritmoGreenSparkControl agsc = new
AlgoritmoGreenSparkControl(mainControl, algoritmo);
            agsc.ejecutar();
        }
    }

    public Algoritmo getAlgoritmo() {
        return algoritmo;
    }

    public void setAlgoritmo(Algoritmo algoritmo) {
        this.algoritmo = algoritmo;
    }

```

Anexo 2.3 – Classe RedeControl

```
package br.edu.unifei.Control;

import br.edu.unifei.Model.Redes;
import br.edu.unifei.View.RedesView;

/**
 *
 * @author Rafael Coutinho
 */
public class RedeControl {
    RedesView redeView;
    Redes rede;
    LinkControl linkControl;

    int[] larguraDeBanda;

    public static final int GEANT2 = 1;
    public static final int KYATERA = 2;

    public RedeControl(int nomeRede, int numeroConexoes) {
        this.redeView = new RedesView();
        this.rede = inicializarRede(nomeRede);
        this.linkControl = new LinkControl(this.rede);
    }

    private Redes inicializarRede(int rede) {
        if(rede == GEANT2) {
            this.larguraDeBanda = new int[] {1,3,6,48,192,768};
            return new Redes("data/geant2","Geant2", larguraDeBanda);
        }
    }
}
```

```

if(rede == KYATERA){
    this.larguraDeBanda = new int[] {24,48,192};
    return new Rede("data/kyatera","Kyatera", larguraDeBanda);
}
return null;
}

public RedeView getRedeView() {
    return redeView;
}

public void setRedeView(RedeView redeView) {
    this.redeView = redeView;
}

public Rede getRede() {
    return rede;
}

public void setRede(Rede rede) {
    this.rede = rede;
}

public LinkControl getLinkControl() {
    return linkControl;
}

public void setLinkControl(LinkControl linkControl) {
    this.linkControl = linkControl;
}

public int[] getLarguraDeBanda() {
    return larguraDeBanda;
}

```

```
public void setLarguraDeBanda(int[] larguraDeBanda) {  
    this.larguraDeBanda = larguraDeBanda;  
}  
  
}
```

Anexo 2.4 – Classe ConexaoControl

```
package br.edu.unifei.Control;

import edu.asu.emit.qyan.alg.model.Path;
import java.util.ArrayList;
import java.util.Random;
import br.edu.unifei.Model.Conexao;
import br.edu.unifei.Model.Redes;
import br.edu.unifei.View.ConexaoView;

/**
 *
 * @author Rafael Coutinho
 */
public class ConexaoControl {
    ArrayList<Conexao> conexoes;
    ArrayList<Path> listaDeCaminhos;
    int numeroConexoes;
    Random rand;
    ConexaoView conexaoView;
    int[] larguraDeBanda;
    Redes rede;

    public ConexaoControl(int numeroConexoes, int[] larguraDeBanda, Redes rede) {
        this.numeroConexoes = numeroConexoes;
        this.larguraDeBanda = larguraDeBanda;
        this.rede = rede;
        rand = new Random();
        conexoes = new ArrayList<Conexao>();
        conexaoView = new ConexaoView();
    }
}
```

```

    gerarConexoes();
}

private void gerarConexoes() {
    Conexao conexao;
    int fonte, destino, largura;
    int numConexoes = numeroConexoes;
    while(numConexoes > 0){
        do{
            fonte = rand.nextInt(rede.get_vertex_list().size());
            destino = rand.nextInt(rede.get_vertex_list().size());
        }while(fonte==destino);
        largura = larguraDeBanda[rand.nextInt(larguraDeBanda.length)];
        conexao = new Conexao(fonte, destino, largura);
        conexoes.add(conexao);
        numConexoes--;
    }
}

public int getNumeroConexoes() {
    return numeroConexoes;
}

public void setNumeroConexoes(int numeroConexoes) {
    this.numeroConexoes = numeroConexoes;
}

public ArrayList<Conexao> getConexoes() {
    return conexoes;
}

public void setConexoes(ArrayList<Conexao> conexoes) {
    this.conexoes = conexoes;
}

```

```
public ConexaoView getConexaoView() {  
    return conexaoView;  
}  
  
public void setConexaoView(ConexaoView conexaoView) {  
    this.conexaoView = conexaoView;  
}  
  
}
```

Anexo 2.5 – Classe AlgoritmoTodoVerdeControl

```
package br.edu.unifei.Control;

import br.edu.unifei.Interfaces.AlgoritmoControlInterface;
import br.edu.unifei.Model.Algoritmo;
import br.edu.unifei.Model.CaminhoOptico;
import br.edu.unifei.Model.Conexao;
import br.edu.unifei.Model.Link;
import br.edu.unifei.Model.Redes;
import br.edu.unifei.View.ConexaoView;
import edu.asu.emit.qyan.alg.control.YenTopKShortestPathsAlg;
import edu.asu.emit.qyan.alg.model.Path;
import edu.asu.emit.qyan.alg.model.abstracts.BaseVertex;
import java.util.ArrayList;
import java.util.List;

public class AlgoritmoTodoVerdeControl implements AlgoritmoControlInterface {
    Rede rede;
    YenTopKShortestPathsAlg yenAlg;
    Algoritmo algoritmo;
    ConexaoView conexaoView;
    MainControl mainControl;
    int quantidadeCaminhos;

    public AlgoritmoTodoVerdeControl(MainControl mainControl, Algoritmo
algoritmo) {
        this.mainControl = mainControl;
        this.rede = mainControl.getRedeControl().getRede();
        this.algoritmo = algoritmo;
        this.quantidadeCaminhos = getQuantidadeCaminhos();
    }
}
```

```

        conexaoView = new ConexaoView();
    }

    public void executar(){
        Conexao conexao;
        CaminhoOptico menorCaminho;
        List<Path> listaMenoresCaminhos;
        boolean bloqueio;

        conexaoView.mensagem("\nConexao do algoritmo " + algoritmo.getNome() + "
na rede " + rede.getNome());

        //varrendo as conexoes
        for(int i=0; i<mainControl.getConexaoControl().getConexoes().size();i++){
            conexao = mainControl.getConexaoControl().getConexoes().get(i);
            listaMenoresCaminhos = criarListaDeCaminhos(conexao); //fase 1
            conexaoView.inicioConexao(i+1);
            conexaoView.conexaoInfo(conexao.getFonte(),          conexao.getDestino(),
conexao.getLarguraDeBandaNecessaria());
            conexaoView.inicioFase1();
            conexaoView.fimFase1(listaMenoresCaminhos,
yenAlg.get_result_list().size());
            bloqueio          =          verificarProbabilidadeBloqueio(conexao,
mainControl.getRedeControl().getLinkControl(),listaMenoresCaminhos);
            if(!bloqueio){
                menorCaminho          =          calculaEnergiaPorCaminho(conexao,
mainControl.getRedeControl().getLinkControl()); //fase 2
                //verificando se não alcançou o limite total de gasto de energia
                if(mainControl.getLimiteDeEnergia()  >=  getGastoDeEnergia()  +
menorCaminho.getConsumoDeEnergia()) {
                    algoritmo.setGastoDeEnergia(algoritmo.getGastoDeEnergia()  +
menorCaminho.getConsumoDeEnergia());

```

```
mainControl.getRedeControl().getLinkControl().atualizaCapacidadeResidual(menorCaminho
,conexao.getLarguraDeBandaNecessaria());
```

```
mainControl.getRedeControl().getLinkControl().atualizarPesosDosLinks();
        //mainControl.getRedeControl().getLinkControl().mostrarLinks();
    }else{
        bloqueiaConexao("Limite de consumo de energia do sistema foi
alcançado");
    }
} else{
    bloqueiaConexao("Nenhum caminho com capacidade residual suficiente");
}
}
```

```
public int getNumeroBloqueios(){
    return algoritmo.getNumeroDeBloqueios();
}
```

```
public int getGastoDeEnergia(){
    return algoritmo.getGastoDeEnergia();
}
```

```
//fase 1
```

```
public List<Path> criarListaDeCaminhos(Conexao conexao){
    yenAlg = new YenTopKShortestPathsAlg(rede);
    List<Path> shortest_paths_list = yenAlg.get_shortest_paths(
        rede.get_vertex(conexao.getFonte()),
        rede.get_vertex(conexao.getDestino()),
        quantidadeCaminhos);
    return shortest_paths_list;
}
```

```

public boolean verificarProbabilidadeBloqueio(Conexao conexao, LinkControl
linkControl,List<Path> shortest_paths_list) {
    ArrayList<Path> listaBloqueios = new ArrayList<Path>();
    List<BaseVertex> vertices;
    Link link;
    boolean bloqueio =false;

    //monta uma lista dos caminhos sem capacidade residual para suportar a largura
de banda
    for(int i=0; i< shortest_paths_list.size(); i++){
        vertices = shortest_paths_list.get(i).get_vertices();
        for(int j=0; j< vertices.size()-1; j++){
            link = linkControl.getLink(vertices.get(j).get_id(),
vertices.get(j+1).get_id());

            if(link.getCapacidadeResidual()<conexao.getLarguraDeBandaNecessaria()){
                listaBloqueios.add(shortest_paths_list.get(i));
            }
        }
    }

    //varre a lista anterior, removendo os caminhos onde nao há capacidade residual
shortest_paths_list.removeAll(listaBloqueios);

    //verifica se restou algum caminho com capacidade residual suficiente
    if(shortest_paths_list.isEmpty()){
        return true;
    }
    return false;
}

//fase 2

```

```

public CaminhoOptico calculaEnergiaPorCaminho(Conexao conexao, LinkControl
linkControl) {
    conexaoView.inicioFase2();
    List<BaseVertex> vertices;
    Link link;
    CaminhoOptico caminho = null;
    CaminhoOptico caminhoMenor = null, primeiroCaminho = null;
    for(int i=0; i< yenAlg.get_result_list().size(); i++){
        caminho = new CaminhoOptico(yenAlg.get_result_list().get(i).get_vertices(),
yenAlg.get_result_list().get(i).get_weight());
        vertices = yenAlg.get_result_list().get(i).get_vertices();
        for(int j=0; j< vertices.size()-1; j++){
            link = linkControl.getLink(vertices.get(j).get_id(),
vertices.get(j+1).get_id());

caminho.setConsumoDeEnergia(caminho.getConsumoDeEnergia()+link.getConsumoDeEner
gia());
        }
        conexaoView.consumoEnergiaCaminho(i+1,
caminho.getConsumoDeEnergia());
        if(i==0) primeiroCaminho = caminhoMenor = caminho;
        else
if(caminho.getConsumoDeEnergia(<caminhoMenor.getConsumoDeEnergia())
caminhoMenor = caminho;
    }
    if(caminhoMenor != null){
        conexaoView.fimFase2(caminhoMenor.toString());
        return caminhoMenor;
    }
    return null;
}

private void bloqueiaConexao(String motivo) {
    algoritmo.addBloqueio();
}

```

```
        conexaoView.conexaoBloqueada(motivo);
    }

    private int getQuantidadeCaminhos() {
        int[] largura = rede.getLarguraDeBandaPermitida();
        int maiorLargura = largura[0];
        for(int i=1; i<largura.length;i++){
            if(largura[i] > maiorLargura) { maiorLargura = largura[i]; }
        }
        return (int)(Math.log(maiorLargura * 51.84));
    }

}
```

Anexo 3 – Principais Classes “View”

Anexo 3.1 – Classe RedeView

```
package br.edu.unifei.View;

import br.edu.unifei.Control.MainControl;
import br.edu.unifei.Model.Redes;
import java.util.ArrayList;

/**
 *
 * @author Rafael Coutinho
 */
public class RedeView {

    void resultadosFinais(ArrayList<MainControl> mainControlList) {
        double prob;
        System.out.println("\n-----Resultados Finais-----
-");
        System.out.println("Numero de conexoes: " +
mainControlList.get(0).getConexaoControl().getNumeroConexoes());

        for(int i=0; i<mainControlList.size(); i++){
            System.out.println("\nAlgoritmo " +
mainControlList.get(i).getAlgoritmoControl().getAlgoritmo().getNome() + " na rede " +
mainControlList.get(i).getRedeControl().getRede().getNome());
```

```

        System.out.println("Numero de bloqueios: " +
mainControlList.get(i).getAlgoritmoControl().getAlgoritmo().getNumeroDeBloqueios());
        prob = (
mainControlList.get(i).getAlgoritmoControl().getAlgoritmo().getNumeroDeBloqueios() /
        (double)
mainControlList.get(i).getConexaoControl().getNumeroConexoes()) * 100;
        System.out.println("Probabilidade de bloqueio: " + prob + "%");
        System.out.println("Gasto de energia: " +
mainControlList.get(i).getAlgoritmoControl().getAlgoritmo().getGastoDeEnergia() + "
Watts");
    }

}

}

```

Anexo 3.2 – Classe ConexaoView

```
package br.edu.unifei.View;

import edu.asu.emit.qyan.alg.model.Path;
import java.util.List;
import br.edu.unifei.Control.MainControl;

/**
 *
 * @author Rafael Coutinho
 */
public class ConexaoView {
    MainControl mainControl;

    public ConexaoView(){
    }

    public void inicioConexao(int i) {
        System.out.println("\n-----Conexao " + i + "-----
--");
    }

    public void conexaoInfo(int fonte, int destino, int larguraDeBandaNecessaria) {
        System.out.println("Conexao do vertice " + fonte + " ao vertice " + destino + "
com largura de banda " + larguraDeBandaNecessaria);
    }

    public void inicioFase1() {
        System.out.println("\nFase 1 - Caminhos encontrados:");
    }
}
```

```

public void fimFase1(List<Path> shortest_paths_list, int size) {
    System.out.println(":"+shortest_paths_list);
    System.out.println("Numero de solucoes encontradas: " + size);
}

public void conexaoBloqueada(String motivo) {
    System.out.println("Conexao Bloqueada! Motivo: " + motivo);
}

public void inicioFase2() {
    System.out.println("\nFase 2 - Escolhendo o melhor caminho:");
}

public void consumoEnergiaCaminho(int i, int consumoDeEnergia) {
    System.out.println("Consumo de energia do caminho " + i + ": " +
consumoDeEnergia);
}

public void fimFase2(String toString) {
    System.out.println("O caminho escolhido foi " + toString);
}

public void mensagem(String string) {
    System.out.println(string);
}

}

```

Anexo 3.3 – Classe Main

```
package br.edu.unifei.View;

import br.edu.unifei.Control.AlgoritmoControl;
import br.edu.unifei.Control.ConexaoControl;
import br.edu.unifei.Control.MainControl;
import br.edu.unifei.Control.RedControl;
import br.edu.unifei.Model.AlgoritmoGreenSpark;
import java.util.ArrayList;

/**
 *
 * @author Rafael Coutinho
 */
public class Main {

    public static void main(String[] args) {
        int limiteDeEnergia = 99999999;
        int numeroConexoes = 1000;
        ArrayList<MainControl> mainControlList = new ArrayList<MainControl>();

        RedeControl redeControl = new
RedeControl(RedeControl.GEANT2,numeroConexoes);
        ConexaoControl conexaoControl = new ConexaoControl(numeroConexoes,
redeControl.getLarguraDeBanda(), redeControl.getRede());

        redeControl = new RedeControl(RedeControl.GEANT2,numeroConexoes);

        AlgoritmoGreenSpark greenSpark;
        MainControl geant2_greenSpark = new MainControl(redeControl,
conexaoControl, AlgoritmoControl.GREEN_SPARK, limiteDeEnergia, numeroConexoes);
```

```

        greenSpark                =                (AlgoritmoGreenSpark)
geant2_greenSpark.getAlgoritmoControl().getAlgoritmo();
        greenSpark.setK(3);
        geant2_greenSpark.executar();
        mainControlList.add(geant2_greenSpark);

        redeControl = new RedeControl(RedeControl.GEANT2,numeroConexoes);

        MainControl    geant2_todoVerde    =    new    MainControl(redeControl,
conexaoControl, AlgoritmoControl.TODO_VERDE, limiteDeEnergia, numeroConexoes);
        geant2_todoVerde.executar();
        mainControlList.add(geant2_todoVerde);

        new RedeView().resultadosFinais(mainControlList);
    }

}

```