



**Universidade de Brasília**

Instituto de Ciências Exatas  
Departamento de Ciência da Computação

# WWSim: Interface Gráfica para o Simulador de Redes Ópticas WDMsim

Guilherme H. M. Fernandes

Monografia apresentada como requisito parcial  
para conclusão do Bacharelado em Engenharia da Computação

Orientador  
Prof. Dr. André Costa Drummond

Brasília  
2014

Universidade de Brasília — UnB  
Instituto de Ciências Exatas  
Departamento de Ciência da Computação  
Bacharelado em Engenharia da Computação

Coordenador: Prof. Dr. Ricardo Zelenovsky

Banca examinadora composta por:

Prof. Dr. André Costa Drummond (Orientador) — CIC/UnB

Prof. Msc. João José da Costa Gondim — CIC/UnB

Prof. Msc. Marcos Fagundes Caetano — CIC/UnB

### **CIP — Catalogação Internacional na Publicação**

Fernandes, Guilherme H. M..

WWSim: Interface Gráfica para o Simulador de Redes Ópticas WDM-Sim / Guilherme H. M. Fernandes. Brasília : UnB, 2014.

49 p. : il. ; 29,5 cm.

Monografia (Graduação) — Universidade de Brasília, Brasília, 2014.

1. Java, 2. Redes Ópticas, 3. WDM, 4. RWA, 5. Simulador

CDU 004.4

Endereço: Universidade de Brasília  
Campus Universitário Darcy Ribeiro — Asa Norte  
CEP 70910-900  
Brasília-DF — Brasil



# Resumo

A construção de um sistema de representação e modelagem de redes de transmissão de dados apresenta um desafio de implementação e de compreensão de usuário. A representação fiel de tal rede pode ser um empecilho até para os mais especializados, e poucas formas de representação foram concebidas para o projeto de redes ópticas. Esta monografia apresenta os resultados de projeto de implementação de uma interface gráfica para o simulador de redes ópticas WDMsim que permite manipulação direta de usuário que apresente fidelidade com a infraestrutura real, facilidade de interação com o usuário, capacidade de alteração do algoritmo de alocação de comprimento de onda e que permita uma visão didática dos conceitos de redes ópticas.

**Palavras-chave:** Java, Redes Ópticas, WDM, RWA, Simulador

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Motivação . . . . .	2
1.2	Objetivos . . . . .	2
1.2.1	Objetivo Geral . . . . .	2
1.2.2	Objetivos Específicos . . . . .	2
1.3	Organização da Monografia . . . . .	2
<b>2</b>	<b>Fundamentos em Redes Ópticas WDM</b>	<b>4</b>
2.1	Contextualização das Redes WDM . . . . .	4
2.2	Multiplexação por Divisão de Comprimento de Onda . . . . .	5
2.3	Caminho Óptico . . . . .	5
2.4	Topologia Física e Virtual . . . . .	6
2.5	Agregação de Tráfego . . . . .	7
2.6	Conversão de Comprimento de Onda . . . . .	8
2.7	Roteamento e Alocação de Comprimentos de Onda . . . . .	8
2.8	Tráfego em Redes Ópticas . . . . .	9
2.9	Probabilidade de Bloqueio e Taxa de Bloqueio de Banda . . . . .	9
2.10	Simuladores de Redes . . . . .	10
<b>3</b>	<b>Simulador WDMSim</b>	<b>11</b>
3.1	Algoritmo RWA . . . . .	11
3.2	Parâmetro de Simulação XML . . . . .	12
3.3	Resultado da Simulação . . . . .	13
<b>4</b>	<b>Funcionalidades do WWSim</b>	<b>15</b>
4.1	Interface Gráfica Web . . . . .	15
4.2	Servidor de Simulação . . . . .	15
4.3	Requisitos . . . . .	16
4.3.1	Manipulação de Arquivos XML . . . . .	16
4.3.2	Ferramentas de Edição . . . . .	18

4.3.3	Cenário de Tráfego . . . . .	20
4.3.4	Ferramentas de Simulação . . . . .	21
4.3.5	Gerar Estatísticas . . . . .	22
4.4	Requisitos Desejáveis . . . . .	23
4.4.1	Opções de Edição . . . . .	23
4.4.2	Salvar e Recuperar Simulação . . . . .	23
4.4.3	Acompanhamento de Simulação . . . . .	24
4.4.4	Manipulação Dinâmica de Eventos . . . . .	24
<b>5</b>	<b>Projeto e Arquitetura do WWSim</b>	<b>25</b>
5.1	Pacotes da Aplicação . . . . .	26
5.2	Padrões de Projeto . . . . .	27
5.3	Implementação . . . . .	28
5.4	Diagramas de Classe . . . . .	30
<b>6</b>	<b>Conclusão</b>	<b>32</b>
6.1	Contribuições . . . . .	32
6.2	Trabalhos Futuros . . . . .	32
	<b>Referências</b>	<b>34</b>
<b>A</b>	<b>Manual de Usuário</b>	<b>35</b>
A.1	Tela Inicial . . . . .	35
A.2	Criação da Topologia . . . . .	35
A.3	Adição de Chamadas . . . . .	37
A.4	Execução da Simulação . . . . .	38
A.5	Criação de uma Classe RWA . . . . .	38

# Lista de Figuras

2.1	Optical Cross-Connect . . . . .	5
2.2	Multiplexação por Divisão de Comprimento de Onda . . . . .	6
2.3	Caminhos Ópticos . . . . .	7
2.4	Conversores de Comprimento de onda . . . . .	8
3.1	Interface RWA . . . . .	12
3.2	Diagrama do WDMSim . . . . .	13
3.3	Exemplo de arquivo XML de Entrada . . . . .	13
3.4	Arquivo de Trace . . . . .	14
4.1	Arquitetura Web . . . . .	16
4.2	Manipular Arquivo . . . . .	17
4.3	Ferramentas de Edição . . . . .	18
4.4	Cenário de Tráfego . . . . .	20
4.5	Ferramentas de Simulação . . . . .	21
4.6	Acompanhamento de Simulação . . . . .	24
5.1	Diagrama de Pacotes . . . . .	26
5.2	Arquitetura da Aplicação . . . . .	26
5.3	Diagrama de BD . . . . .	27
5.4	Classes de Criação da Tela . . . . .	28
5.5	Diagrama de Classes Functions . . . . .	29
5.6	Diagrama de Classes de Cliente . . . . .	30
5.7	Diagrama de Classes do Servidor . . . . .	31
A.1	Tela Principal . . . . .	35
A.2	Criação de Roteador . . . . .	36
A.3	Adicionar um novo link entre roteadores . . . . .	36
A.4	Remover um roteador já existente . . . . .	36
A.5	Editar roteador já criado . . . . .	37
A.6	Importar e Exportar em XML . . . . .	37

A.7 Rede Simples Modelada . . . . .	38
A.8 Arquivo XML Exemplo . . . . .	38
A.9 Manter Chamadas . . . . .	39
A.10 Executar Simulação . . . . .	39
A.11 RWA Personalizado . . . . .	40



# Lista de Tabelas

3.1	Elementos XML do WDMSim . . . . .	14
4.1	Campos de configuração de Roteadores . . . . .	19
4.2	Campos de configuração de Enlace . . . . .	20
4.3	Campos de configuração de Chamada . . . . .	20
4.4	Campos presentes ao ser requisitar uma simulação . . . . .	21
4.5	Eventos registrados no arquivo de trace . . . . .	22
4.6	Atalhos do Sistema . . . . .	24
5.1	Serviços REST Disponíveis . . . . .	27
5.2	Servlets Disponíveis para manipulação de arquivos . . . . .	28

# Lista de Abreviaturas

BBR	Bandwidth Blocking Rate
BP	Blocking Probability
COS	Class Of Service
CSS	Cascading Style Sheets
ECSA	Exchange Carriers Standards Association
FDM	Frequency Division Multiplexing
FF	First Fit
FTTH	Fiber To The Home
GWT	Google Web Toolkit
HTML	HyperText Markup Language
JSON	JavaScript Object Notation
LED	Light Emitting Diode
LU	Least Used
MBBR	Mean Bandwidth Blocking Rate
MBP	Mean Blocking Probability
MU	Most Used
MVP	Model-View-Presenter
Ns-2	Network Simulator 2
OPNET	Optimized Network Engineering Tools

OTCL Object oriented Tool Command Language

OXC Optical Cross-Connect

REST Representational State Transfer

RPC Remote Procedure Call

RWA Routing and Wavelength Assignment

SONET Synchronous Optical Network

TCP/IP Transfer Control Protocol / Internet Protocol

TMD Tempo Médio de Duração

WDM Wavelength Division Multiplexing

WLCR Weighted Least-Congestion Routing

WRS Wavelength-Routing Switch

XML eXtensible Markup Language

# Capítulo 1

## Introdução

As redes de comunicação de dados digitais apresentam um importante papel para a sociedade atualmente, permitindo um grande fluxo de informações. Elas estão presentes cada vez mais no dia a dia das pessoas e suas aplicações incluem telefonia, rádio, televisão, a Internet, entre outras.

A Internet tem destaque dentre as redes devido às suas inúmeras possibilidades de serviços. Diversos dispositivos com capacidades de conexão se ligam à rede sob os mesmos padrões de comunicação. Redes globais, nacionais, locais, corporativas e redes móveis se aglomeram e constroem a rede como um todo. Dentro desse conjunto de tecnologias responsáveis pela transmissão de dados se destaca a tecnologia óptica.[9]

A tecnologia óptica é notória pela sua grande capacidade de transmissão. Em 2002, uma rede óptica intercontinental de 250 mil quilômetros foi criada com a capacidade de 2,56 Tb/s e desde então diversos laudos apontam que a conectividade aumentou drasticamente a partir de 2004. O recorde atual de transmissão é de 1 petabit (1000 terabit) por segundo sobre um fibra de 52,4 Km com 12 núcleos[11]. Novas tecnologias de FTTH(*Fiber To The Home*) podem prover acima de 100 Mb/s de conexão de internet entre o provedor e uma rede doméstica local [4].

Informações sobre a evolução da Internet relatam que, de 1995 a 2011, o número de usuários da Internet saltou de 39,14 milhões para 2,27 bilhões. Grande parte dessa evolução é devido ao desenvolvimento das tecnologias de fibra óptica.[14]

Para prover formas de se avaliar e melhorar o projeto de redes ópticas, é preciso ferramentas que auxiliem os cálculos de desempenho e permitam elaborar as melhores estratégias para a implementação da estrutura da rede.

A avaliação de desempenho de sistemas pode ser realizada através de três métodos: analítico, medição e simulação. Devido à complexidade dos ambientes de redes de computadores, em geral, não é possível modelá-los analiticamente de maneira precisa. Além disso, a implementação de ambientes reais de testes (*testbeds*) muitas vezes é inviável

por questões de custo e de disponibilidade. Devido a isso,, o método de simulação é amplamente utilizado pela comunidade científica de redes de computadores para avaliar as soluções propostas.

Esta monografia apresenta a implementação de uma interface gráfica para o simulador de redes ópticas com multiplexação por divisão de comprimento de onda, o WDMsim [5].

## 1.1 Motivação

Este projeto do WWSim tem como motivação a construção e descrição de uma interface gráfica web para o simulador de redes ópticas WDMsim. A facilidade de interação com o usuário que modela e avalia a rede é tida como essencial pois permite que o simulador seja usado com propósito didático. O simulador pode ser utilizado para ilustrar os conceitos de redes ópticas WDM tais como caminho óptico, agregação de tráfego e roteamento e alocação de comprimento de onda.

## 1.2 Objetivos

### 1.2.1 Objetivo Geral

Proporcionar a implementação e validação de uma interface gráfica Web capaz de gerar e modelar redes ópticas WDM.

### 1.2.2 Objetivos Específicos

- Automatizar a geração de arquivo de parâmetro de entrada para o simulador WDM-Sim.
- Permitir a execução remota de simulação de rede modelada em interface gráfica.
- Permitir o acesso a arquivos de resultados como estatísticas e arquivo de trace com eventos da simulação.

## 1.3 Organização da Monografia

A organização do restante desta monografia está a seguir.

O Capítulo 2 apresenta alguns conceitos mais relevantes nos quais o simulador de redes se baseia. O capítulo inclui explicações sobre o funcionamento das redes ópticas e também define o conceito de RWA (*Routing Wavelength Assignment*) que é responsável

pela atribuição de comprimentos de onda e de rotas aos fluxos recebidos por um dispositivo OXC(Optical Cross-Connect).

O Capítulo 3 mostra o funcionamento e os princípios que guiam o WDMSim, projeto de simulador desenvolvido em Java que busca simular as redes ópticas utilizando eventos discretos. O capítulo mostra os parâmetros e arquivos de entrada e de saída gerados pelo simulador.

O Capítulo 4 apresenta os requisitos do sistema funcionalidades do sistema desenvolvido e incluem os casos de uso pensados para os sistema. Os casos de uso descritos visam descrever a melhor experiência do usuário em relação ao uso do programa.

O Capítulo 5 mostra a arquitetura pensada para o programa e o projeto de desenvolvimento seguido. O capítulo apresenta decisões tomadas durante a implementação do projeto com o objetivo de melhorar fatores de qualidade como manutenibilidade e interoperabilidade.

O Capítulo 6 mostra o que se pôde extrair deste trabalho e propostas de trabalhos futuros. No capítulo, estão descritos possíveis expansões do simulador e possibilidades de uso.

# Capítulo 2

## Fundamentos em Redes Ópticas WDM

Este capítulo apresenta os fundamentos em que se baseou a criação e implementação do simulador WWSim além de uma contextualização sobre o uso e difusão das redes ópticas.

### 2.1 Contextualização das Redes WDM

No início da utilização da Internet, certas empresas de comunicação que mantinham redes de telecomunicação as utilizavam para suportar o serviço de telefonia em paralelo ao serviço de Internet. A infraestrutura utilizada, o cobre, era suscetível a interferências elétricas e apresentava grande limitação em termos de banda.

Dadas as limitações do cobre, o teste de redes ópticas se difundiu e um padrão foi criado: o SONET (*Synchronous Optical Network*). Criado pela ECSA (*Exchange Carrier Standards Association*) em conjunto com a *Bellcore*, o padrão define o protocolo para a transferência de fluxos de bit sobre um canal óptico com o uso de lasers ou diodos emissores de luz (LEDs).

Tecnologias de Multiplexação por comprimento de onda foram criadas para permitir que haja transmissão simultânea com o uso de diferentes faixas do espectro. Essa tecnologia permite que um seletor de remodulação receba um sinal WDM de entrada, selecione um canal óptico específico e posicione os dados do canal em um novo canal de saída gerado[2]. Isso expande imensamente as possibilidades de transmissão de um canal óptico.

Houve a criação de novos comutadores para comprimentos de onda, os OXC (*Optical Cross-connect*), capazes de interligar fibras ópticas de maneira transparente, isso permitiu o surgimento de rede de salto único (*Single Hop*), cujo fluxo pode atravessar toda a rede sem ser necessária a comutação eletrônica entre nós intermediários. Fibras ópticas e multiplexação por comprimento de onda se mostraram tecnologias promissoras que

possuem expectativa de satisfazer a necessidade de banda passante requisitada atualmente pela Internet.

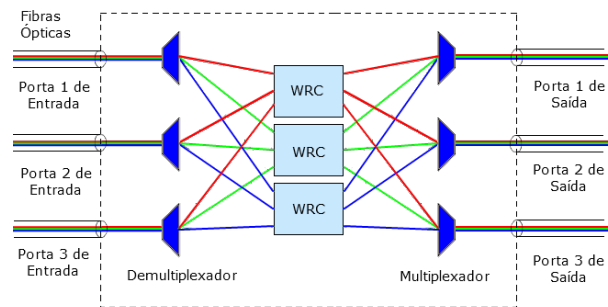


Figura 2.1: Optical Cross-Connect (OXC) com três comprimentos de onda por fibra [6]

## 2.2 Multiplexação por Divisão de Comprimento de Onda

Em tecnologia óptica, Multiplexação por Divisão de Comprimento de Onda (do Inglês *Wavelength Division-Multiplexing*) é a tecnologia que agrupa um número de sinais de portadores ópticos em uma única fibra óptica com o uso de diferentes comprimentos de onda de feixes de laser. A técnica permite comunicação bidirecional sobre um cabo de uma fibra e também multiplicação da capacidade. O termo WDM é aplicado a um portador óptico, ou comprimento de onda. O termo FDM (*Frequency Division Multiplexing*) não é aplicado por remeter a um portador de rádio. Ambos os termos se relacionam devido à relação em que o comprimento de onda multiplicado pela frequência é igual à velocidade da luz.

A tecnologia WDM utiliza um multiplexador no agente transmissor para agrupar os sinais na fibra e, no agente receptor, utiliza um demultiplexador para dividir as diferentes portadoras. A figura 2.2 ilustra os conceitos de multiplexador e demultiplexador utilizados para realizar a divisão de canais em faixas de comprimentos de onda. Tal conceito foi publicado em 1978 e utilizava 2 sinais combinados. Sistemas mais recentes consegue lidar com até 160 sinais[3].

## 2.3 Caminho Óptico

A transparência das redes ópticas traz à tona o problema de roteamento e alocação de comprimentos de onda já que, para que haja a conexão, é preciso a atribuição de uma faixa dedicada de espectro, o chamado caminho óptico (do Inglês *lightpath*). Os caminhos ópticos são fluxos que utilizam um comprimento de onda atribuído permitindo que o



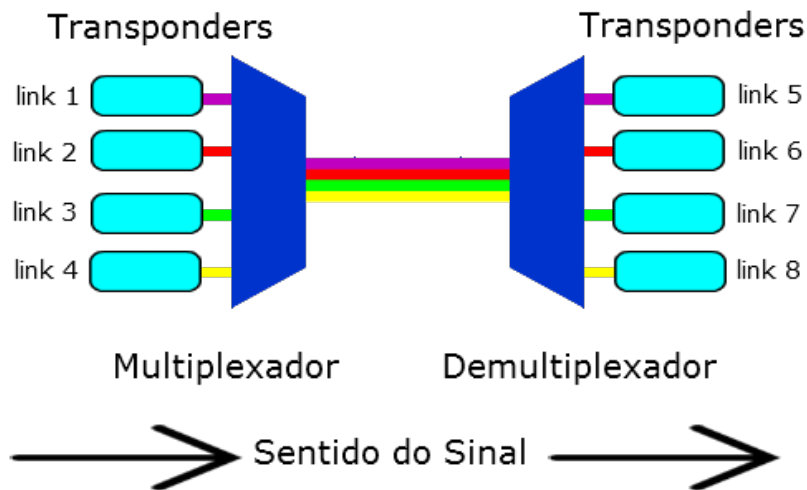


Figura 2.2: Multiplexação por Divisão de Comprimento de Onda

tráfego de dados flua de forma transparente. Um caminho óptico trata-se do conjunto da rota (sequência de enlaces de fibra) e o comprimento de onda alocado. [15]

Caso o respectivo comprimento esteja ocupado, o fluxo pode ser submetido a uma conversão óptico-elétrica que atribui outro comprimento de onda disponível ou pode ser submetido a agregação de tráfego. Tal conversão representa um custo maior em termos de processamento e de tempo e, por isso, deve ser evitado para maximizar a vazão da rede.

A figura 2.3 ilustra a atribuição de caminhos ópticos. Dado que cada caminho óptico recebe um faixa de onda onde pode operar (representado pelas diferentes cores), a atribuição de determinada faixa a um caminho óptico irá depender da topologia de rede e/ou do fluxo de dados que trafegam por ela.

## 2.4 Topologia Física e Virtual

Em uma rede WDM roteada por comprimento de onda, um caminho óptico pode ser estabelecido do roteador fonte até o destino e pode se estender por vários enlaces sem a necessidade de sair do meio óptico e portanto sobrepassa os nós intermediários. O uso

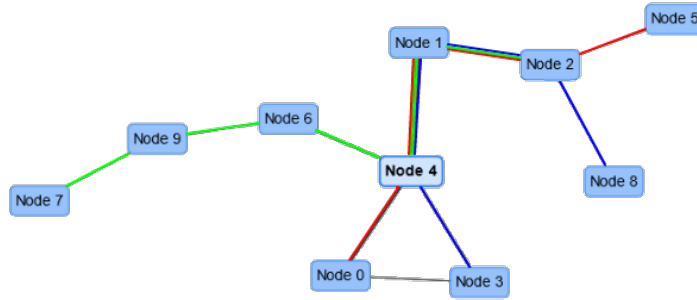


Figura 2.3: Ilustração análoga à atribuição de caminhos ópticos

de comutadores ópticos WRSs (do Inglês : *Wavelength-Routing Switches*) em roteadores intermediários permite criar roteadores logicamente “vizinhos” ou virtuais a partir de uma topologia física onde tais roteadores estão geograficamente distantes[15].

O conjunto de tais enlaces virtuais é chamado de topologia virtual e depende estritamente de um algoritmo de roteamento e alocação de comprimento de onda (RWA, discutido na seção 2.7) para serem estabelecidos de forma eficiente que otimize o uso dos recursos de rede.

O conjunto de enlaces onde os roteadores estão fisicamente conectados é chamado de topologia física, não depende do algoritmo RWA e pode ser representada como um grafo ponderado.

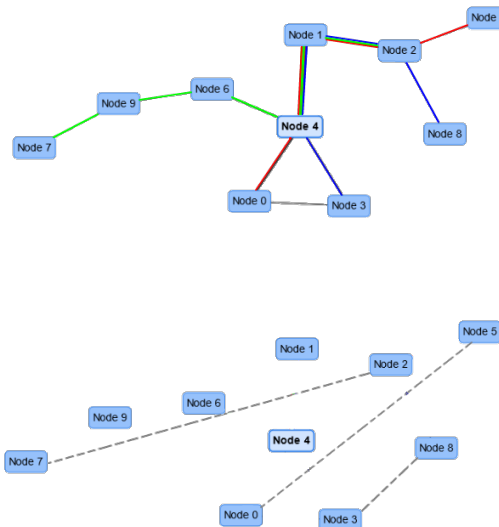


Figura 2.4: O uso de caminhos ópticos permite a criação de enlaces virtuais

## 2.5 Agregação de Tráfego

O uso dedicado de um comprimento de onda para determinado fluxo representa um grande desperdício do potencial uso do comprimento de onda, pois parte dos tráfegos são de baixa velocidade (*Subwavelength Traffic*).

Para a atenuação desse problema, utiliza-se a agregação de tráfego, que propõe que os fluxos sejam agregados em comprimentos de onda para que haja poucas despesas adicionais relacionadas à conversão opto-elétrica. Os caminhos ópticos são reutilizados para encaminhar os tráfegos de baixa velocidade sem necessidade de criar um novo caminho óptico.

## 2.6 Conversão de Comprimento de Onda

As restrições de continuidade de comprimento de onda podem ser atenuadas se os equipamentos contiverem conversores de comprimento de onda. Os conversores transferem um sinal de entrada para um comprimento diferente de saída. A figura 2.4 mostra os tipos de conversões de comprimento de onda que um OXC pode realizar.

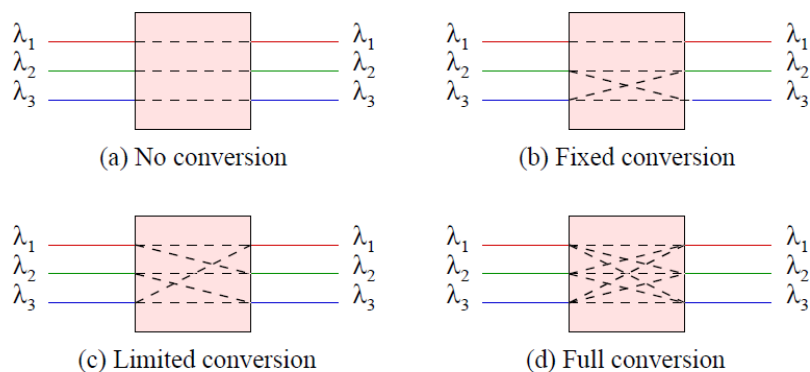


Figura 2.5: Tipos de conversão de Comprimento de Onda[6]

## 2.7 Roteamento e Alocação de Comprimentos de Onda

A decisão de qual caminho óptico seguir cabe ao algoritmo de roteamento e alocação de comprimento de onda (do Inglês: *Routing Wavelength Assignment* ou RWA). O algoritmo RWA pode levar em consideração informações da topologia da rede, da disponibilidade ou capacidade de determinado canal.

Vários algoritmos já foram definidos e descritos para o problema do roteamento e alocação de comprimento de onda.

Para roteamento estático, o algoritmo de Dijkstra apresenta bons resultados e levam em consideração a estrutura de grafo ponderado da rede e determina o menor caminho de forma otimizada [13].

Para roteamento dinâmico, o algoritmo WLCR (*Weighted Least-Congestion Routing*) é uma alternativa pois leva em consideração o estado de congestionamento de determinado link para tomar decisões quanto ao caminho. [12]

Para alocação de comprimento de onda, destacam-se os algoritmos *First-Fit*(FF), *Least-Used*(LU) e o *Most-Used*(MU). O algoritmo *First-Fit* busca o primeiro comprimento de onda disponível seguindo a uma ordem sequencial. O algoritmo *Least-Used* busca o comprimento de onda menos usado para balancear o uso comprimentos disponíveis. E o algoritmo *Most-Used* reusa os comprimentos mais utilizados para restringir ao menor número de canais.[12]

## 2.8 Tráfego em Redes Ópticas

Em redes ópticas WDM com comutação por circuito, caminhos ópticos são estabelecidos e terminados na medida da necessidade da alocação de recursos para o atendimento das demandas das redes clientes. Nesse contexto, uma unidade de demanda é representada por uma requisição de conexão ou chamada. Comumente, uma chamada inclui a definição de um nó de origem, um nó de destino e a quantidade de banda passante requisitada. É responsabilidade do plano de controle da rede alocar recursos (caminhos ópticos) para o atendimento de uma chamada.

Em simuladores de redes ópticas as demandas por conexão são geralmente simuladas utilizando o mesmo modelo aplicado em redes telefônicas, ou seja, utilizando um processo estocástico chamado Poisson. O processo de Poisson define uma distribuição de chegadas de requisição na qual o tempo médio de duração das chamadas é normalmente modelado por uma distribuição exponencial negativa.

A taxa média de chegada de chamadas é definida a partir da carga imposta à rede, medida em Erlang. A carga pode ser definida como  $E = T \times D$ , onde  $E$  é a carga (Erlang),  $T$  é a taxa média de chegada e  $D$  é o tempo médio de duração das chamadas. Para cada simulação, deve-se definir as variáveis  $E$  e  $D$ , de forma que a taxa de chegada possa ser calculada.

Dessa forma, pode-se realizar a simulação de tráfego em redes ópticas variando-se os cenários de carga e também, a quantidade de banda passante requisitada. Tipicamente, em uma simulação, define-se um conjunto de granularidades de banda passante que caracterizem um determinado cenário de tráfego e são executadas várias simulações de forma a obter-se um conjunto de resultados referentes a um intervalo de carga de interesse.

## 2.9 Probabilidade de Bloqueio e Taxa de Bloqueio de Banda

Uma das métricas relevantes para se avaliar um rede óptica é a probabilidade de bloqueio (do Inglês: *Blocking Probability*). A medida é definida como a razão entre o número de chamadas aceitas e o número total de chamadas. Para se atingir uma baixa razão de probabilidade de bloqueio é preciso solucionar conflitos e complicações relacionados à agregação de tráfego discutidos anteriormente.

Outra métrica relevante para avaliação de redes ópticas é a Taxa de Bloqueio de Banda (do Inglês: *Bandwidth Blocking Rate*). A medida é definida como a razão de tráfego bloqueado sobre o total de banda passante requisitada durante determinado tempo de uma simulação. O resultado da métrica irá depender do algoritmo de solução do problema de agregação dinâmica de tráfego.

## 2.10 Simuladores de Redes

O estudo e avaliação de sistemas em redes de comunicação é viabilizado em parte pelo uso de simuladores. Para simuladores de tráfego, pode-se citar o NS2(*Network Simulator 2*) e o OPNET(*Optimized Network Engineering Tools*).

NS2 é um simulador de eventos discretos com foco em pesquisa de rede. NS2 apresenta suporte para simulações em TCP, roteamento, protocolos *multicast* sobre rede com e sem fio[1]. Os parâmetros de entrada do NS2 exigem do usuário tenha domínio da linguagem interpretada OTCL (*Object oriented Tool Command Language*) para a correta descrição da simulação.

A proposta do OPNET, diferentemente, é oferecer um ambiente de simulação de redes com modelagem visual para permitir ser utilizado para planejamento de capacidade e também para gerenciamento de desempenho de aplicação. A sua licença paga permite melhorar o desempenho de tecnologias comerciais.

# Capítulo 3

## Simulador WDMSim

Em 2008, um simulador de redes ópticas foi implementado com o objetivo de avaliar os algoritmos de agregação de tráfego citados no capítulo 2, o chamado WDMSim. O simulador foi uma das contribuições da tese de doutorado “Agregação de Tráfego em Redes Ópticas com Multiplexação por Comprimento de Onda”[7]. O WDMSim, conforme citado pelo autor, contribuiu com a avaliação crítica dos algoritmos de agregação dinâmica de tráfego em redes ópticas WDM propostos na tese.

Este capítulo descreve a estrutura desse simulador. Cabe ressaltar que os elementos descritos neste capítulo não foram contribuições da monografia mas sim de estudos anteriores descritos em maior detalhe na tese “Agregação de Tráfego em Redes Ópticas com Multiplexação por Comprimento de Onda”[7].

Pensado e desenvolvido em Java, o simulador de redes ópticas WDMSim utiliza eventos discretos para simular os eventos de uma rede óptica WDM. Suas capacidades incluem simular chegadas e partidas de requisições em uma rede óptica WDM com capacidade de se definir o algoritmo de agregação de tráfego conforme descrito na seção 2.5 e 2.7.

Como já reforçado na seção 2.5, o aprimoramento na alocação de recursos de uma rede óptica é realizado primordialmente através da escolha do algoritmo de RWA. Por essa razão, é fundamental dar ao usuário a opção de selecionar ou indicar qual o algoritmo que será usado durante a simulação. Assim sendo, o WDMSim foi concebido para receber como parâmetro de simulação um módulo RWA, que contém uma implementação de algoritmo RWA a ser usado na simulação.

### 3.1 Algoritmo RWA

O simulador foi pensado para receber um parâmetro definido pelo usuário que irá conter a definição do algoritmo de agregação, o Módulo RWA. Tal parâmetro é passado como uma classe Java que implementa a interface RWA. A figura 3.1 apresenta a assinatura

dos métodos que o módulo RWA deve implementar para realizar a efetiva agregação do tráfego.

```
public interface RWA {  
    public void flowArrival(Flow flow);  
    public void flowDeparture(long id);  
    public void simulationInterface(PhysicalTopology pt, VirtualTopology vt, ControlPlaneForRWA cp);  
}
```

Figura 3.1: Métodos definidos pela interface RWA

Tal interface, pensada para ser simples ao usuário, contém os métodos que o simulador irá invocar para tomar decisões quanto à aceitação e ao bloqueio das chamadas geradas. A interface lida com a chegada e partida de fluxos.[5, 7]

O método *simulationInterface* é o método de inicialização da classe RWA. Conforme mostrado, o método passa à classe a topologia física e virtual da rede simulada, e um objeto com os comandos de controle do fluxo de dados.

O método *flowArrival* é chamado quando há a chegada de um novo fluxo de dados. O parâmetro Flow é o objeto que contém as informações do fluxo de dados que vai de um nó fonte até um nó de destino.

O método *flowDeparture* é chamado quando há a partida de um fluxo de dados. O parâmetro id é o identificador do fluxo que irá partir.

## 3.2 Parâmetro de Simulação XML

O simulador foi primeiramente proposto para ser executado em Java por meio de linha de comando. Devido a isso, percebe-se certa dificuldade em relação à execução e à passagem dos corretos parâmetros ao simulador.

Dentre os parâmetros, o principal é um arquivo de configuração em formato XML. Tal arquivo deve conter as especificações de topologia física da rede óptica a ser simulada pelo programa. A figura 3.2 indica os parâmetros passados e os resultados gerados pelo simulador.

A tabela 3.1 apresenta os elementos e tags XML usados para representar os componentes de uma simulação a ser passada para o WDMSim.

A figura 3.3 apresenta um exemplo de arquivo de parâmetros da topologia da simulação assim como as informações das chamadas e o nome do módulo RWA.

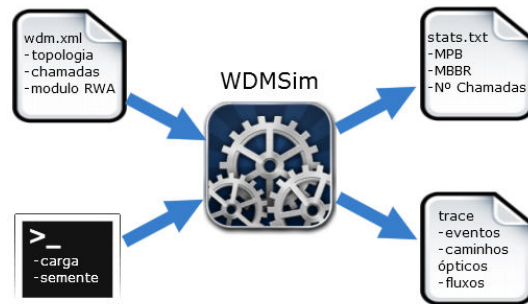


Figura 3.2: Diagrama do fluxo de dados do simulador WDMSim

```

<wdmsim version="0.1">
  <rwa module="MyRWA"/>
  <traffic calls="10000" load="100" max-rate="192">
    <calls holding-time="1" rate="1" cos="1" weight="1"/>
    <calls holding-time="1" rate="3" cos="1" weight="1"/>
    <calls holding-time="1" rate="12" cos="1" weight="1"/>
  </traffic>
  <physical-topology name="Network" wavelengths="8">
    <nodes>
      <node id="0" grooming-in-ports="16" grooming-out-ports="16"
        wlconverters="4" wlconversion-range="2"/>
      <node id="1" grooming-in-ports="16" grooming-out-ports="16"
        wlconverters="4" wlconversion-range="2"/>
      <node id="2" grooming-in-ports="16" grooming-out-ports="16"
        wlconverters="4" wlconversion-range="2"/>
    </nodes>
    <links>
      <link source="0" destination="1" delay="2.5" bandwidth="192" weight="500"/>
      <link source="1" destination="0" delay="2.5" bandwidth="192" weight="500"/>
      <link source="0" destination="2" delay="3.75" bandwidth="192" weight="750"/>
      <link source="2" destination="0" delay="3.75" bandwidth="192" weight="750"/>
      <link source="1" destination="2" delay="5" bandwidth="192" weight="1000"/>
      <link source="2" destination="1" delay="5" bandwidth="192" weight="1000"/>
    </links>
  </physical-topology>
</wdmsim>

```

Figura 3.3: Exemplo de arquivo XML de Entrada

### 3.3 Resultado da Simulação

Como resultado, o programa imprime em tela as principais métricas de rede coletadas ao longo da simulação. Dentre essas métricas, se encontram a probabilidade média de bloqueio (do inglês *Mean Blocking Probability* ou MBP), a média de bloqueio de banda passante (do inglês *Mean Blocking Bandwidth Rate* ou MBBR) e também o número de chamadas criadas entre os nós da topologia.

O programa é capaz também de gerar um arquivo de trace que contém os eventos gerados pelo simulador para coletar os parâmetros de rede. A figura 3.4 contém um exemplo de arquivo de trace gerado pelo WDMSim. No arquivo, é possível visualizar os eventos relacionados à criação, chegada, partida, e ao bloqueio de fluxos de dados.



Elemento	Atributos	Descrição
wdmsim	version	Descreve os componentes de uma simulação
rwa	module	Contém o nome do módulo RWA a ser usado na simulação
traffic	calls, load, max-rate	Contém as informações do tráfego
calls	holding-time, rate, cos, weight	Caracteriza uma chamada a ser simulada
physical-topology	name, wavelengths	Descreve a topologia física da rede
nodes	-	Contém os roteadores da topologia física
node	id, grooming-in-ports, grooming-out-ports, wlconverters, wlconversion-range	Elemento que caracteriza a entidade roteador óptico
links	-	Contém os enlaces da topologia de rede
link	source, destination, delay, bandwidth, weight	Representa um enlace da rede
virtual-topology	-	Contém as informações da topologia virtual

Tabela 3.1: Elementos presentes em um arquivo XML de configuração

Também é possível visualizar os eventos de criação e remoção de caminhos ópticos.

```

flow-accepted - 7 4 13 10000 0 1 7
flow-arrived 0.046460320449569575 8 3 9 5000 0 1
lightpath-created 8 3 9 14_0
flow-departed 0.05027616411721965 4 - - - -
lightpath-removed 4 2 6 10_0 17_0 18_0

```

Figura 3.4: Exemplo de arquivo de Trace de Eventos

# Capítulo 4

## Funcionalidades do WWSim

Este capítulo apresenta as principais funcionalidades e casos de uso concebidos para a implementação da interface gráfica. O capítulo apresenta também certos requisitos de arquitetura do sistema implementado.

O WWSim deve englobar todas as funcionalidades já implementadas pelo WDMSim e dar continuação ao já implementado e descrito no capítulo 3. A interface RWA, já idealizada e definida em trabalhos anteriores, foi utilizada na implementação do WWSim para estender a funcionalidade do WDMSim de permitir ao usuário implementar o próprio algoritmo RWA em Java, conforme descrito na seção 3.1.

### 4.1 Interface Gráfica Web

Uma interface para um simulador de redes representa um desafio em complexidade de representação gráfica e de funcionalidades de simulação. Essas características devem se integrar para proporcionar uma melhor experiência ao usuário da aplicação.

A interface foi escolhida para ser executada em *browser*. Essa escolha é tomada devido à universalidade de plataformas que o HTML (*HiperText Markup Language*), Javascript e o CSS (*Cascading Style Sheets*) proporcionam dado que são padrões já consolidados na criação de páginas web. A interface funcionará como um tradutor da rede modelada pelo usuário e o arquivo XML necessário para o funcionamento da simulação conforme descrito na seção 3.2.

### 4.2 Servidor de Simulação

O servidor da aplicação deve ser capaz de receber os dados passados pela interface gráfica. O uso de serviços web é uma escolha viável para esse tipo de aplicação, pois

permite à interface gráfica lidar somente com aspectos de visualização e deixar a execução da simulação e recebimento dos parâmetros ao servidor da aplicação. A figura 4.1 esquematiza a implementação da arquitetura da aplicação.

O servidor de simulação deve apresentar mecanismos de controle de concorrência pois o sistema deve ser multiusuário.

O servidor deve também utilizar a implementação do WDMSim desenvolvida como contribuição da tese descrita no capítulo 3 sem realizar grandes alterações no código implementado, algo similar a uma caixa-preta.

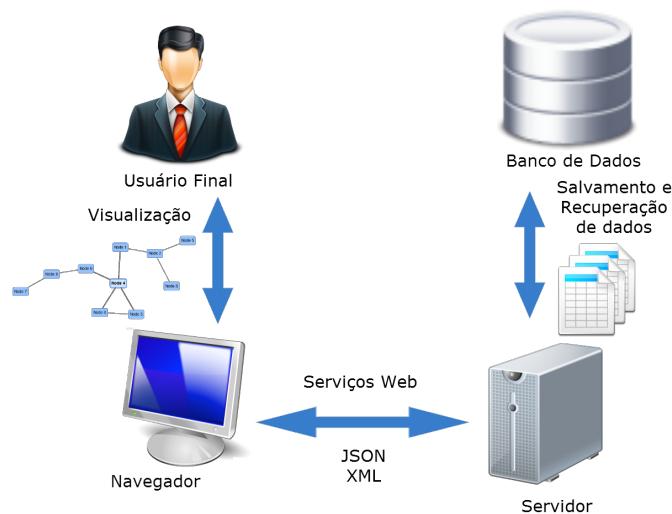


Figura 4.1: Esquema de serviço web requerido

## 4.3 Requisitos

A camada gráfica precisa prover certos recursos essenciais para a realização de uma simulação correta. Para isso, foram pensadas certas funcionalidades que buscam solucionar a questão da interação com o usuário com a aplicação. Todas elas possuem como ator o usuário.

### 4.3.1 Manipulação de Arquivos XML

O simulador deve conter opções que permitam manipular os arquivos XML. Um menu com a opção "Arquivo" deve estar disponível para realizar tais funções. A figura 4.2 apresenta os casos de uso de manipulação de arquivos.

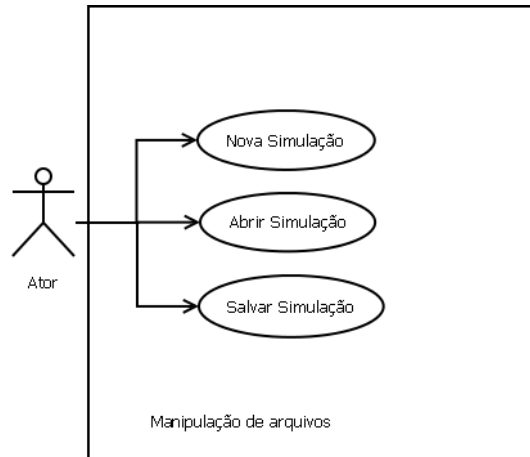


Figura 4.2: Casos de Uso para Manipulação de Arquivo XML

### Nova Simulação

A opção novo reinicia dados do simulador e permite ao usuário iniciar os novos parâmetros do programa. O caso de uso é acionado quando o usuário aciona a opção em Arquivo -> Novo no menu principal da aplicação. A criação de uma simulação corresponde analogamente à criação de um elemento **wdmsim** conforme descrito na tabela 3.1.

### Abrir Simulação

A opção de abrir simulação deve conseguir ler e interpretar um arquivo de configuração XML passado pelo usuário. Dentre as informações contidas no arquivo, estão a topologia da rede, definição de nós da rede e seus atributos e as chamadas de pacotes de dados que simularão o tráfego na rede conforme descrito na seção 3.2. A partir de tais informações, o simulador gera um topologia gráfica equivalente aos dados passados e permite assim executar a simulação.

O caso de uso é acionado quando o usuário aciona a opção em Arquivo -> Abrir no menu principal.

### Salvar Simulação

A opção de salvar simulação deve conseguir gerar um arquivo XML com configurações da simulação. Dentre as informações contidas no arquivo, estão a topologia da rede, definição de nós da rede e seus atributos e as chamadas de pacotes de dados que simularão o tráfego na rede conforme descrito na seção 3.2. A partir de tais informações, o simulador gera um topologia gráfica equivalente aos dados passados e permite assim executar a simulação.

O caso de uso é acionado quando o usuário aciona a opção em Arquivo -> Salvar presente no menu principal.

### 4.3.2 Ferramentas de Edição

Ferramentas de edição devem estar disponíveis ao usuário para tornar mais fácil a atualização do estado da topologia de rede visível ao usuário. A figura 4.3 apresenta os casos de uso da edição dos elementos da rede a ser modelada.

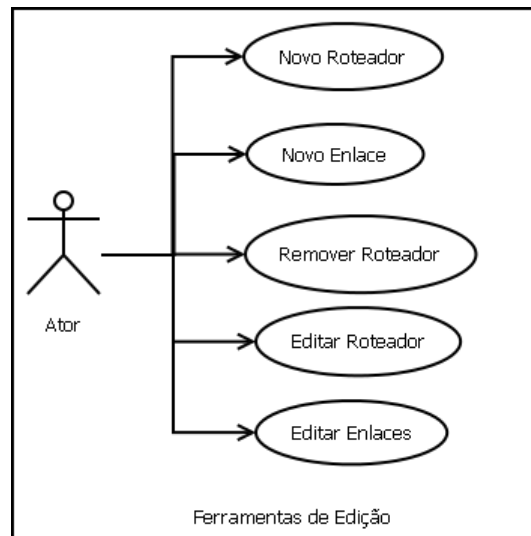


Figura 4.3: Casos de uso das ferramentas de edição

#### Novo Roteador

A funcionalidade novo roteador faz parte da barra de ferramentas do simulador posicionada na lateral esquerda da tela. Ela serve para criar um novo roteador na rede com as configurações padrão de um roteador óptico. A criação de um roteador corresponde analogamente à criação de um elemento **node** conforme descrito na tabela 3.1.

O caso de uso é acionado quando o usuário aciona a opção na barra de ferramentas ou pelo respectivo atalho descrito na seção 4.4.1

#### Novo Enlace

A funcionalidade novo enlace ajuda o usuário a criar um novo elo entre os nós da rede. Para criá-lo, basta selecionar dois nós e apertar o botão de novo link. A criação de um enlace corresponde à criação de um elemento **link** conforme descrito na tabela 3.1.

O caso de uso é acionado quando o usuário aciona a opção na barra de ferramentas ou pelo respectivo atalho descrito na seção 4.4.1

## Remover Roteador

A funcionalidade remover nó deleta um nó da rede e suas conexões. Para usar, basta selecionar o nó desejado e acionar o botão de remover. A remoção de um roteador equivale à remoção lógica de um elemento **node**.

O caso de uso é acionado quando o usuário aciona a opção na barra de ferramentas ou pelo respectivo atalho descrito na seção 4.4.1

## Editar Roteador

A funcionalidade editar roteador permite selecionar um nó da rede e modificar suas configurações. Ao se selecionar um nó e clicar a opção editar Nó, uma caixa de diálogo aparece e mostra as características do nó a serem modificadas.

A tabela 4.1 apresenta os campos que devem estar disponíveis à edição de um roteador da rede. A configuração e edição de tais campos equivalem à edição dos campos do elemento **node** conforme descrito na tabela 3.1.

Campo	Descrição
Id	Número identificador de um nó
Agregação	Tipo de agregação que o nó realiza
Agregação de entrada	Parâmetro que indica a capacidade de entrada
Agregação de saída	Parâmetro que indica a capacidade de saída
Conversor de comprimento	Parâmetro que indica a capacidade de conversão
Intervalo de conversão	Parâmetro que indica o quanto o comprimento pode variar

Tabela 4.1: Campos de configuração de Roteadores

O caso de uso é acionado quando o usuário aciona a opção na barra de ferramentas ou pelo respectivo atalho descrito na seção 4.4.1.

## Editar Enlaces

A funcionalidade editar links presentes permite alterá-los e removê-los. Para isso o usuário seleciona o nó ao qual o links é relacionado e seleciona a opção editar links.

A tabela 4.2 apresenta os campos que devem estar disponíveis à edição de um enlace da rede. A edição de tais campos deve ser equivalente à edição do elemento **link** conforme descrito na tabela 3.1.

O caso de uso é acionado quando o usuário clica na opção no painel de edição ou na caixa de diálogo aberta ao se editar um nó.

Campo	Descrição
Id	Número identificador do link
De	Indica a origem do link
Para	Indica o destino do link
Atraso	Indica o atraso do link em 1 ms/Km
Peso	Indica o peso do link para ponderação da rede
Largura de Banda	Indica a capacidade do link em banda

Tabela 4.2: Campos de configuração de Enlace

### 4.3.3 Cenário de Tráfego

A funcionalidade manter chamadas presentes permite adicionar, alterar, remover chamadas de pacotes na simulação. A figura 4.4 apresenta os casos de uso da edição do cenário de tráfego.

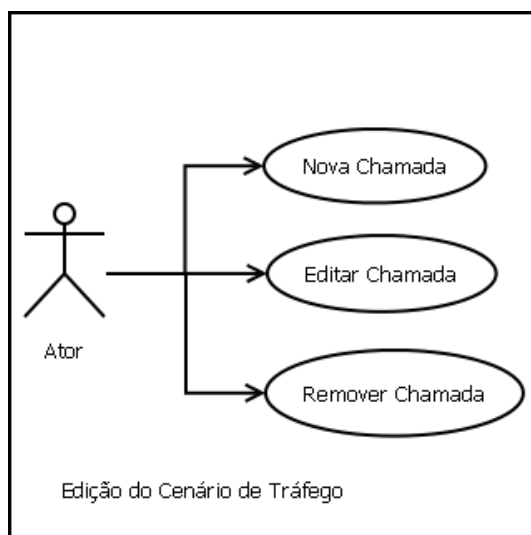


Figura 4.4: Possibilidades de Edição do cenário de Tráfego

A tabela 4.3 apresenta os campos que devem estar disponíveis à edição de uma chamada da simulação. A edição de tais campos deve ser equivalente à edição do elemento **calls** conforme descrito na tabela 3.1.

Campo	Descrição
Taxa	Velocidade de transmissão da chamada
Classe de Serviço ( <b>COS</b> )	Indica a classe de serviço ao qual uma chamada pertence
Peso	Indica a ponderação ao qual a chamada está submetida
Tempo Médio de Duração	Indica o tempo médio de realização da chamada

Tabela 4.3: Campos de configuração de Chamada

O caso de uso é acionado quando o usuário utiliza o painel de edição lateral. O botão de adicionar presente no painel cria uma chamada com valores padrão, o botão de excluir o remove da lista e é possível alterar os valores da chamada na lista.

#### 4.3.4 Ferramentas de Simulação

Ferramentas de simulação devem estar disponíveis para tornar viável a manipulação de parâmetros de carga e de execução da simulação. A figura 4.5 mostra os casos de uso das ferramentas de simulação.

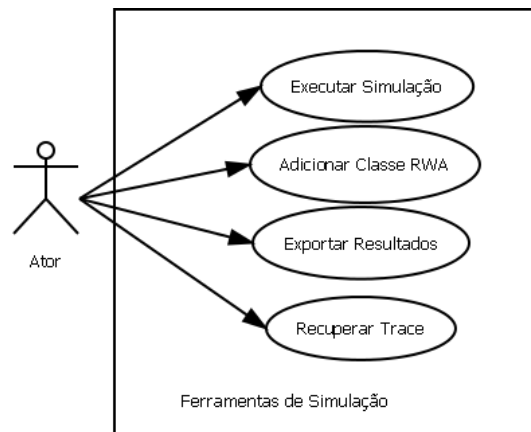


Figura 4.5: Casos de Uso das Ferramentas de Simulação

#### Executar Simulação

O simulador deve possuir habilidades de requisitar a execução da simulação modelada em interface gráfica ao servidor de simulação. A funcionalidade deve pedir informações de simulação ainda ausentes na modelagem ao se pressionar o botão de executar.

A tabela 4.4 apresenta os campos pedidos para se completar a modelagem da simulação e assim permitir executar remotamente a simulação.

Campo	Descrição
Routing Wavelength Assignment	O módulo RWA que contém a implementação do algoritmo de agregação
Carga	Representa o número de execuções da simulação para se obter melhor convergência de resultados
Versão	Versão usada do simulador WDMsim
Faixas de Onda	O número discreto de faixas no qual um enlace óptico pode ser dividido

Tabela 4.4: Campos presentes ao ser requisitar uma simulação



O caso de uso é acionado quando o usuário aciona a opção em Simulação -> Executar. A funcionalidade não permite ao usuário executar a simulação se não há topologia cadastrada ou se não são registradas chamadas para a simulação.

### Adicionar Classe RWA

A funcionalidade de adicionar uma classe de RWA permite ao usuário criar e implementar uma classe RWA que contém a lógica do algoritmo de roteamento e de alocação de comprimento de onda conforme descrito na seção 3.1. A funcionalidade deve compilar a classe e permitir ao usuário utilizá-la na simulação. O caso de uso é acionado quando o usuário aciona a opção em Simulação -> RWA Personalizado. A adição de uma classe equivale à adição de um elemento **rwa** conforme a tabela 3.1.

### Exportar Resultados

A funcionalidade exportar resultados permite ao usuário recuperar o arquivo de resultados gerado pela simulação e salvá-lo em disco. O arquivo contém certas estatísticas da simulação como quantidade de chamadas entre os nós, probabilidade média de bloqueio e taxa de bloqueio de banda média conforme descrito na seção 3.2.

### Recuperar Trace

A funcionalidade recuperar *trace* permite ao usuário recuperar o arquivo com o registro dos eventos gerados pela simulação e salvá-lo em disco. A tabela 4.5 contém os tipos de eventos simulados pelo WDMSim.

Evento	Descrição
flow-accepted	Ocorre quando o nó aceita um determinado fluxo
flow-blocked	Ocorre quando o nó bloqueia um determinado fluxo
flow-arrived	Ocorre quando o nó recebe uma solicitação de fluxo
flow-departed	Ocorre quando o nó solicita um fluxo
lightpath-created	Ocorre quando um caminho óptico é criado
lightpath-removed	Ocorre quando um caminho óptico é removido

Tabela 4.5: Eventos registrados no arquivo de trace

### 4.3.5 Gerar Estatísticas

A funcionalidade gerar estatísticas deve permitir ao usuário manipular e coletar estatísticas da simulação. Deve possibilitar gerar gráficos e tabelas com as métricas de rede coletadas e relacionadas à simulação. O arquivo de estatísticas, conforme citado na seção 3.3, apresenta o MBBR e o MBP.

## 4.4 Requisitos Desejáveis

Esta seção aborda os requisitos que foram traçados ao longo do desenvolvimento do WWSim e que, portanto, não foram prioridade durante a implementação.

### 4.4.1 Opções de Edição

O simulador deve conter funcionalidades que agilizem a criação e edição da topologia de rede. As funções devem ser disponibilizadas em um menu “Editar”.

#### **Criar Topologia Comuns**

O simulador deve apresentar a opção de simplificar a criação de topologias comuns ao contexto das redes. O caso de uso é acionado quando o usuário aciona a opção em Editar -> Criar Topologia. Dentro do menu, devem existir as opções Anel, Estrela, Aleatório e Completa, e o acionamento de tais opções quando acionadas criam a respectiva topologia.

#### **Copiar e Colar Roteadores**

O simulador deve poder copiar e colar em tela os nós da rede selecionados. O caso de uso é acionado quando o usuário aciona a opção em Editar -> Copiar e Colar ou pelo respectivo atalho.

#### **Deletar Simulação**

A funcionalidade permite deleta a topologia criada, eliminando os roteadores e enlaces criados no uso do programa. O caso de uso é acionado quando o usuário aciona a opção em Editar -> Deletar.

#### **Atalhos**

O simulador deve poder invocar certas funcionalidades a partir de atalhos do teclado. Os atalhos concebidos para melhorar a usabilidade do simulador estão descritos na tabela 4.6.

### 4.4.2 Salvar e Recuperar Simulação

O simulador deve poder salvar a simulação criada e permite ao usuário recuperar os dados submetidos através da interface. Ao se executar uma simulação, o seu registro é armazenado em banco de dados e a URL do sistema é alterada para que a simulação seja acessada novamente.

Atalho	Funcionalidade
Ctrl+Alt+A	Novo Roteador
Ctrl+Alt+C	Copiar e Colar Roteadores
Delete	Remover Roteador
Ctrl+Alt+E	Editar Roteador
Ctrl+Alt+L	Novo Enlace
Click Duplo	Editar Roteador

Tabela 4.6: Atalhos do Sistema

### 4.4.3 Acompanhamento de Simulação

Uma forma de acompanhar a simulação deve ser provida para permitir iniciar, pausar e controlar o tempo da simulação. A figura 4.6 apresenta um protótipo da barra de rolagem que acompanha o tempo de simulação dos eventos.



Figura 4.6: Protótipo da barra de rolagem que controla a visualização dos eventos

A medida que a barra de rolagem progride, os eventos registrados pela simulação são ilustrados graficamente para o usuário. O WWSim deve ilustrar os eventos conforme descrito na tabela 4.5. A barra de rolagem deve estar disponível ao usuário somente após a execução da simulação.

### 4.4.4 Manipulação Dinâmica de Eventos

A funcionalidade de adição de eventos é planejada para o simulador. A função permite alterar e adicionar os eventos, descrito na tabela 4.5, de chamadas e falhas ao longo da simulação e a partir deles gerar estatísticas, gráficos e tabelas.

# Capítulo 5

## Projeto e Arquitetura do WWSim

O projeto de arquitetura foi pensado para web. Dado isso, todos os serviços foram concebidos para serem executados a partir de um navegador de internet.

A arquitetura do sistema é baseada no *framework* GWT(*Google Web Toolkit*), um projeto de código aberto capaz de desenvolver aplicações complexas baseadas em navegadores de Internet. A ferramenta possui ampla documentação e permite ao desenvolvedor criar código HTML e JavaScript a partir de código em Java. O GWT realiza a conversão para javascript e HTML a partir de um *Cross-Compiler* presente em suas bibliotecas principais.

A arquitetura de software que melhor se ajusta ao *framework* GWT é a MVP(Model View Presenter). Tal arquitetura de software divide as responsabilidades do software em três tipos de componentes: aqueles que representam os dados e entidades as serem manipulados(Model), aqueles que geram representação visual ou saída de dados(View) e aqueles que controlam o acesso aos dados e contêm as regras de negócio(Presenter).

Dado essa decisão arquitetural, o código foi dividido em três pacotes principais: `client`, `server` e `shared`. Cada pacote contém código que pode ser executado no cliente, no servidor ou compartilhadamente. A figura 5.1 mostra o diagrama de pacotes.

Conforme descrito na seção 4.2, a arquitetura do WWSim segue o modelo cliente-servidor. O servidor contém dois componentes principais: o WDMSim, que contém a lógica de execução da simulação e a lógica de geração de resultados, de estatísticas e de registro de eventos; e o Banco de Dados H2 que contém as tabelas relacionais que armazenam as simulações executadas e suas estatísticas. A figura 5.2 ilustra tais decisões arquiteturais do WWSim.

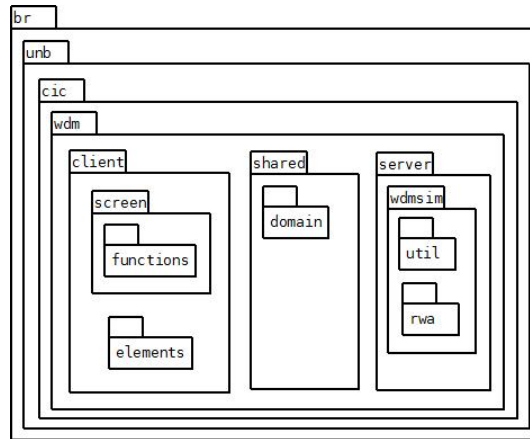


Figura 5.1: Diagrama de Pacotes

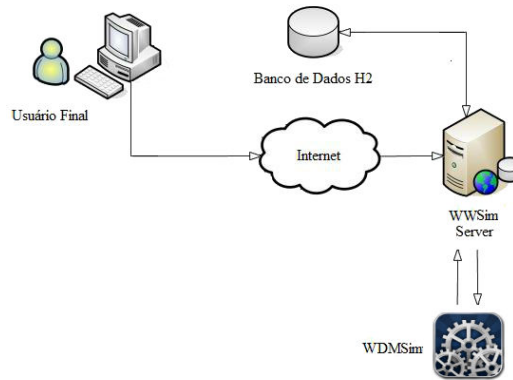


Figura 5.2: Arquitetura Geral da Aplicação

## 5.1 Pacotes da Aplicação

O pacote `shared` contém o código de objetos de dados utilizados para comunicação entre cliente e servidor e também modelam o banco de dados relacional. O banco de dados relacional é modelado na figura 5.3. Esse pacote é equivalente à camada de *Model* no padrão MVP.

O pacote `client` contém o código que controla a visualização dos dados necessários para o uso do simulador. Classes inclusas nesse pacote são convertidas para código HTML e Javascript quando requisitadas pelo navegador. O envio e requisições de dados de simulação são realizados por meio de requisições REST (Representational State Transfer) assíncronas ou submissões de formulários HTML. O pacote é equivalente à camada de *View*.

O pacote `server` contém o código de acesso ao banco de dados, é o responsável por disponibilizar os serviços da aplicação relacionados à execução da simulação e lógica de

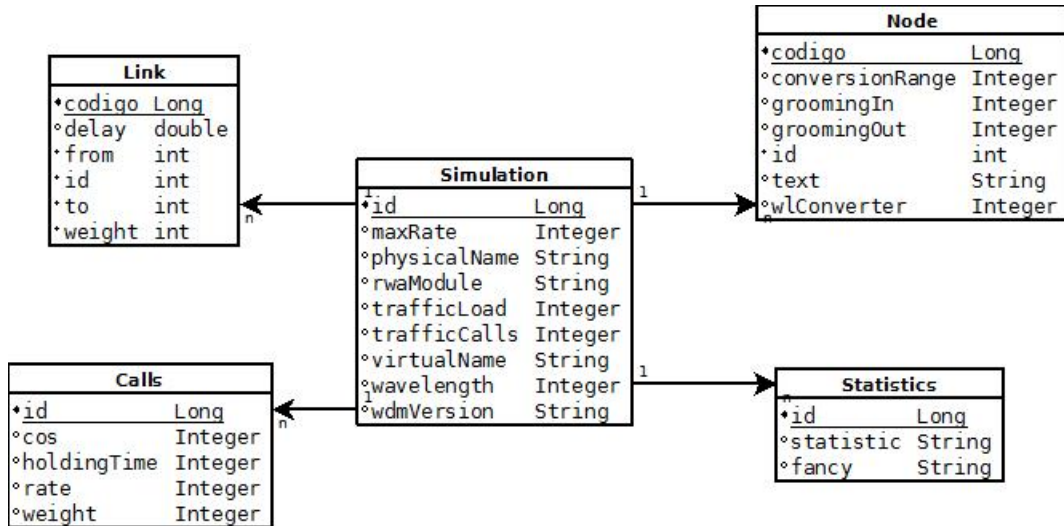


Figura 5.3: Diagrama de Banco de Dados relacional

histórico da aplicação. O pacote é equivalente à camada de *Presenter* do MVP. Uma série de serviços REST foram implementados no pacote `server` para facilitar a interação com o usuário da aplicação. A tabela 5.1 apresenta os serviços web que podem ser invocados pelo navegador do usuário ou por uma máquina que necessite de lógica RPC (*Remote Procedure Call*). O pacote `server` possui também uma série de *servlets*, classes responsáveis por receber requisições em uma aplicação web java. Elas são responsáveis por realizar manipulação de arquivos conforme mostrado na figura 5.2.

Serviço	Descrição
openSimulation	Recupera simulação anteriormente salva
saveSimulation	Salva a simulação em banco
addCustomRwa	Adiciona uma classe RWA personalizada
getRwaOptions	Recupera a lista de classes RWA disponíveis
createUploadUrl	Cria uma url para realiza upload de arquivo XML
executeSimulation	Executa a simulação e salvá-lo em banco
openStatistics	Recupera estatísticas geradas pela execução

Tabela 5.1: Serviços REST Disponíveis

## 5.2 Padrões de Projeto

Na arquitetura da aplicação certos padrões de projeto orientados a objeto foram utilizados para tornar o código mais simples e fácil de se manter.

O padrão *Singleton* determina que somente uma instância de determinada classe seja criada para toda a aplicação. O padrão foi utilizado para a criação dos objetos principais da visão do usuário.

Classe da Servlet	Descrição
UploadSimulationServlet	Cria lógica de receber arquivo XML de parâmetro e salvá-lo em banco
DownloadServlet	Recupera arquivo XML de simulação salva em banco
DownloadResultsServlet	Realizar o download de resultados da simulação
DownloadTraceServlet	Recuperar arquivo de trace de simulação salva em banco
DownloadStatsServlet	Recuperar arquivo de estatísticas de simulação salva em banco

Tabela 5.2: Servlets Disponíveis para manipulação de arquivos

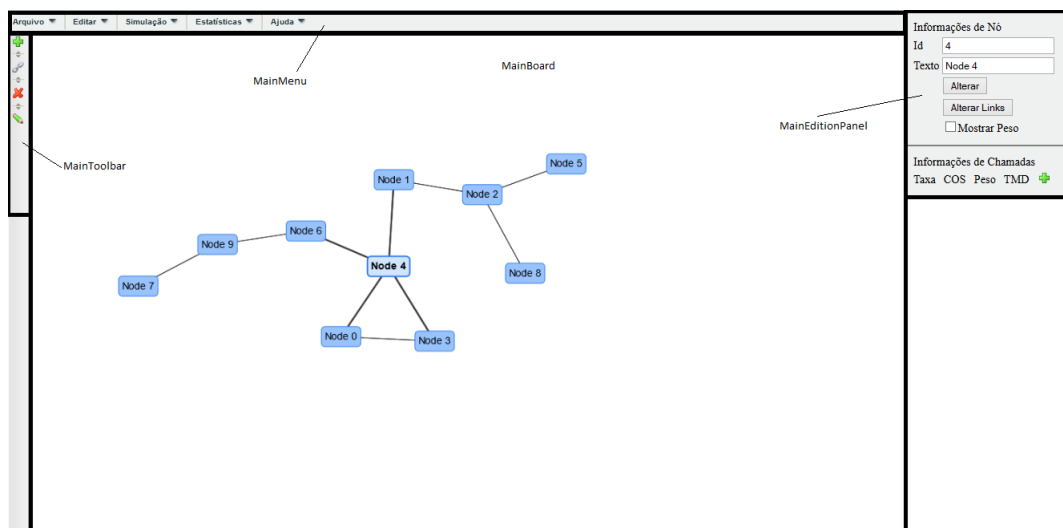


Figura 5.4: Uso do padrão Singleton na criação da camada de View

O padrão *Command* torna mais fácil a construção de objetos que possuem lógica diferente e permite uni-los sob uma mesma assinatura de método. O uso do padrão permitiu que os objetos acionados por eventos de tela fossem generalizados para outros tipos de evento.

### 5.3 Implementação

A implementação do projeto do simulador foi feito de forma incremental e iterativa. O controle de versionamento foi feito utilizando a ferramenta Git e está disponível para download no link. <https://bitbucket.org/guihred/wdmsim/>

O desenvolvimento seguiu 5 fases: idealização, prototipação, desenvolvimento, validação e documentação.

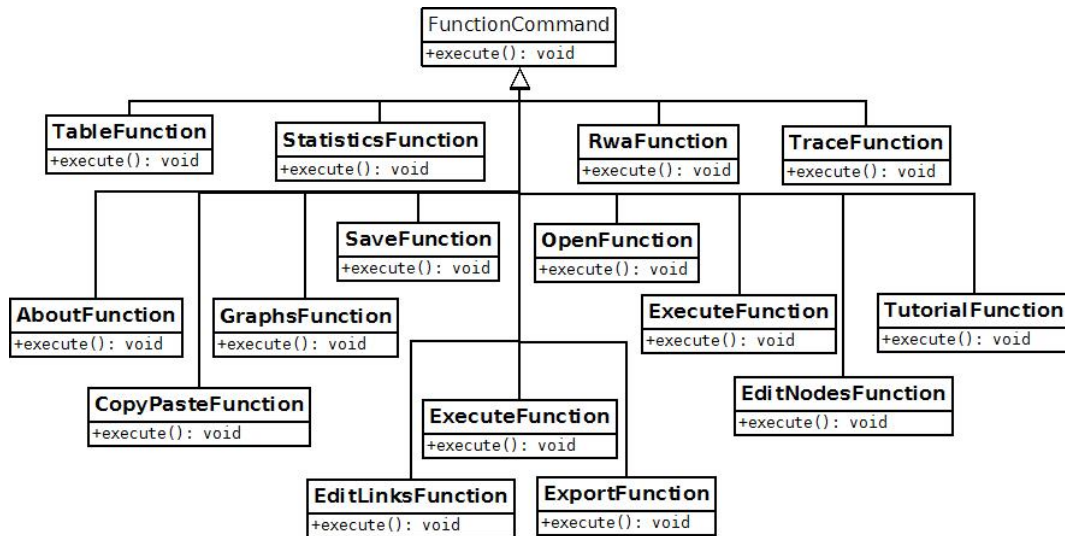


Figura 5.5: Uso do padrão Command na criação das funcionalidades da aplicação

Na fase de idealização, foram levantados as principais funcionalidade a que o simulador deveria atender. A partir desses casos de uso foi possível desenvolver o protótipo da aplicação.

Na fase de prototipação, foi definida a biblioteca de visualização da rede que oferece melhor adequação às funcionalidades do sistema [10]. A API permite configurar aspectos da visualização da rede de maneira integrada com o GWT, a partir daí foi possível implementar o protótipo funcional com os casos de uso Novo Roteador, Novo Link e Remove.

Na fase de desenvolvimento, foram definidos os serviços que o servidor da aplicação deveria oferecer. O uso da biblioteca RestyGWT, [8], permitiu definir serviços acessáveis pelos métodos HTTP padrão. Foram implementadas 89 classes Java ao todo, totalizando 5601 linhas relevantes de código Java.

Na fase de validação, testes de implementação do sistema foram feitos para comprovar as características das funcionalidades apresentadas anteriormente nessa monografia. Nos testes, os casos de uso descritos foram seguidos de modo a encontrar erros de implementação.

Também foi gerado o Javadoc, série de arquivos padrão para a documentação de código Java. Esse arquivos têm a intenção de demonstrar aspectos da implementação e permitir a futuros desenvolvedores prestar manutenção de código.



## 5.4 Diagramas de Classe

Para melhor visualização da organização da aplicação, foram desenhados diagramas de classe esquematizando os objetos, entidades e pacotes do projeto.

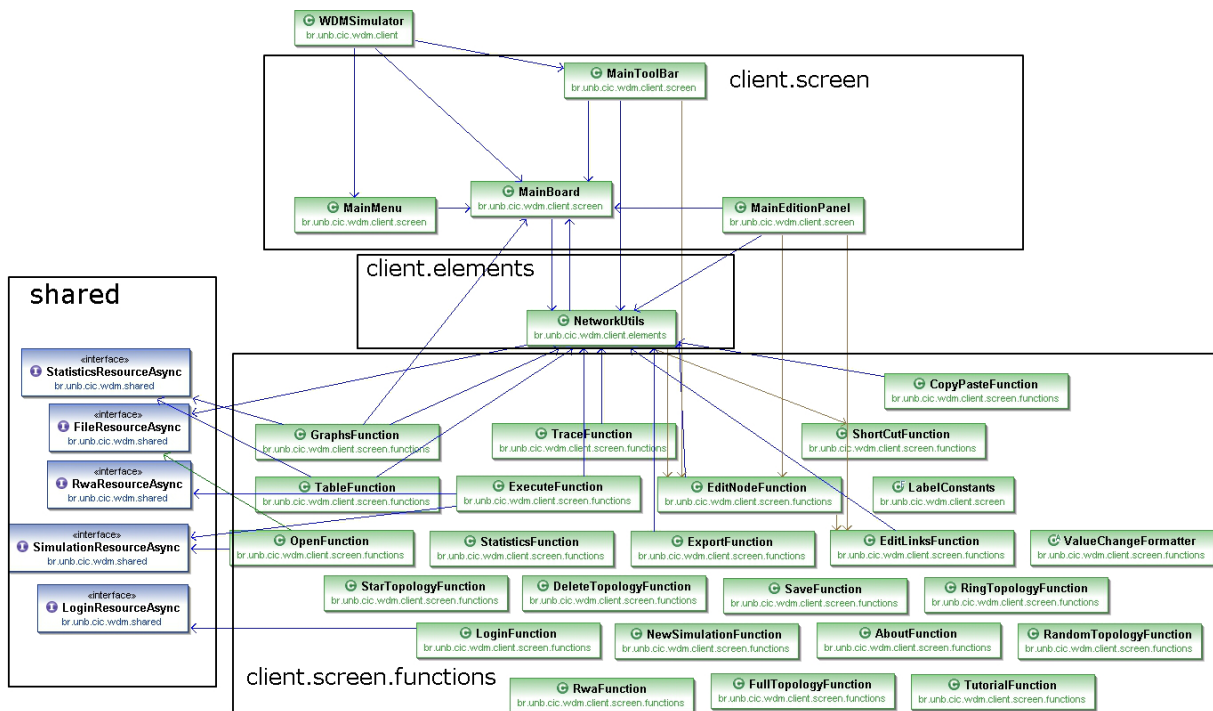


Figura 5.6: Diagrama de Classes de Cliente

Na figura 5.6, é possível visualizar o diagrama das classes e associações que são responsáveis por gerar a visualização do WWSim em navegador. A classe superior `WDM Simulator` é responsável por iniciar a renderização da página do simulador e realiza a iniciação das classes do pacote `client.screen`. Uma das classes mais importantes é `NetworkUtils`, dentro do pacote `client.elements`, que é responsável por centralizar a lógica de visualização da rede. O pacote `client.screen.functions` contém um conjunto de classes que realizam o fluxo de execução dos casos de uso descritos anteriormente. E por último, o pacote `shared` contém as interfaces que permitem realizar a chamada a serviços remotos.

Na figura 5.7, é possível visualizar as classes e associações que são responsáveis por manter a lógica de salvamento e execução de simulações e por manter os serviços Web remotos e URLs de submissão de arquivos. No topo, são apresentadas as classes que mantêm os serviços Web da aplicação. À esquerda, é possível visualizar as servlets necessárias para receber requisição e submissão de arquivos. Ao centro, encontra-se a classe de utilidades da aplicação como por exemplo a conversão de XML, compilação de classe RWA, e execução de simulação. À direita, é possível visualizar a classe responsável por realizar operações no banco de dados. E à direita inferior, a classe `SimulatorGWT` que é

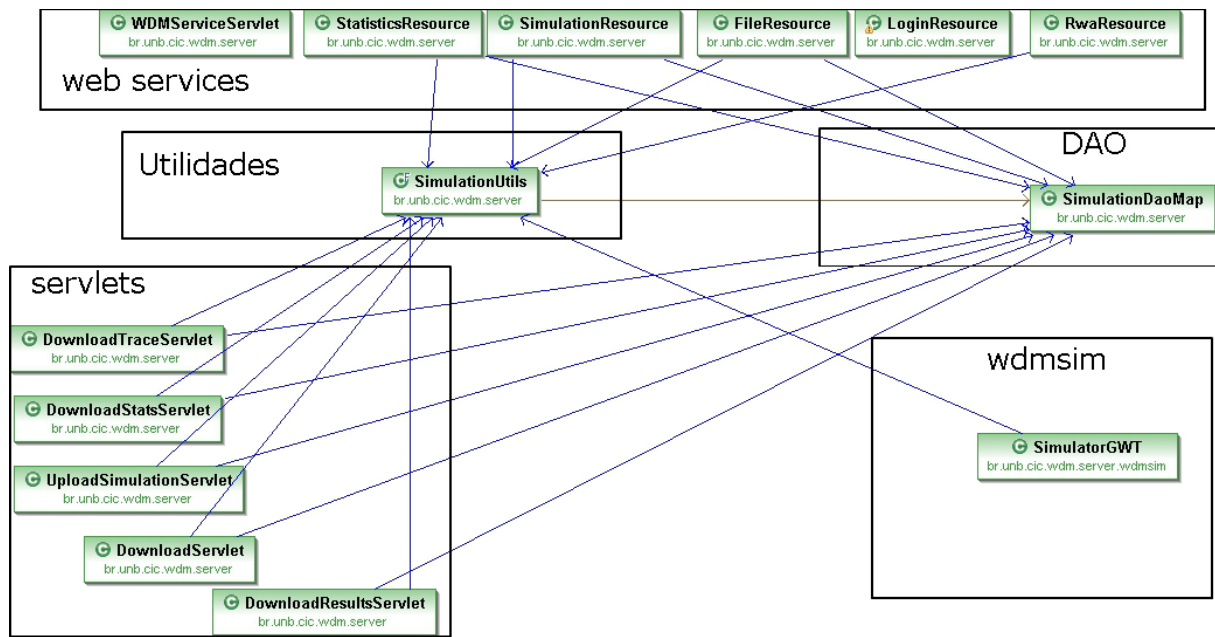


Figura 5.7: Diagrama de Classes do Servidor

responsável por inicial o fluxo de execução da simulação e está encapsulado para melhor entendimento do componente.

# Capítulo 6

## Conclusão

Este capítulo apresenta as contribuições da monografia e a implementação realizada. Também estão descritas as possibilidades de expansão da implementação do sistema.

### 6.1 Contribuições

Com essa monografia, foi possível concluir a implementação da interface Web com a capacidade de modelar a topologia física de redes ópticas WDM. A interface é acessável por meio de um navegador Web com suporte a HTML5.

Foi possível construir a interface gráfica capaz de gerar e receber arquivos que seguem à sintaxe XML definida na tabela 3.1, de modo a produzir um arquivo válido como parâmetro para o simulador WDMsim.

Essa monografia também contribuiu com a criação de uma série de serviços REST capazes de receber como parâmetro um objeto de simulação no formato JSON e executar seguindo as premissas do WDMsim conforme a seção 3.2. Tais serviços também permitem recuperar informações de estatísticas de simulação geradas e também de registro de eventos de rede.

Foi possível criar uma interface gráfica capaz de realizar acesso aos arquivos de estatísticas e aos arquivos de eventos por meio de chamadas a os serviços definidos no servidor. Tais serviços, implementados nesse projeto, permitem gerenciar as simulações executadas, salvá-las e recuperá-las

### 6.2 Trabalhos Futuros

Certos casos de uso porém, não puderam ser implementados devido à falta de definição e de conhecimento específico na biblioteca de visualização. Esses casos de uso foram a Acompanhamento de Simulação e Manipulação Dinâmica de Eventos.

O acompanhamento de simulação, conforme descrito na seção 4.4.3, ilustra os eventos criados pelo simulador e mostra a criação ou remoção de um caminho óptico, algo como um enlace de determinada cor sendo criado entre os roteadores. Esse trabalho propõe a realização de tal funcionalidade como trabalho futuro, pois essa trará novos entendimentos sobre o funcionamento das redes ópticas e do simulador WDMSim.

A manipulação dinâmica de eventos também é tida como um melhoramento. A funcionalidade, conforme descrita na seção 4.4.4, prevê a possibilidade de se adicionar os eventos, descrito na tabela 4.5, na simulação e tornar possível ao usuário gerar cálculos e executar a visualização de tais eventos.

# Referências

- [1] The network simulator - ns-2. <http://www.isi.edu/nsnam/ns/>, 2014. 10
- [2] S.B. Alexander, S.W. Chaddick, R.C. Litz, and C.D. Smith. Remodulating channel selectors for wdm optical communication systems. <http://www.google.com/patents/US8306424>, 2012. US Patent 8,306,424. 4
- [3] Gurusamy Mohan C. Siva Murthy. Wdm optical networks: Concepts, design and algorithms, 2001. 5
- [4] FTTH Council. North american ftth status—march 31, 2011. [www.ftthcouncil.org](http://www.ftthcouncil.org), 2011. 1
- [5] A. C. Drummond. WDMsim: WDM Optical Network Simulator. <http://www.lrc.ic.unicamp.br/wdmsim/>, 2013. 2, 12
- [6] George N. Rouskas e Harry G. Perros. A tutorial on optical networks. Department of Computer Science, 2002. Raleigh, NC, USA. 5, 8
- [7] André Costa Drummond e Nelson Luís Saldanha da Fonseca. Agregação de Tráfego em Redes Ópticas com Multiplexação por Comprimentos de Onda, 2013. 11, 12
- [8] Fusesource. Making restful services accessible to gwt applications. <http://restygwt.fusesource.org/index.html>, 2014. RestGWT. 29
- [9] Keith W. Ross James F. Kurose. *Computer Networking: A top Down Approach*. Addison-Wesley, 6 edition, 2013. 1
- [10] Almende Organizing Networks. Network visualization api. <http://almende.github.io/chap-links-library/network.html>, 2014. Chap Library. 29
- [11] NTT. World record one petabit per second fiber transmission over 50-km. <http://www.ntt.co.jp/news2012/1209e/120920a.html>, 2012. 1
- [12] Afonso J. F. Cardoso e Carlos R. L. Francês Paulo H. G. Bezerra. Performance evaluation of algorithms for wavelength assignment in optical wdm networks, 2010. International Journal of Computer Science and Network Security. 8, 9
- [13] Steven S. Skiena. *The Algorithm Design Manual*. Springer, 2 edition, 2008. 8
- [14] W3C and Mozilla Developer Network. Evolução da web. <http://www.evolutionoftheweb.com/#/growth/day>, 2012. 1
- [15] Keyao Zhu. Traffic grooming in an optical wdm mesh network, 2002. 6, 7

# Anexo A

## Manual de Usuário

Este anexo apresenta o Manual de usuário. Ele tem o propósito de trazer as funcionalidades desenvolvidas e ilustrá-las para a correta averiguação dos capacidades desenvolvidas.

### A.1 Tela Inicial

A tela inicial contém menu, uma barra de ferramentas, um painel lateral e um quadro principal onde será visualizada a rede modelada. A figura A.1 mostra a tela do WDM Simulator.

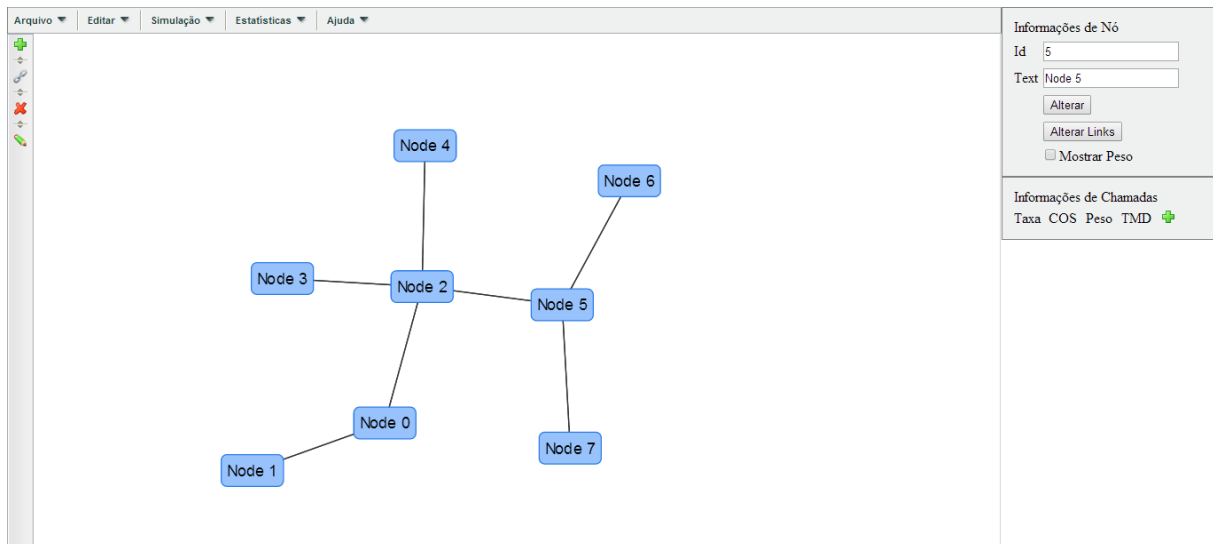


Figura A.1: Tela principal do WDM Simulator

### A.2 Criação da Topologia

A partir da tela inicial é possível começar a construção da rede.

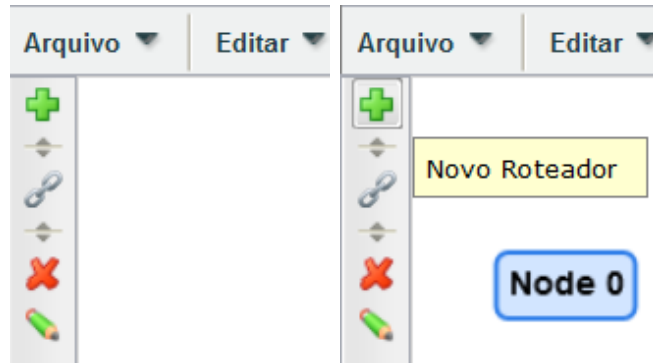


Figura A.2: Pode-se criar um roteador novo

A criação de um roteador é feita pelo clique do botão novo roteador ou pelo atalho Ctrl+Alt+A conforme ilustrado na figura A.2. O roteador criado é iniciado com as configurações padrão de um roteador.

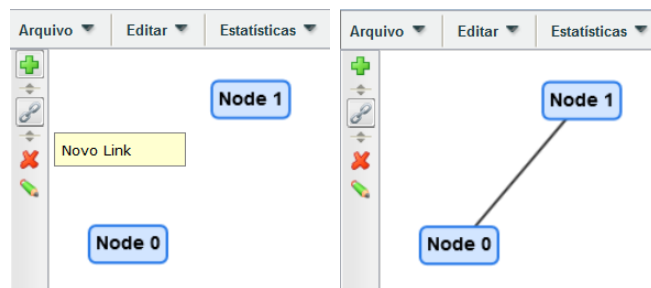


Figura A.3: Adicionar um novo link entre roteadores

A criação de um novo enlace é feita com a seleção simultânea de dois roteadores na rede. Para seleccionar dois roteadores selecione com o cursor enquanto pressiona a tecla Ctrl. Após seleccionar pressione o botão Novo Link ou use o atalho Ctrl+Alt+L. A figura A.3 ilustra a criação de um link.

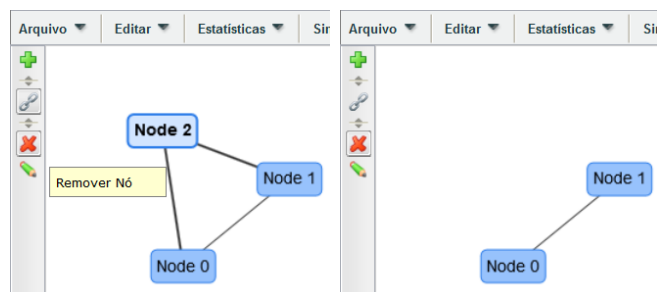


Figura A.4: Remover um roteador já existente

A remoção de um roteador é feita com a sua seleção e o clique no botão Remover Nó ou pelo atalho Delete. A figura A.4 ilustra como é feita a remoção.

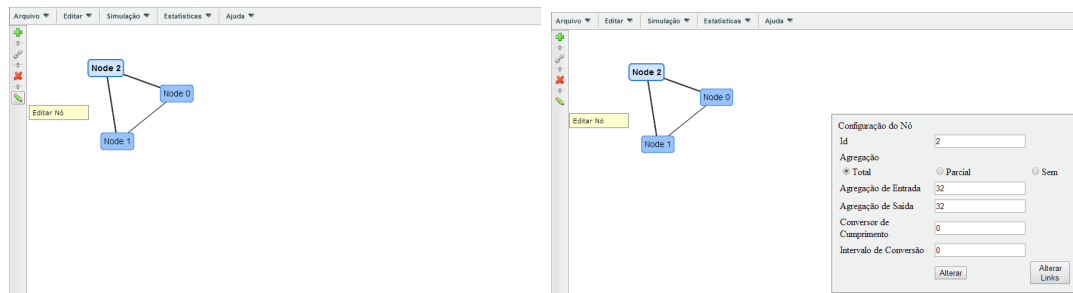


Figura A.5: Editar roteador já criado

Um roteador criado pode ter suas configurações alteradas a partir do botão Editar Nó. A funcionalidade pode ser acionada quando se clica duas vezes no roteador. A função apresenta uma tela com a configuração do roteador selecionado conforme mostrado na figura A.5.

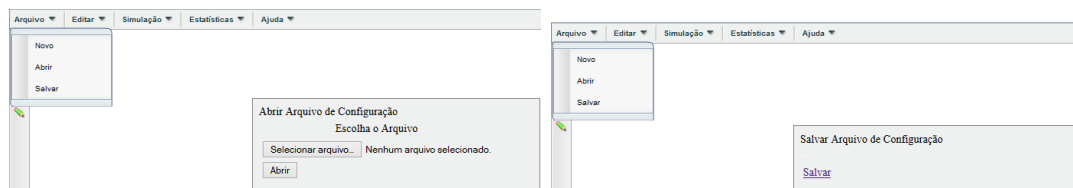




Figura A.6: É possível salvar ou submeter um arquivo em XML com a informação da simulação

Dada a rede já modelada e montada, é possível salvar a simulação de rede gerada em arquivo XML. A figura A.6 mostra como as funcionalidades podem ser acionadas.

A figura A.8 apresenta o arquivo gerado pela rede modelada na figura A.7.

### A.3 Adição de Chamadas

A adição de chamadas de simulação é feita no painel lateral. A figura A.9 apresenta a tela de chamadas. No painel, é possível mudar o Tempo Médio de Duração(TMD), Classe de Serviço (COS), o peso e a taxa de transmissão. É possível adicionar uma chamada com um clique no botão de adicionar superior verde , e remover as chamadas no botão lateral vermelho .



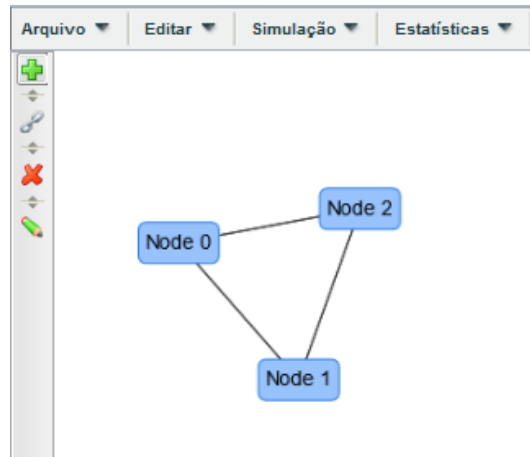


Figura A.7: Rede Simples Modelada

```
<?xml version="1.0" encoding="UTF-8"?>
<wdmsim version="0.1">
<rwa module="My2RWA"/>
<trace file="trace.tr"/>
<traffic calls="10000" load="10" max-rate="0"/>
<virtual-topology name="NSFNet"/>
<physical-topology name="NSFNet" wavelengths="16">
<nodes>
<node grooming-in-ports="32" grooming-out-ports="32" id="0" wlconversion-range="0"
wlconverters="0"/>
<node grooming-in-ports="32" grooming-out-ports="32" id="1" wlconversion-range="0"
wlconverters="0"/>
<node grooming-in-ports="32" grooming-out-ports="32" id="2" wlconversion-range="0"
wlconverters="0"/>
</nodes>
<links>
<link bandwidth="10000" delay="1.0" destination="0" id="0" source="2" weight="1500"/>
<link bandwidth="10000" delay="1.0" destination="2" id="1" source="0" weight="1500"/>
<link bandwidth="10000" delay="1.0" destination="1" id="2" source="2" weight="1500"/>
<link bandwidth="10000" delay="1.0" destination="2" id="3" source="1" weight="1500"/>
<link bandwidth="10000" delay="1.0" destination="1" id="4" source="0" weight="1500"/>
<link bandwidth="10000" delay="1.0" destination="0" id="5" source="1" weight="1500"/>
</links>
</physical-topology>
</wdmsim>
```

Figura A.8: Arquivo XML de Entrada/Saída respectivo à figura A.7

## A.4 Execução da Simulação

Para executar a simulação, basta selecionar a opção executar simulação no menu. Uma caixa de diálogo será apresentada com as informações necessárias para se completar as informações para a execução. A figura A.10 ilustra a caixa de diálogo e como executá-la.

## A.5 Criação de uma Classe RWA

A criação de uma classe RWA pode ser feita pela função adicionar RWA Personalizado.

A figura A.11 apresenta uma implementação do padrão de módulos RWA definidos pelo WDMSim. A classe possui três métodos principais `simulationInterface`, `flowArrival` e `flowDeparture`.

Informações de Chamadas				
Taxa	COS	Peso	TMD	
100	1	1	1	+
5000	0	5	6	x

Figura A.9: Funcionalidade de Manter Chamadas

Simulação ▾

- Executar
- Exportar Resultados
- Trace

Configuração da Simulação

Routing and Wavelength Assignment

MyRWA     My2RWA

Chamadas

Carga

Taxa Máxima

Faixas de Banda

Figura A.10: Funcionalidade de executar simulação de rede

O método a ser chamado primeiro é o `simulationInterface` e é responsável por passar os primeiros parâmetros da rede: `PhysicalTopology`, contém informações da topologia de enlaces físicos da rede; `VirtualTopology`, contém informações dos enlaces virtuais da topologia da rede; `ControlPlaneForRWA`, contém os métodos de controle de eventos da simulação.

O método `flowArrival` é invocado quando há determinado fluxo de dados a ser roteado. No exemplo, faz-se a opção pela estratégia *Least-Used* onde os comprimentos de onda menos usados têm preferência na escolha do melhor caminho óptico.

O método `flowDeparture` é invocado quando o fluxo de dados encerra. No exemplo, não há implementação para esse método.

```

public class My2RWA implements RWA {
    private ControlPlaneForRWA cp;
    private WeightedGraph graph;
    private PhysicalTopology pt;
    private VirtualTopology vt;

    public void flowArrival(Flow flow) {
        int [] nodes;
        int [] links;
        int [] wvls;
        long id;
        LightPath [] lps = new LightPath [1];
        lps [0] = getLeastLoadedLightpath (flow);
        if (lps [0] instanceof LightPath) {
            if (cp.acceptFlow (flow.getID (), lps)) {
                return;
            }
        }
        nodes = Dijkstra.getShortestPath (graph, flow.getSource (), flow.getDestination ());
        if (nodes.length == 0) {
            cp.blockFlow (flow.getID ());
            return;
        }
        links = new int [nodes.length - 1];
        for (int j = 0; j < nodes.length - 1; j++) {
            links [j] = pt.getLink (nodes [j], nodes [j + 1]).getID ();
        }
        wvls = new int [links.length];
        for (int i = 0; i < pt.getNumWavelengths (); i++) {
            for (int j = 0; j < links.length; j++) {
                wvls [j] = i;
            }
            if ((id = vt.createLightpath (links, wvls)) >= 0) {
                lps [0] = vt.getLightpath (id);
                cp.acceptFlow (flow.getID (), lps);
                return;
            }
        }
        cp.blockFlow (flow.getID ());
    }
    @Override
    public void flowDeparture (long id) {
    }
    private LightPath getLeastLoadedLightpath (Flow flow) {
        long abw_aux, abw = 0;
        LightPath lp_aux, lp = null;
        TreeSet <LightPath> lps = vt.getAvailableLightpaths (flow.getSource (),
            flow.getDestination (), flow.getRate ());
        if (lps != null && !lps.isEmpty ()) {
            while (!lps.isEmpty ()) {
                lp_aux = lps.pollFirst ();
                //Get the available bandwidth
                abw_aux = vt.getLightpathBWAvailable (lp_aux.getID ());
                if (abw_aux > abw) {
                    abw = abw_aux;
                    lp = lp_aux;
                }
            }
        }
        return lp;
    }
    public void simulationInterface (PhysicalTopology pt, VirtualTopology vt, ControlPlaneForRWA cp) {
        this.pt = pt;
        this.vt = vt;
        this.cp = cp;
        this.graph = pt.getWeightedGraph ();
    }
}

```

Figura A.11: Implementação do algoritmo RWA na forma de classe Java