

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

MARCOS TADEU DE ARAUJO TORRES

**UMA PROPOSTA DE FRAMEWORK PARA
SIMULAÇÃO DE REDES OTN**

VITÓRIA
2010

MARCOS TADEU DE ARAUJO TORRES

**UMA PROPOSTA DE FRAMEWORK PARA
SIMULAÇÃO DE REDES OTN**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Engenharia Elétrica.
Orientador: Prof. Dr. Anilton Salles Garcia.
Co-orientador: Prof. Dr. Antônio M. F. Frasson

VITÓRIA
2010

*"Não se deve ir atrás de objetivos fáceis. É preciso buscar o que só
pode ser alcançado por meio dos maiores esforços."*

(Albert Einstein)

Dedicatória

Dedico essa dissertação a minha família pelo apoio dado em todos os anos de estudo.

Agradecimentos

Aos meus pais pelo apoio, carinho e dedicação.

Aos meus irmãos Amanda, Rodrigo e Bárbara pela amizade e compreensão.

Ao meu orientador, Anilton Salles Garcia, pela confiança e apoio durante a realização deste trabalho.

Ao meu co-orientador Prof. Dr. Antônio M. F. Frasson pelo apoio e pelas discussões acerca do tema em questão.

Aos meus amigos Felipe Pedroni, Rodrigo Stange, Pedro Paulo Barcelos, Fabbiano Ferrari e a todos que trabalharam comigo em projetos relacionados com esta dissertação.

A todos os amigos do LPRM, do LabTel, da EngComp, CT e UFES pelos bons momentos de convivência e amizade.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico, CNPq, pelo suporte financeiro que permitiu a realização deste trabalho.

À PADTEC pelos projetos desenvolvidos com a UFES.

A todos que de alguma forma contribuíram com a realização deste trabalho e que me ajudaram a crescer como ser humano.

Sumário

Lista de Figuras	viii
Lista de Tabelas.....	x
Lista de Siglas/Acrônimos	xi
Capítulo 1: Introdução.....	17
Capítulo 2: Redes OTN	20
2.1 Introdução.....	20
2.2 Recomendações ITU referentes às Redes OTN	22
2.3 Conceitos Gerais.....	23
2.4 Estrutura de Camadas	26
2.5 Considerações Finais	35
Capítulo 3: Simulações.....	36
3.1 Simulação computacional.....	37
3.2 Trabalhos relacionados	40
3.3 Ambientes de simulação	41
3.4 O ambiente de simulação Omnet ++	44
3.5 Conclusão	49
Capítulo 4: Proposta do Framework	50
4.1 Modelagem dos componentes	54
4.1.1 Camadas OTN	55
4.1.2 Canal óptico e mensagens OTN	63
4.1.3 Gerador de tráfego	66
4.1.4 Módulos extras	67
4.2 Estrutura geral do framework.....	67
4.3 Conclusão	69
Capítulo 5: Simulações e resultados preliminares	71
5.1 Testes	71
5.2 Módulos Extras.....	75
5.2.1 Plano de controle GMPLS	75

5.2.2	<i>Discovery</i>	75
5.2.3	Comunicação externa	76
5.3	Conclusão	80
Capítulo 6: Conclusão		81
6.1	Contribuições e Resultados	82
6.2	Dificuldades.....	83
6.3	Trabalhos Futuros	83
Referências Bibliográficas		85

Lista de Figuras

Figura 2.1: Diagrama ilustrando a relação entre algumas normas OTN da ITU-T. (Iniewski, McCrosky, & Minoli, 2008)	23
Figura 2.2: Exemplo de modelagem funcional utilizando elementos da recomendação (ITU-T, G.805, 2000).	25
Figura 2.3: Hierarquia OTN. (Iniewski, McCrosky, & Minoli, 2008)	26
Figura 2.4: TCP's das camadas OTN (EXFO, 2006).	27
Figura 2.5: Estrutura básica de transporte OTN (EXFO, 2006).	27
Figura 2.6: Exemplo de multiplexação em redes OTN (ITU-T, G.709, 2009)	29
Figura 2.7: Monitoramento de conexões tandem (ITU-T, G.709, 2009).	31
Figura 2.8: Funções da adaptação e terminação da OTS (ITU-T, G.798, 2006).	32
Figura 2.9: Funções de terminação da OTS (ITU-T, G.798, 2006).	32
Figura 2.10: Funções da adaptação entre as camadas OTS e OMS (a) OTSn/OMSn_A_So, (b)OTSn/OMSn_A_Sk (ITU-T, G.798, 2006).	33
Figura 2.11: Estrutura das interfaces OTN. (ITU-T, G.709, 2009)	34
Figura 3.1: Eventos discretos x contínuos. (Chwif, 1999)	37
Figura 3.2: Relação hierárquica entre os módulos simples, compostos e uma rede no OMNeT++ (Varga & Hornig, An overview of the OMNeT++ simulation environment, 2008).	45
Figura 3.3: Configuração de uma simulação no OMNeT++ (Gamer & Scharf, 2008).	46
Figura 3.4: Interface da IDE do OMNeT++ 4.1 editando um arquivo NED (Site do OMNeT++).	47
Figura 3.5: Diagrama de Sequências no OMNeT++ (Site do OMNeT++).	48
Figura 4.1: Estrutura proposta do <i>Framework</i>	52
Figura 4.2: Função de terminação de trilha OTSn_TT_Sk (ITU-T, G.798, 2006).	56
Figura 4.3: OTSn_TT_Sk. (a) código em linguagem NED. (b) Representação gráfica.	58
Figura 4.4: Declaração de métodos e atributos da classe OTSn_TT_Sk.	59
Figura 4.5: Equipamento OTN modelado no OMNeT++.	61
Figura 4.6: Representação visual das camadas OTN: a) (ITU-T, G.872, 2001) b)framework.	62
Figura 4.7: Módulo de gerência local.	63
Figura 4.8: Tempo de envio nos canais do OMNeT++.	64
Figura 4.9: Tempos de transmissão e propagação com e sem conversão O-E-O.	65
Figura 4.10: Organização dos componentes do <i>framework</i> dentro do OMNeT++.	68
Figura 4.11: Componentes do <i>framework</i> e modos de uso.	70
Figura 5.1: Equipamento OTN genérico modelado no OMNeT++	72

Figura 5.2: Rede OTN de teste.	72
Figura 5.3: Diagrama de sequência em teste de correlação de alarmes em situação normal. ..	73
Figura 5.4: Diagrama de sequência em teste de correlação de alarmes em situação com dLOS-P na OTS.	74
Figura 5.5: Resultados de consulta SNMP a um roteador da simulação na interface do Cacti.	79
Figura 5.6: Interface Java para conexão com a simulação.	80

Lista de Tabelas

Tabela 2.1: Taxa nominal de bits definida na (ITU-T, G.709, 2009).....	29
Tabela 4.1: Resumo do que foi desenvolvido nos trabalhos anteriores.....	54
Tabela 4.2: Resumo do que foi desenvolvido nesta dissertação.....	68
Tabela 5.1: Resultado dos testes simples de desempenho do protótipo do framework.	78
Tabela 6.1: Estado atual dos módulos do framework.....	82

Lista de Siglas/Acrônimos

3R - *Regeneração, Retemporização e Reformatação*

AI - *Adapted Information*

AI_TSF-O - *Adapted Information Trail Signal Fail Overhead*

AI_TSF-P - *Adapted Information Signal Fail Payload*

API - *Application Programming Interface*

ASON - *Automatically Switching Optical network*

ATM - *Asynchronous Transfer Mode*

AP - *Access Point*

BDI-O - *Backward Defect Indication Overhead*

BDI-P - *Backward Defect Indication Payload*

BER - *Bit Error Ratio*

CBR - *Constant bitrate*

CI - *Characteristic Information*

CP - *Connection Point*

dLOS-P - *Loss of signal payload defect*

dLOS-O - *Loss of signal overhead defect*

DTH - *Digital Transport Hierarchy*

DWDM - *Dense WDM*

FEC - *Forward Error correction*

Gbps - *Gigabit per second*

GFP - *Generic Framing Procedure*

GMPLS - *Generalized Multiprotocol Label Switching*

IDE - *Integrated Development Environment*

IEEE - *Institute of Electrical and Electronics Engineers*

IETF - *Internet Engineering Task Force*

IP - *Internet Protocol*

IPv6 - *IP version 6*

ITU - *International Telecommunication Union*

ITU-T - *ITU Telecommunication Standardization Sector*

LC - *Link Connection*

MAC - *Medium Access Control*

MANET - *Mobile Ad Hoc Network*

MI - *Management Information*

NC - *Network Connection*

NE - *Network Element*

NED - *Network Description*

OAM&P - *Operation, Administration, Maintenance and Provisioning*

OCh - *Optical Channel Layer*

ODU - *Optical Channel Data Unit*

OH - *Overhead*

OMS - *Optical Multiplex Section*

ONE - *Optical Network Element*

OPS - *Optical Physical Section*

OPU - *Optical Channel Payload Unit*

OSC - *Optical Supervisory Channel*

OTH - *Optical Transport Hierarchy*

OTN - *Optical Transport Network*

OTS - *Optical Transmission Section*

OTU - *Optical Channel Transport Unit*

OXC - *Optical Cross-connect*

PLD - *Payload*

PMI - *Payload Missing Indication*

PRBS - *Pseudo-Random Bit Sequence*

RFC - *Request for Comments*

RI - *Remote Information*

RI_BDI-O - *Remote Information_Backward Defect Indication Overhead*

RI_BDI-P - *Remote Information_Backward Defect Indication Payload*

ROADM - *Reconfigurable Optical Add-Drop Multiplexer*

RP - *Remote Point*

SCTP - *Stream Control Transmission Protocol*

SDH - *Synchronous Digital Hierarchy*

Sk - *Sink*

SNC - *SubNetwork Connection*

SNMP - *Simple Network Management Protocol*

So - *Source*

SONET - *Synchronous Optical NETwork*

TCM - *Tandem Connection Monitoring*

TCP - *Termination Connection Point*

TCP - *Transmission Control Protocol*

TDM - *Time Division Multiplexing*

TS - *Tributary Slots*

TTI - *Trail Trace Identifier*

UDP - *User Datagram Protocol*

UML - *Unified Modeling Language*

WDM - *Wavelength-division multiplexing*

Resumo

As novas tecnologias de redes ópticas apresentam um desafio considerável para os pesquisadores envolvidos com a nova geração de redes de alto desempenho, notadamente no que se refere à otimização da infra-estrutura atual de fibras ópticas disponíveis nos backbones.

Neste contexto as simulações se apresentam como alternativas flexíveis para a modelagem e análise de novas idéias e projetos. Além disso, um ambiente uniforme para a simulação de redes OTN é essencial e fornece aos pesquisadores a capacidade de reproduzir e comparar resultados de diferentes frentes de pesquisa.

Este trabalho tem como objetivo apresentar uma proposta de arquitetura para um framework de simulações de redes ópticas de transporte, independente de fabricantes, desenvolvido em código aberto e voltado especificamente para as redes OTN (*Optical Transport Network*). O ambiente de simulação adotado é o OMNeT++.

A proposta de arquitetura é resultado da integração dos trabalhos desenvolvidos pelo grupo de pesquisas em telecomunicações da UFES e está focada principalmente na descrição funcional e interação dos componentes ópticos OTN e as normas e especificações da ITU-T.

As principais contribuições da dissertação são a apresentação da estrutura geral da arquitetura e da modelagem e interação dos seus principais componentes. São apresentados também um protótipo inicial do framework e alguns resultados preliminares de pesquisas que estão utilizando este protótipo. Além disso, é apresentado um módulo de comunicação externa que está em desenvolvimento e que, futuramente, vai agregar muitas funcionalidades ao framework.

Abstract

New Technologies for optical networks present a significant challenge for researchers involved with the new generation of high-performance networking, especially when we are talking about optimizing the current infrastructure available in the optical fiber backbones.

In this context the computer simulations present themselves as flexible alternatives for the modeling and analysis of new ideas and projects. Moreover, a uniform environment for the simulation of OTN networks is essential. That would give researchers the ability to reproduce and compare results from other researchers.

This paper aims to present a proposal for a industry independent and open source framework architecture for simulation of optical transport networks, designed specifically for OTN (Optical Transport Network). The simulation environment adopted is OMNeT++.

The proposed architecture resulted from the integration of works from UFES's telecommunications research group and is mainly focused on the functional description and interaction of the optical components and ITU-T's standards and specifications for OTN.

The main contributions of this thesis are to present the general structure of the architecture, and the modeling and interaction of its main components. Are also presents an initial prototype of the framework and some preliminary results of researchs that are using this prototype. Furthermore, we present an external communication module that is under development and will add many features to the framework in the future.

Capítulo 1: Introdução

O cenário atual das redes de telecomunicações, principalmente das redes ópticas, é de mudanças rápidas. Nesse cenário as novas tecnologias de redes ópticas com roteamento de comprimentos de onda apresentam um desafio considerável para os pesquisadores envolvidos com a nova geração de redes de alto desempenho, notadamente no que se refere à otimização da infra-estrutura atual de fibras ópticas disponíveis nos backbones.

A OTN (*Optical Transport Network*) é uma arquitetura da ITU (*International Telecommunication Union*) que busca estender muitas das funcionalidades do SONET/SDH (*Synchronous Optical Network/ Synchronous Digital Hierarchy*) para as redes DWDM (*Dense Wavelength-division multiplexing*), fornecendo uma padronização para os fabricantes de equipamentos. Recentemente a OTN vem se tornando um requerimento para os elementos ópticos de redes de nova geração (*Next Generation Network*), sendo observados vários aspectos como, por exemplo, a conformidade de novos equipamentos aos padrões propostos nas normas e como utilizar um plano de controle unificado para as redes de transporte OTN, por exemplo, ASON (*Automatic Switching Optical Network*), padronizado pela ITU, e GMPLS (*Generalized Multiprotocol Label Switching*), padronizado pelo IETF (*Internet Engineering Task Force*).

O sucesso dessas novas tecnologias depende, em grande parte, da existência de ferramentas de modelagem e análise para avaliar eficientemente novas idéias e projetos. As simulações se apresentam como uma das opções deste tipo de ferramenta, e, neste cenário, um ambiente uniforme para a simulação de redes OTN torna-se essencial. Esse ambiente permite fornecer aos pesquisadores a capacidade de reproduzir e comparar resultados de diferentes frentes de pesquisa mais facilmente.

Neste contexto é que se insere a presente dissertação, cujo principal objetivo é a apresentação de uma proposta de arquitetura para um framework de simulações de redes ópticas de transporte, independente de fabricantes, desenvolvido em código aberto e voltado especificamente para as redes OTN (*Optical Transport Network*). O framework proposto está sendo desenvolvido no ambiente de simulação de eventos discretos OMNeT++ (Varga & Hornig, An overview of the OMNeT++ simulation environment, 2008).

São especificados o ambiente de simulação adotado, o tipo de abstração utilizada, a estrutura geral da arquitetura e a modelagem dos principais componentes. A arquitetura está focada principalmente na descrição funcional e interação dos componentes ópticos OTN e as

normas e especificações da ITU-T. Esta proposta é resultado da integração dos trabalhos desenvolvidos em (F.F.Ferrari, 2009), (Tessinari, 2009), (Frigini, 2010) e utiliza como base a modelagem proposta em (Favoreto, 2009). É esperado que o framework, além de modelar a camada de transporte OTN segundo as recomendações, atenda três desafios específicos: escalabilidade, flexibilidade e realismo.

Para alcançar os objetivos propostos, o trabalho é desenvolvido de acordo com a seguinte metodologia:

- Estudos relativos às normas da ITU para as redes OTN, principalmente no que tange a arquitetura e descrição funcional dos componentes ópticos;
- Estudo da teoria relacionada com a simulação dos sistemas de comunicações;
- Análise dos principais sistemas de simulação para redes de telecomunicações, verificando as limitações e oportunidades de cada um e visando a escolha de um ambiente como base para a proposta do framework;
- Elaboração da proposta de arquitetura para o framework, considerando a modelagem dos equipamentos, os requisitos pretendidos no projeto e tendo sempre em vista a facilidade de manutenção e o acréscimo de novas funcionalidades;
- Desenvolvimento de uma primeira versão do framework, boa parte sendo resultado da integração de módulos que estavam sendo desenvolvidos em paralelo pelo grupo;
- Validação da estrutura geral da arquitetura e demonstração de alguns resultados experimentais.

Como resultado esperado deste trabalho está a proposta de arquitetura para um framework voltado para a simulação de redes ópticas, que atenda as normas da ITU para redes OTN e que permita modelar, analisar e validar projetos de redes fornecendo, desse modo, um ambiente centralizado e único aos pesquisadores. A principal contribuição da dissertação é a apresentação não só da estrutura geral da arquitetura, como também da modelagem e interação dos seus principais componentes. A arquitetura desenvolvida é resultado da integração e adaptação dos trabalhos sendo desenvolvidos pelo grupo. Além disso, são apresentados alguns resultados preliminares de pesquisas que estão em desenvolvimento, obtidos com a utilização do protótipo do framework.

Para apresentar adequadamente a proposta de framework, este trabalho é organizado nos seguintes capítulos: o capítulo 2 apresenta a arquitetura OTN, suas principais

funcionalidades, assim como define uma série de conceitos utilizados no restante do trabalho. O capítulo 3 apresenta uma análise sucinta do cenário atual dos simuladores de redes, o porquê da escolha do OMNeT++ e as suas principais funcionalidades. O capítulo 4 apresenta a descrição do framework de simulação, sua arquitetura, a modelagem dos principais módulos, os requisitos de projeto e as restrições presentes na estrutura proposta. No capítulo 5 é descrito o estado atual do protótipo do framework, assim como são apresentados alguns resultados experimentais iniciais. Finalmente, no capítulo 6 são descritas as conclusões do trabalho e as idéias sugeridas para trabalhos futuros.

Capítulo 2: Redes OTN

Assim como novas demandas aparecem no mundo das redes de comunicações, novos padrões e tecnologias surgem para supri-las. A OTN é uma tecnologia desenvolvida para ser uma evolução das redes de transporte de núcleo, combinando a tecnologia DWDM com os benefícios da tecnologia SONET/SDH e trazendo a promessa de atender as demandas por maiores capacidades de banda e melhora no desempenho das redes. Este capítulo apresenta uma visão global da OTN, suas vantagens em relação às outras tecnologias, as recomendações em que se baseia, e uma breve descrição da sua estrutura.

2.1 Introdução

Na primeira geração das redes ópticas, essas eram essencialmente usadas para transmissão e aumento da capacidade (Ramaswami & Sivarajan, 2002), já que a fibra óptica é um meio que provê baixa taxa de erro de bits e alta capacidade de transmissão em comparação com outras tecnologias. Projetadas para serviços ponto a ponto e carregando um sinal óptico por fibra, todas as operações de comutação e outras funções inteligentes dessa primeira geração eram feitas eletronicamente, a partir de uma conversão O-E-O em cada elemento de rede.

Como exemplos dessa primeira geração podem ser citados as redes SONET e SDH, tecnologias maduras de transporte digital sobre a camada óptica, estabelecidas em praticamente todos os países do mundo. Quando SONET/SDH foi pensado, no começo da década de 80, o tráfego das telecomunicações era predominantemente de voz. Esse cenário mudou durante os últimos anos e a demanda por tráfego para internet e telefonia móvel tem crescido.

Essa demanda, até agora, tem sido suprida por uma combinação do aumento das taxas de linha (TDM – *Time Division Multiplexing*) e a transmissão de vários comprimentos de onda por uma mesma fibra (WDM - *Wavelength-Division Multiplexing*). Entretanto, conforme as redes evoluem para maiores taxas de transmissão, os limites físicos impostos pelo meio de transporte (fibra óptica) se tornam críticos. Além disso, criar multiplexadores para taxas de transmissão muito altas se confronta com a complexidade do sistema eletrônico para gerar esse sinal e efeitos como a dispersão tendem a piorar à medida que a banda de frequências utilizada aumenta como consequência de maiores taxas de transmissão.

Neste contexto tem crescido a idéia de que as redes ópticas podem fornecer funções mais complexas do que apenas uma conexão ponto-a-ponto, incorporando operações de comutação e roteamento diretamente na camada óptica, assim como o monitoramento e gerenciamento da rede em diversos níveis.

Com o objetivo de delinear essas funções bem como a arquitetura da camada óptica, a ITU criou a OTN (ITU-T, G.872, 2001), (ITU-T, G.709, 2009). Muitos dos conceitos usados na OTN são baseados nas redes SDH/SONET, como a estrutura em camadas, proteção e outras funções de gerenciamento. No entanto, alguns elementos-chaves são adicionados como o gerenciamento de canais ópticos fim a fim no domínio óptico sem a necessidade de conversão para o domínio elétrico e a inclusão do FEC (*Forward Error Correction*) que permite um melhor gerenciamento de erros e linhas ópticas mais extensas.

A OTN oferece também suporte ao DWDM, com a transmissão de dados simultaneamente em múltiplos comprimentos de onda onde cada comprimento se comporta como uma fibra “virtual”. Desse modo, uma única fibra pode prover múltiplos canais de dados ponto a ponto.

Em seguida são detalhados alguns dos principais serviços onde a OTN oferece aprimoramentos sobre as tecnologias anteriores:

- Escalabilidade aprimorada: SONET/SDH possui uma interface que multiplexa fluxos de baixa taxa em fluxos com maiores taxas utilizando uma hierarquia TDM no domínio eletrônico. No entanto, a granularidade de banda que SONET/SDH é capaz de multiplexar está a uma ou duas ordens de magnitude dos sinais WDM, de modo que para carregar esse tipo de sinal tem-se um aumento na quantidade de *overhead* trafegando pela rede. A OTN define um esquema de multiplexação similar ao utilizado nas redes SONET/SDH, porém transporta dados nativamente com taxas de 2,5, 10 e 40 Gbps (*Gigabit per second*) com um *overhead* otimizado para esse tipo de rede.

- Transporte transparente do sinal cliente: Alguns sinais clientes são encapsulados diretamente dentro da OTN, enquanto que em outros tipos de tráfego utiliza-se o GFP (*Generic Framing Procedure*). Desse modo a rede OTN pode transportar de forma transparente qualquer tipo de sinal digital, independente de aspectos específicos de cada cliente e é por isso chamada de encapsulador digital (*digital wrapper*). É essa transparência que permite o transporte de uma grande quantidade de tipos de serviço, entre eles: serviços de comprimento de onda, serviços SONET/SDH e algumas outras tecnologias como Ethernet, Fiber Channel, ATM (*Asynchronous Transfer Mode*), Frame Relay, e IP. Esses sinais são transportados sem que haja terminação do sinal cliente em cada elemento de rede, tornando

possível o transporte sem alterações nas características intrínsecas do sinal original (formato, taxa de bits e *clock*).

- Capacidade de usar FEC: O SONET/SDH possui um mecanismo de FEC, porém utiliza para isso bytes não padronizados do cabeçalho. A OTN possui um campo maior e otimizado para redes OTN (ITU-T, G.709, 2009). Isso permite o alcance, sem regeneração, de maiores distâncias de transmissão, uma vez que o algoritmo é capaz de corrigir uma quantidade maior de erros de bit.

- Melhor nível de TCM (*Tandem Connection Monitoring*): Redes OTN provêm suporte a até seis níveis de TCM independentes (contra um nível suportado por SONET/SDH), tornando possível o monitoramento paralelo de vários segmentos da rede. Uma explicação mais detalhada de como o TCM funciona é apresentada mais adiante, na seção referente à estrutura da OTN.

- OAM&P (*Operation, Administration, Maintenance and Provisioning*): Redes OTN provêm várias funcionalidades de Operação, Administração, Manutenção e Aprovisionamento, muitas dessas herdadas das redes SONET/SDH e expandidas para as camadas elétricas da rede OTN. Essas informações são carregadas fim a fim no frame, permitindo que informações de gerenciamento sejam transmitidas entre diferentes operadores de rede.

2.2 Recomendações ITU referentes às Redes OTN

Uma rede OTN é composta por um conjunto de ONEs (*Optical Network Element*), interconectados por links de fibras ópticas, e capazes de prover transporte, multiplexação, roteamento, gerenciamento, supervisão e mecanismos de sobrevivência para os canais ópticos contendo os sinais cliente. Uma série de recomendações é definida pela ITU a fim de padronizar a OTN. As recomendações citadas a seguir constituem o núcleo da arquitetura.

A recomendação G.805 (ITU-T, G.805, 2000) descreve uma arquitetura funcional genérica das redes de transporte, que serve como base para a modelagem de todas as tecnologias para redes de transporte, como, por exemplo, ATM, SDH e OTN. Uma série de componentes arquiteturais genéricos é definida de modo a permitir a descrição funcional de uma rede de transporte independente da tecnologia utilizada.

Com base na modelagem fornecida pela recomendação G.805 a recomendação G.872 (ITU-T, G.872, 2001) apresenta a arquitetura funcional da OTN, estruturando a rede em camadas do tipo cliente/servidor. Além disso, são descritas as funcionalidades específicas de

cada uma dessas camadas, como funcionam as associações entre elas e o tipo de informação característica que é trafegada.

A recomendação G.709 (ITU-T, G.709, 2009) define as interfaces OTN com base em uma hierarquia de multiplexação digital e óptica, especificando as taxas de bits, a estrutura dos frames, o formato dos cabeçalhos de cada camada e o formato para mapeamento de sinais clientes. É nessa recomendação que se define o funcionamento do FEC (sugestão de uso do código Reed-Solomon) e do encapsulamento digital OTN.

A recomendação G.798 (ITU-T, G.798, 2006) define uma série de blocos funcionais que permitem a modelagem dos elementos da rede OTN. Os blocos funcionais baseiam-se na hierarquia de camadas da recomendação G.872 e nos esquemas de multiplexação apresentados na recomendação G.709, descrevendo as funções de adaptação e de terminação de trilhas das camadas OTN. Nessa recomendação também é detalhado como funciona a correlação de alarmes e defeitos.

A Figura 2.1, retirada de (Iniewski, McCrosky, & Minoli, 2008), ilustra as dependências entre algumas dessas recomendações.

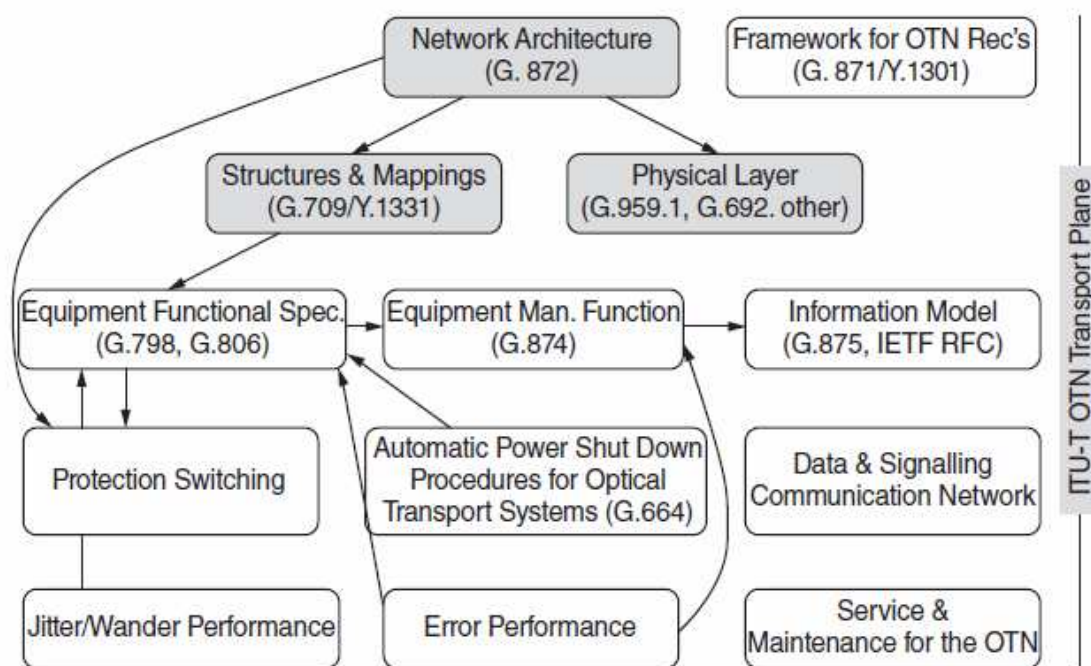


Figura 2.1: Diagrama ilustrando a relação entre algumas normas OTN da ITU-T. (Iniewski, McCrosky, & Minoli, 2008)

2.3 Conceitos Gerais

Como descrito na seção anterior a recomendação G.805 apresenta uma série de componentes arquiteturais, que são utilizados na modelagem das redes de transporte e,

conseqüentemente, na arquitetura OTN. Para melhor entendimento da presente dissertação é necessário conhecimento sobre alguns desses componentes, como eles podem ser usados na modelagem de uma rede de transporte genérica, além da explicação de certos termos utilizados no restante do texto.

Segundo a recomendação G.805, uma rede de transporte é responsável por transferir informações do usuário de um local para outro. As redes de transporte foram assim analisadas para identificar funções genéricas que são independentes das tecnologias de implementação, e, deste modo, conseguir uma descrição abstrata das funcionalidades da rede usando um pequeno número de componentes arquiteturais.

Esses componentes são definidos pelas funções que executam no processamento de informações ou pelas interações que desempenham em relação a outros componentes arquiteturais. Em geral, as funções descritas são definidas e caracterizadas pelo processamento da informação entre suas entradas e saídas. Os componentes arquiteturais são associados de modo a formar os elementos de rede a partir do qual redes reais são construídas. Algumas convenções diagramáticas são usadas para dar suporte a descrição desses componentes, elas são definidas na recomendação e são utilizadas nos exemplos presentes neste capítulo e no restante da dissertação.

A arquitetura funcional genérica de uma rede de transporte é baseada em uma estrutura de camadas no modelo cliente-servidor, onde a camada servidora é responsável por transportar os dados da camada cliente.

O transporte das informações em uma camada é feito através das trilhas (*trails*) e conexões (existem três tipos de conexões que são definidos a seguir). De maneira geral, as conexões são responsáveis por transferir os dados, enquanto que a trilha é responsável por monitorar as informações sendo transferidas no início e no fim do caminho. Sendo assim, uma trilha representa a transferência com monitoração fim-a-fim das informações da camada. Essa transferência pode ser realizada através de diversas conexões, que são as responsáveis por transportar de fato as informações.

Um exemplo de modelagem funcional da camada cliente e servidora pode ser visto na Figura 2.2. Suponha a transmissão de um sinal cliente por uma rede. Essa camada cliente precisa monitorar se a camada servidora está transmitindo os dados corretamente. Essa checagem é feita nas terminações de trilha (*Trail Termination*), representadas pelos triângulos invertidos da figura.

A terminação de trilha no lado fonte da camada cliente insere um cabeçalho daquela camada nos dados que vão ser transportados, enquanto a terminação de trilha no lado destino

retira este cabeçalho conferindo se tudo está como deveria estar. Constituída da carga útil (*payload*) mais o cabeçalho (*overhead*), este “pacote” de dados é chamado de informação característica (CI - *Characteristic Information*), ou seja, é a saída de uma terminação de trilha na origem da transmissão, ou a entrada de uma terminação de trilha no lado destino. Essa é a informação passada da camada cliente para a camada servidora, logo, a camada servidora é responsável por transportar a CI da camada cliente.

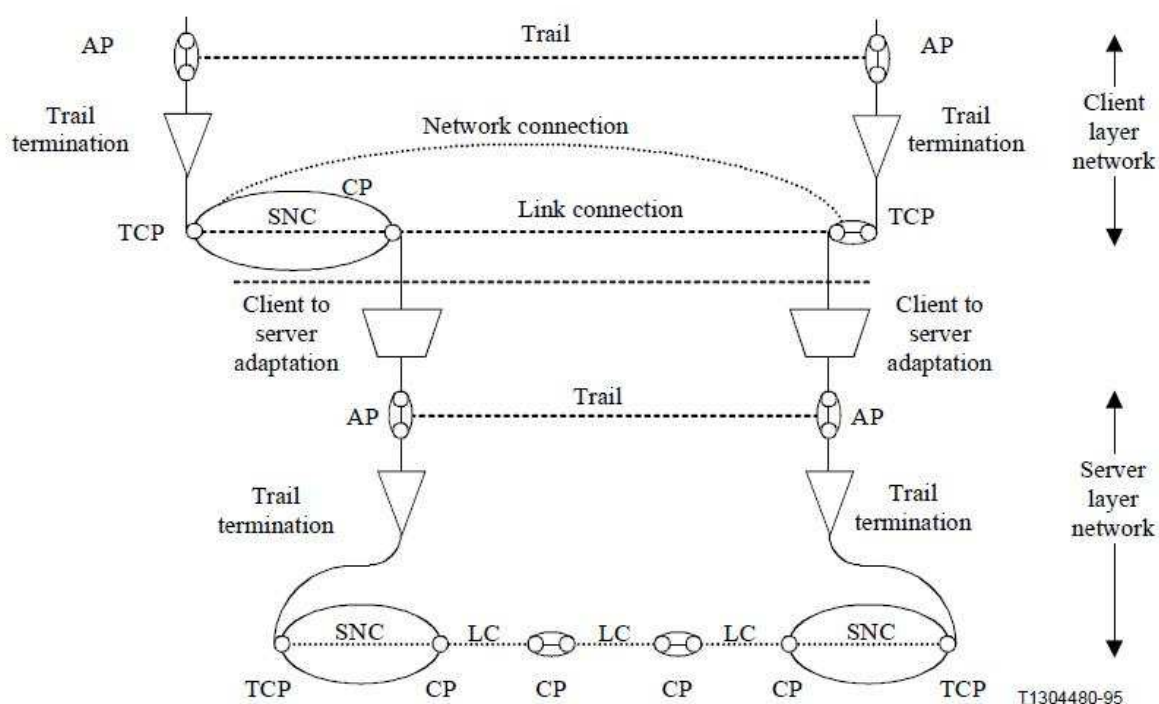


Figura 2.2: Exemplo de modelagem funcional utilizando elementos da recomendação (ITU-T, G.805, 2000).

A informação característica ao sair da terminação de trilha do lado fonte é conduzida para o TCP (*Termination Connection Point*), que está sempre ligado diretamente a uma terminação de trilha. Uma conexão de rede (NC - *Network Connection*) é uma associação entre dois TCP's e tem como função transferir os dados da terminação de trilha. Uma NC ainda é uma abstração funcional e é implementada por conexões de *link* (LC - *Link Connection*), ou conexões de subrede (SNC - *SubNetwork Connection*). Uma LC é o componente mínimo e indivisível de uma camada, enquanto que uma SNC é um particionamento horizontal daquela camada, constituído internamente por LC's e outras sub-redes. O ponto de conexão (CP - *Connection Point*) é o ponto de referência que pode ligar uma SNC a uma LC, ou uma LC a outra LC.

A informação característica da camada cliente (CI) é passada pelo CP à função de adaptação da camada servidora, representada pelo trapézio na Figura 2.2. A função de adaptação é responsável por realizar as principais funções daquela camada, adaptando a informação característica da camada cliente para ser transportada pelas trilhas da camada servidora. A função de adaptação tem como entrada a informação característica da camada cliente (CI), e como saída a informação adaptada (AI - *Adapted Information*) da camada servidora. Finalmente, uma função de adaptação está ligada a terminação de trilha da mesma camada pelo ponto de acesso (AP - *Access Point*), por onde é transferida a informação adaptada.

As terminações de trilha e os outros elementos da camada servidora possuem as mesmas funcionalidades que os elementos equivalentes da camada cliente. Observe na Figura 2.2 que na camada servidora existe diversos LC's, reforçando o fato de que embora para a camada cliente exista apenas uma conexão ponto-a-ponto entre dois CP's, na camada servidora pode existir uma rede muito mais complexa do que a vista pelo cliente. No caso apresentado na Figura 2.2, por exemplo, poderia existir uma camada servidora como a do desenho para cada conexão ponto-a-ponto dos CP's da camada servidora e que neste caso seria a camada cliente dessas novas camadas servidoras.

2.4 Estrutura de Camadas

Baseada na metodologia de modelagem descrita na recomendação G.805, a OTN tem sua arquitetura funcional dividida em camadas. Essas camadas são organizadas em duas hierarquias, uma responsável pela parte óptica, a OTH (Optical Transport Hierarchy) e outra pela parte digital DTH (Digital Transport Hierarchy) (Figura 2.3).

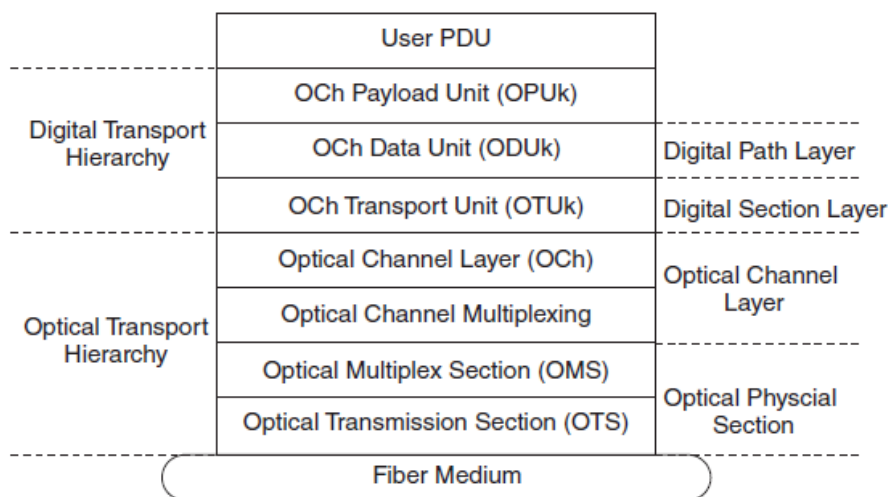


Figura 2.3: Hierarquia OTN. (Iniewski, McCrosky, & Minoli, 2008)

Como pode ser observado na Figura 2.3 a OTN é composta por seis camadas, definidas de acordo com as diferentes funcionalidades providas pela OTN: OPU (*Optical Channel Payload Unit*), ODU (*Optical Channel Data Unit*), OTU (*Optical Channel Transport Unit*), OCh (*Optical Channel Layer*), OMS (*Optical Multiplex Section*) e OTS (*Optical Transmission Section*).

Cada uma dessas camadas e suas respectivas funcionalidades são distribuídas pela rede e ativadas quando atingem seus TCP's, ilustrados na Figura 2.4.

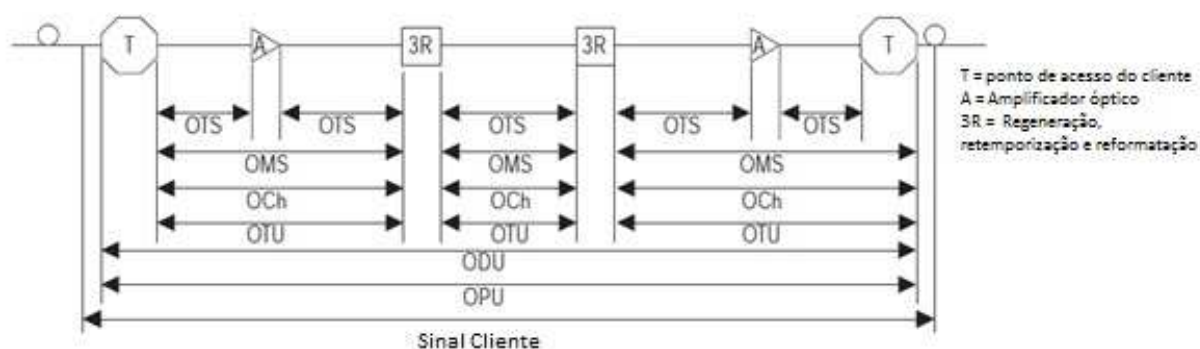


Figura 2.4: TCP's das camadas OTN (EXFO, 2006).

Conforme descrito na seção anterior, as camadas são baseadas no conceito cliente/servidor, de modo que cada camada fornece um serviço para a camada imediatamente acima na hierarquia. Para que o mapeamento do sinal cliente respeite este conceito, eles são encapsulados usando a estrutura ilustrada na Figura 2.5.

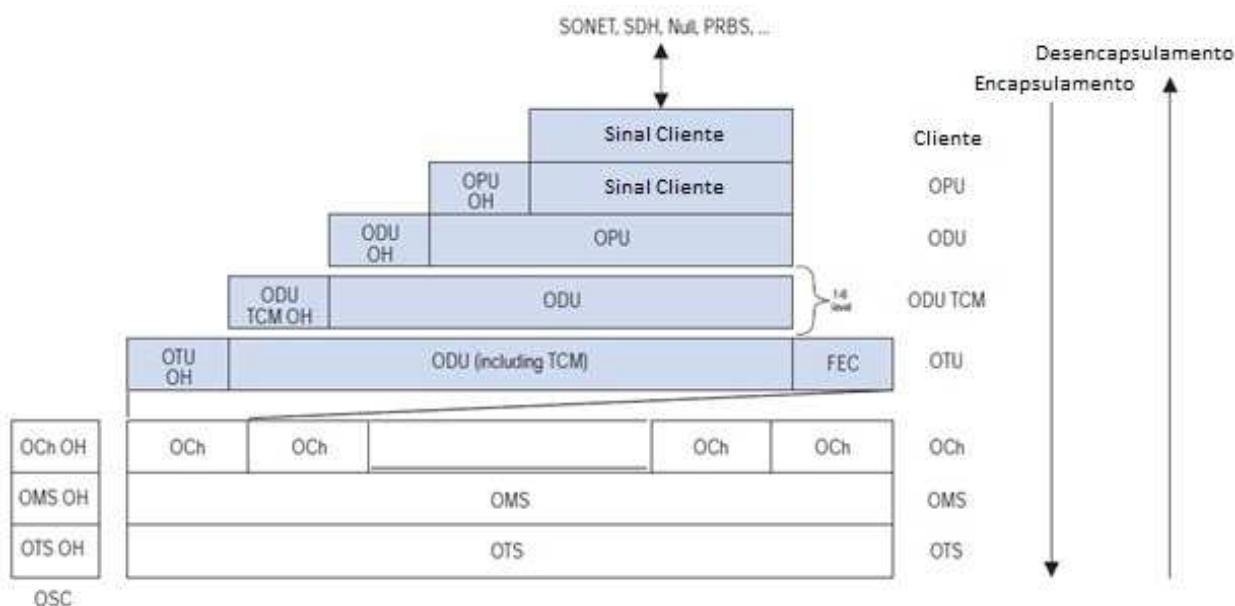


Figura 2.5: Estrutura básica de transporte OTN (EXFO, 2006).

Como ilustrado na Figura 2.5, para se criar um frame OTU, o sinal cliente é primeiro adaptado na camada OPU. Essa adaptação consiste em ajustar a taxa do sinal cliente à taxa da OPU. Seu cabeçalho contém informações que suportam a adaptação do sinal cliente e permitem operações de justificação para ajuste da taxa de transmissão. Uma vez adaptado, a OPU é mapeada em uma ODU. A ODU mapeia a OPU e adiciona o cabeçalho necessário para garantir supervisão fim-a-fim e a monitoração de até seis níveis de conexões tandem. A ODU também realiza a multiplexação digital e se apresenta como um caminho digital fim-a-fim para o sinal cliente.

Finalmente, a ODU é mapeada na OTU, que provê alinhamento de quadros, assim como monitoramento de sessões e o FEC. Segundo a G.709, uma estrutura de frame OTU completa (OPU, ODU e OTU) é capaz de fornecer funcionalidades de OAM&P através das informações presentes nos cabeçalhos.

Seguindo a estrutura apresentada na Figura 2.5, OTU_k's ($k = 1,2,3$ e 4 , explicados a seguir) são transportados usando a OCh, camada onde cada canal OTU recebe um comprimento de onda específico. Cabe ressaltar que as terminações de trilha OTU coincidem com as terminações de trilha OCh, uma vez que é nesse ponto que acontecem as conversões digital/óptico e óptico/digital. Em seguida, vários canais podem ser mapeados na OMS, responsável por multiplexar os comprimentos de onda gerados na OCh, e então transportados pela camada OTS que fornece um caminho óptico entre dois NEs (*Network Element*). Para cada uma das camadas OCh, OMS e OTS há um cabeçalho próprio, com propósitos de gerenciamento no nível óptico. Esses cabeçalhos são transportados em um canal separado do *payload* (*out-of-band*) chamado de canal supervisor ou OSC (*Optical Supervisory Channel*).

Conforme se observa nos parágrafos anteriores, as camadas digitais são geralmente representadas com a abreviação do seu nome acrescidos da letra k , como em OPU_k, ODU_k e OTU_k. A letra k representa o nível na hierarquia digital e, por conseguinte, as diferentes taxas de transmissão suportadas. Na OTN, conforme as taxas aumentam o tamanho dos *frames* OTU permanece o mesmo. Maiores taxas de transmissão são atingidas simplesmente aumentando a taxa de transmissão do *frame* OTU. Essa é uma grande diferença em relação ao conceito tradicional usado nas redes de comunicação (SONET/SDH), com a transmissão acontecendo a cada 125 μ sec. Na tabela 2-1 são mostradas as aproximações para algumas das taxas nominais da OTN.

Tabela 2.1: Taxa nominal de bits definida na (ITU-T, G.709, 2009).

Nível na hierarquia digital	Cliente OPU payload (kbit/s)	Taxa de bits ODU (kbit/s)	Frame OTU Taxa aproximada (kbit/s)
0	1 238 954.310	1 244 160	Sinais ODU1, 2, 3 ou 4
1	2.488.320	2.498.775,126	2.666.057,143
2	9 995 276.962	10 037 273.924	10.709.225,316
3	40 150 519.322	40 319 218.983	43.018.413,559
4	104 355 975.330	104 794 445.815	111.809.973,568
ODUflex para sinais clientes CBR	Taxa de bits do sinal cliente	$239/238 \times$ Taxa de bits do sinal cliente	Sinais ODU2, 3 e 4
ODUflex para sinais clients mapeados em GFP-F	$238/239 \times$ taxa de sinal ODUflex	Taxa de bits configurável	Sinais ODU2, 3 e 4
2e	10 356 012.658	10 399 525.316	ODU3 e ODU4

Com essa definição de níveis da hierarquia digital, é possível multiplexar sinais de diferentes taxas de *bits* em um único *frame* OTUk. Um exemplo de multiplexação pode ser observado na Figura 2.6, onde uma OPU1 é mapeada em uma ODU1; quatro ODU1's são multiplexadas em uma ODU2, que é então encapsulada em uma OTU2, criando o FEC correspondente a essa configuração.

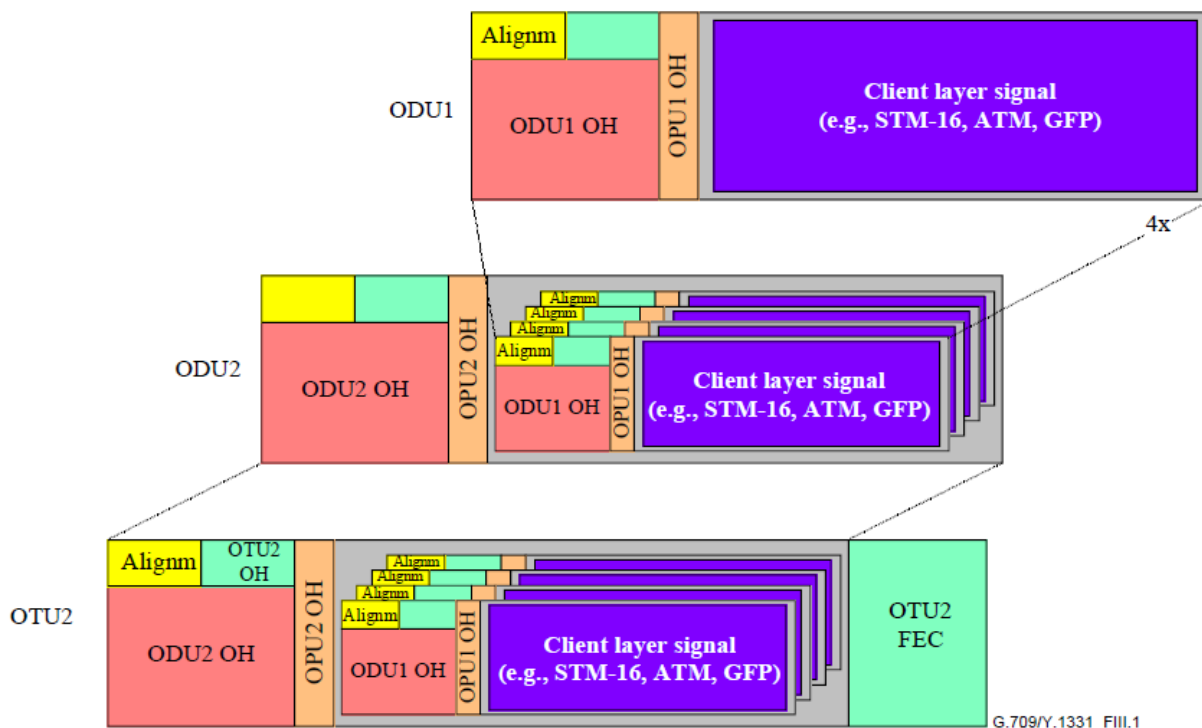


Figura 2.6: Exemplo de multiplexação em redes OTN (ITU-T, G.709, 2009)

A especificação inicial da (ITU-T, G.709, 2003) introduzia três tipos de ODU (i.e. ODU1, ODU2 e ODU3) assim como especificado na Tabela 2.1. Conforme explicado esses ODU_k podem ser mapeados diretamente em um OTU_k (Optical Channel Transport Unit-k, k=1,2 ou 3) ou mapeados em um ou mais *Tributary Slots* (Slots Tributários) de um OPU_j (com k<j).

A última revisão da recomendação (ITU-T, G.709, 2009) introduziu novas características à OTN dando origem aos seguintes tipos de ODU: ODU0, ODU4, ODU2e, ODU3e1 e ODU3e2. A nova hierarquia de multiplexação define um LO (*Lower Order*) ODU_j que pode ser mapeado diretamente em um OTU_j, ou multiplexado em um HO (*High Order*) ODU_k (com j<k).

No caso de um LO ODU_j mapeado em um OTU_j, os seguintes mapeamentos são definidos: ODU1 em OTU1, ODU2 em OTU2, ODU3 em OTU3, ODU4 em OTU4 e ODU2e em OTU2e. No caso da multiplexação de um LO ODU_j em um HO ODU_k, uma nova granularidade de *Tributary Slot* é introduzida (i.e. 1,25Gbps) e os seguintes modos de multiplexação estão definidos:

- ODU0 em ODU1 (com granularidade de TS de 1,25Gbps);
- ODU0, ODU1, ODUflex em ODU2 (com granularidade de TS de 1,25Gbps);
- ODU1 em ODU2 (com granularidade de TS de 2,5Gbps);
- ODU0, ODU1, ODU2, ODU2e e ODUflex em ODU3 (com granularidade de TS de 1,25Gbps);
- ODU1, ODU2 em ODU3 (com granularidade de TS de 2,5Gbps);
- ODU0, ODU1, ODU2, ODU2e, ODU3 e ODUflex em ODU4 (com granularidade de TS de 1,25Gbps);
- ODU2e em ODU3e1 (com granularidade de TS de 2,5Gbps);
- ODU2e em ODU3e2 (com granularidade de TS de 1,25Gbps).

Similarmente ao que acontece nas camadas digitais com o caractere k, nas camadas ópticas o caractere n indica o número de comprimentos de onda que estão sendo carregados na CI daquela camada.

Outra funcionalidade oferecida pela OTN é o TCM (*Tandem Connection Monitoring*), que possibilita a gerência e monitoração da qualidade do sinal através de múltiplas redes. A funcionalidade TCM definida na OTN é capaz de monitorar até seis conexões tandem independentes. Conforme pode ser observado na Figura 2.7, várias topologias de monitoração estão disponíveis.

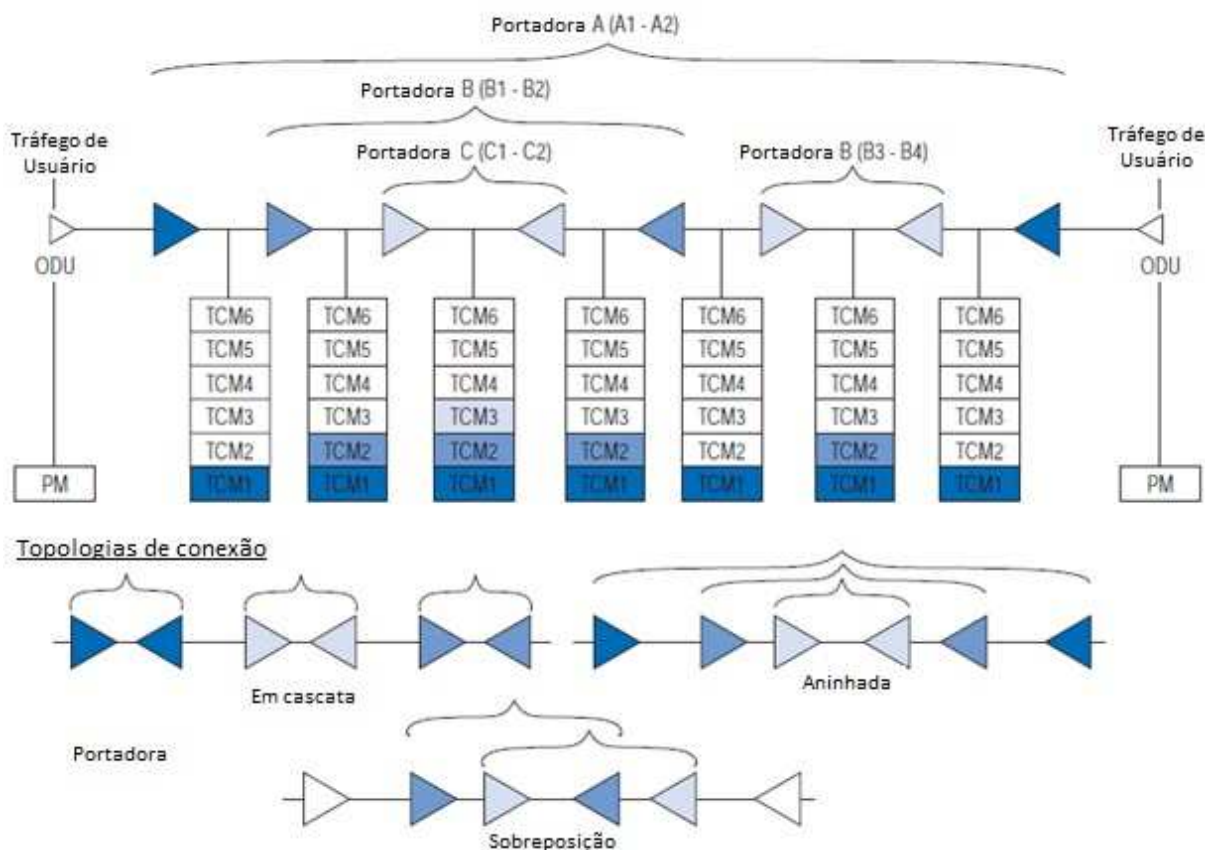


Figura 2.7: Monitoramento de conexões tandem (ITU-T, G.709, 2009).

Na recomendação (ITU-T, G.798, 2006) são especificados os blocos básicos usados para descrever os equipamentos OTN e o conjunto de regras pelas quais esses blocos podem ser combinados. A biblioteca de componentes especificada na recomendação contém todos os blocos funcionais necessários para descrever completamente a estrutura funcional genérica da OTN. Desse modo, para ser compatível com essa recomendação as funcionalidades de um equipamento OTN precisam ser descritas de acordo com os blocos funcionais especificados e as interconexões precisam seguir as regras definidas.

Nessa recomendação, as funcionalidades das camadas descritas anteriormente são organizadas de acordo com o lado de transmissão: lado de origem (*source*), representado por So, ou lado destino (*sink*), representado por Sk, sendo divididas de acordo com a função dentro da camada: conexão, terminação e adaptação.

A função de conexão não está presente em todas as camadas só aparecendo nas camadas OCh e ODU. A função de conexão é representada pelo nome da camada acrescido da letra C (neste caso, OCh_C e ODUk_C). As funções de terminação são representadas pela letra TT (OCh_TT ou OMSn_TT, por exemplo) e as de adaptação por A, sendo que neste

caso também se insere a camada de onde a informação está sendo adaptada, por exemplo, OMSn/OCh_A para a adaptação entre a AI da OMS e a CI da OCh.

A Figura 2.8 mostra uma representação da camada OTS com suas funções de adaptação e terminação, observando as convenções diagramáticas definidas na recomendação G.805.

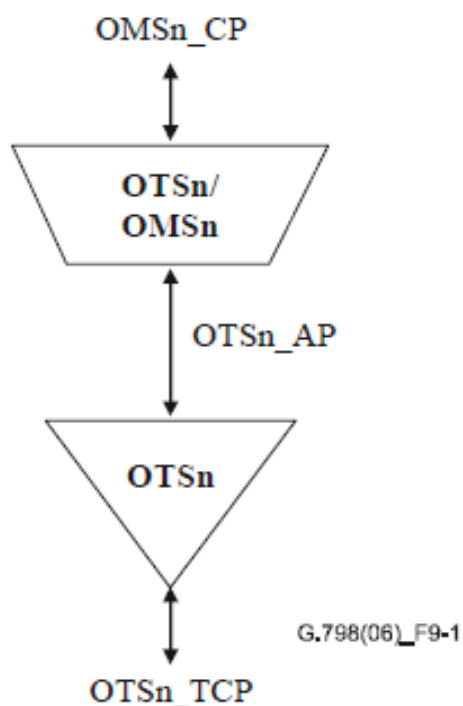


Figura 2.8: Funções da adaptação e terminação da OTS (ITU-T, G.798, 2006).

Na Figura 2.9 está a representação das funções de terminação da OTS, tanto do lado origem quanto destino. Através do OTSn_RP (*OTSn Remote Point*) são enviados o conjunto de sinais RI (*Remote Information*) do lado *sink* para o lado *source*.

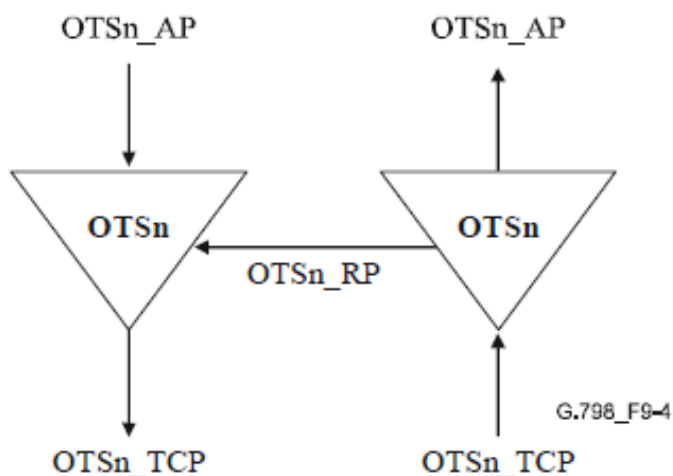


Figura 2.9: Funções de terminação da OTS (ITU-T, G.798, 2006).

Finalmente, na Figura 2.10 vê-se o detalhamento das funções de adaptação relativas a fonte e origem entre as camadas OTS e OMS:

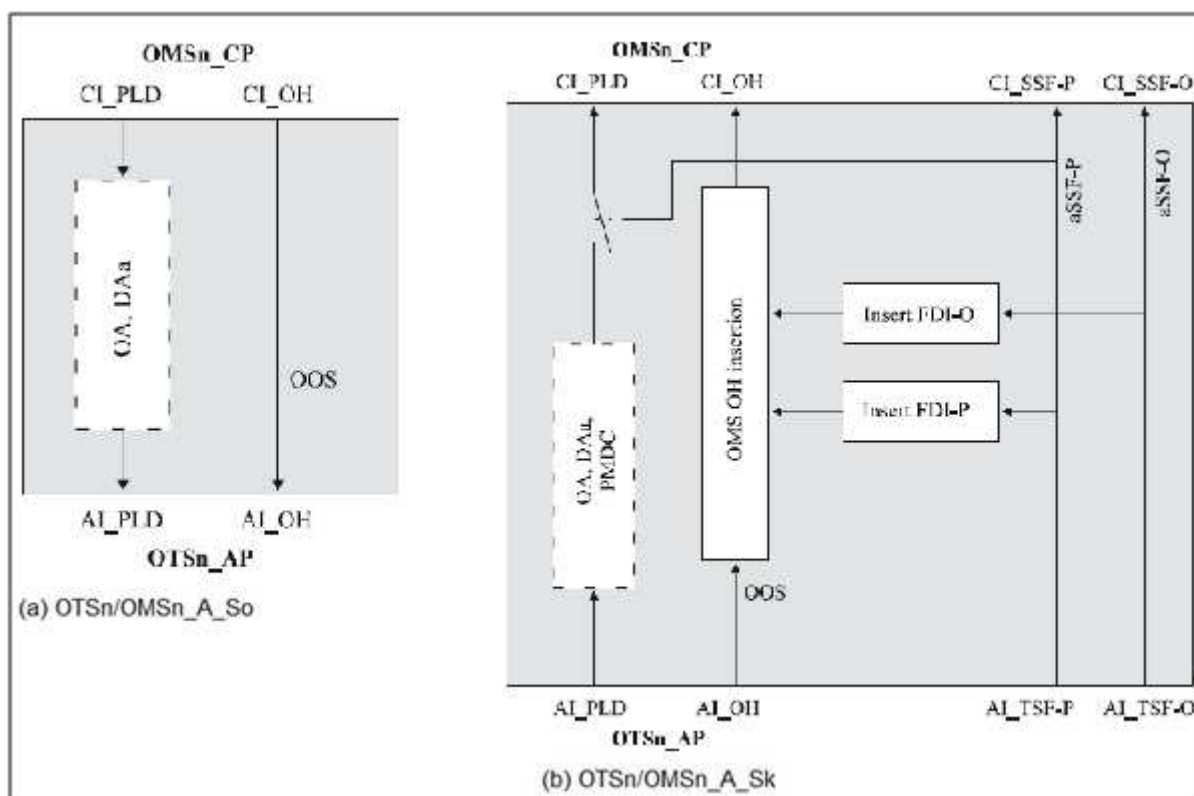


Figura 2.10: Funções da adaptação entre as camadas OTS e OMS (a) OTSn/OMSn_A_So, (b) OTSn/OMSn_A_Sk (ITU-T, G.798, 2006).

Além da funcionalidade completa, descrita até esse momento, a arquitetura OTN define também as especificações para um tipo de funcionalidade reduzida que simplifica as camadas da parte óptica, conforme a Figura 2.11. Em vez de uma estrutura óptica com as camadas OCh, OMS e OTS, existem apenas as camadas OCH e OPS (*Optical Physical Section*) ou dependendo do caso só a OPS. Na funcionalidade reduzida não existe cabeçalho das camadas ópticas e, portanto, não existe canal de serviço nem os mecanismos de monitoramento e isolamento de falhas da camada óptica como definido anteriormente. Além disso, segundo a recomendação G.798, a OTN com funcionalidade reduzida não possui a função de conexão da camada OCh (OCh_C) e, portanto, não possui capacidade de roteamento de comprimentos de ondas.

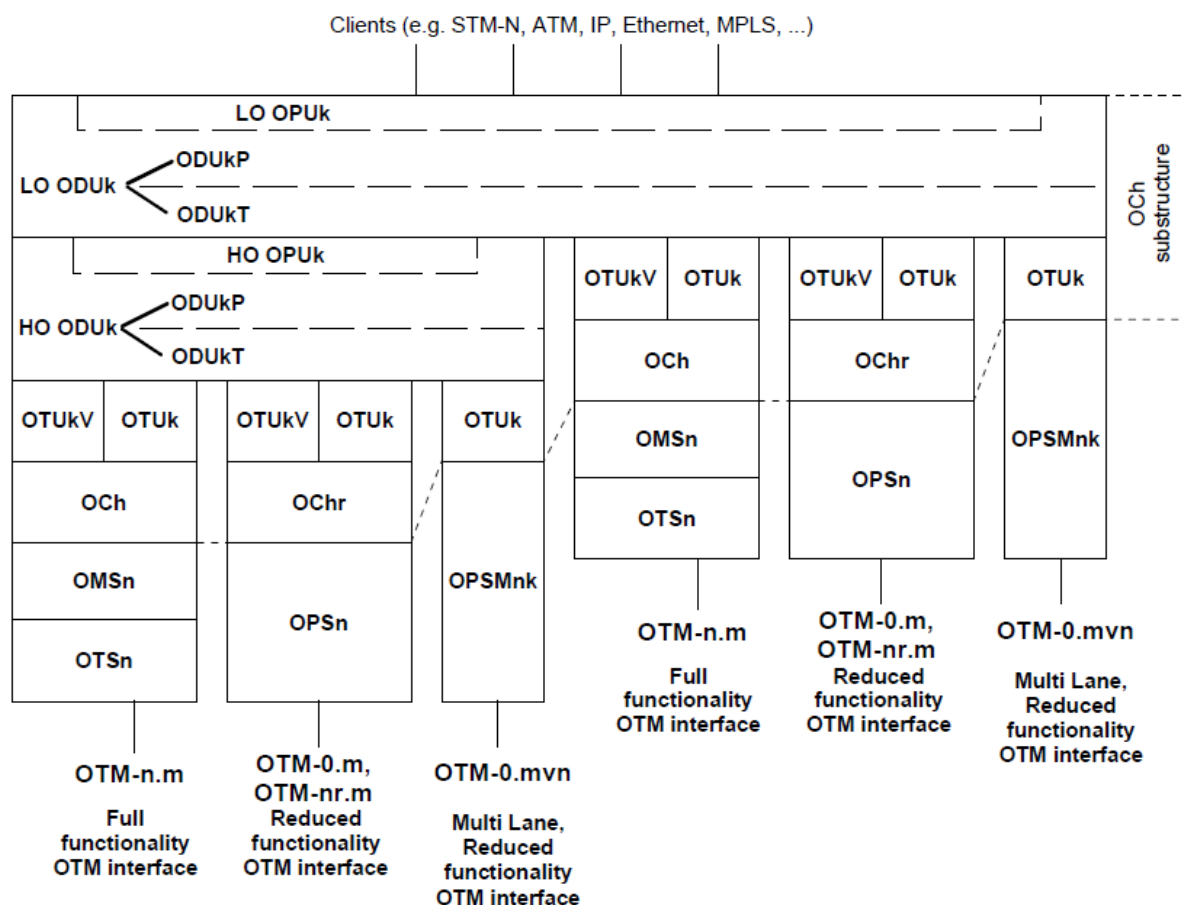


Figura 2.11: Estrutura das interfaces OTN. (ITU-T, G.709, 2009)

As interfaces identificadas por “*multi lane*” na Figura 2.11 surgiram quando o IEEE 802.3 definiu uma interface paralela para as tecnologias 40GBASE-R e 100GBASE-R (*Gigabit Ethernet*) e a ITU-T escolheu definir interfaces paralelas correspondentes para as OTU3 e OTU4. O IEEE 802.3 é uma coleção de padrões do IEEE definindo a camada física e a parte relativa à subcamada MAC (*Media Access Control*) da camada de enlace de dados da Ethernet cabeada. Essas interfaces são identificadas na OTN por “*multi-lane OTN-0.mvn*” e poderiam então ser usadas em aplicações que se beneficiariam do baixo custo dos módulos PHY Ethernet. Dois sinais para a interface OTM-0.mvn são definidos, cada um carregando um sinal OTUk dividido em 4 linhas ópticas: OTM-0.3v4 e OTM-0.4v4, carregando um OTU3 e OTU4 respectivamente (divididos entre as 4 linhas). Os formatos dos sinais OTN foram adicionados à (ITU-T, G.798, 2006), e as especificações da camada física incluídas nas recomendações (ITU-T, G.695, 2009) e (ITU-T, G.959.1, 2009) no final de 2009. Outros modos para transporte de sinais Gigabit Ethernet utilizando a OTN podem ser encontrados na recomendação (ITU-T, G-Sup.43, 2008).

2.5 Considerações Finais

A OTN foi criada com a intenção de combinar os benefícios da tecnologia SONET/SDH com as capacidades de expansão de banda oferecidas pelo DWDM. Como descrito neste capítulo a OTN é definida por um conjunto de especificações começando pela recomendação ITU-T G.872, que descreve a arquitetura da rede, passando pela recomendação G.709, que foca em estruturas, interfaces e mapeamentos – o que inclui o FEC e a capacidade de transportar de forma transparente qualquer tipo de sinal digital, chegando finalmente na recomendação G.798 que define os blocos funcionais usados na modelagem dos elementos da rede.

Este capítulo apresentou, de forma resumida, os principais pontos da arquitetura OTN, suas principais funcionalidades e aprimoramentos em relação a outras tecnologias e foram repassadas as recomendações da ITU-T que definem a tecnologia. Desse modo, espera-se que o leitor tenha uma base para o entendimento do framework que é descrito no restante da dissertação.

Capítulo 3: Simulações

Tradicionalmente pesquisadores têm recorrido a uma variedade de ferramentas e técnicas para entender e avaliar experimentalmente as redes de telecomunicações. Em geral, existem três formas para avaliar o funcionamento e desempenho dessas redes: métodos analíticos, redes experimentais e a simulação computacional.

As redes experimentais são usadas principalmente em operadoras, para testar e validar novos equipamentos e tecnologias. Apesar de produzirem excelentes resultados, já que os testes são feitos na rede real, esse tipo de experimento possui alto custo e normalmente não pode ser utilizado na fase de pesquisa e desenvolvimento já que dificilmente existem equipamentos compatíveis com as tecnologias sendo desenvolvidas e pesquisadas. As redes experimentais também costumam ser dependentes dos detalhes de implementação de cada fabricante e desse modo não são a melhor alternativa para se estudar o desempenho e funcionamento dos padrões e recomendações. Os métodos analíticos, se comparados às redes experimentais, possuem baixo custo, entretanto, para a maioria das redes práticas os modelos são complexos, inflexíveis e muitas vezes intratáveis.

A simulação normalmente é a solução mais prática das alternativas mencionadas (Alberti, 2003), também podendo incorporar tanto métodos analíticos quanto características de sistemas reais. Além disso, facilita o projeto e a comparação de algoritmos já que possibilita o modelamento realista de fontes de tráfego e permite a análise com diferentes níveis de detalhamento e escalas de tempo para observação. Nas simulações também podem ser observados sistemas de grande escala que dificilmente poderiam ser avaliados experimentalmente ou tratados analiticamente. A simulação se mostra, portanto, uma alternativa flexível para a análise de desempenho de sistemas de telecomunicações provendo um método com boa relação custo benefício para a análise e estudo do comportamento dos modelos propostos.

Este capítulo inicia com um breve resumo dos principais termos e conceitos da área de simulação, em seguida apresenta os trabalhos relacionados na área de simulação de sistemas de comunicações, discutindo algumas de suas principais características e citando a presente dissertação. Em seguida são apresentados os principais ambientes de simulação disponíveis, como foi feita a escolha pelo sistema que é usado no *framework*, o OMNeT++, terminando com uma descrição resumida do seu funcionamento.

3.1 Simulação computacional

Antes de prosseguir, torna-se necessário estabelecer alguns termos e conceitos a fim de permitir o melhor entendimento deste trabalho.

Em computação, “simulação” consiste em empregar técnicas que permitem reproduzir o funcionamento de algum tipo de operação ou processo do mundo real. Para isso é necessário que a situação seja modelada computacionalmente de modo a retratar a situação real que se deseja observar. Assim é possível entender seu comportamento e/ou avaliar estratégias para sua melhor operação.

De acordo com a taxonomia proposta por (Nance, 1993), as simulações computacionais podem ser classificadas basicamente em três categorias: simulação Monte Carlo, simulação contínua e simulação de eventos discretos. A simulação Monte Carlo utiliza modelos de incerteza onde a representação do tempo não é necessária. “O termo originalmente se referia a uma situação em que um problema não determinístico difícil é resolvido através da invenção de um processo estocástico que satisfaz as relações do problema determinístico” (Nance, 1993). Nesta, inexistente uma representação explícita do tempo.

Por outro lado, tanto a simulação contínua como a simulação de eventos discretos levam em conta o tempo. A primeira é utilizada para modelar sistemas cujo estado varia continuamente no tempo, como no exemplo de uma xícara de chá aquecendo ou o comportamento de um sinal óptico na fibra. Esse tipo de simulação se utiliza de equações diferenciais para o cálculo das mudanças das variáveis de estado ao longo do tempo. Por outro lado, as simulações de eventos discretos são utilizadas para modelar sistemas que mudam o seu estado em pontos discretos no tempo a partir da ocorrência de eventos, por exemplo, uma manufatura ou uma rede de pacotes. Esses dois tipos de sistemas estão ilustrados pela Figura 3.1:

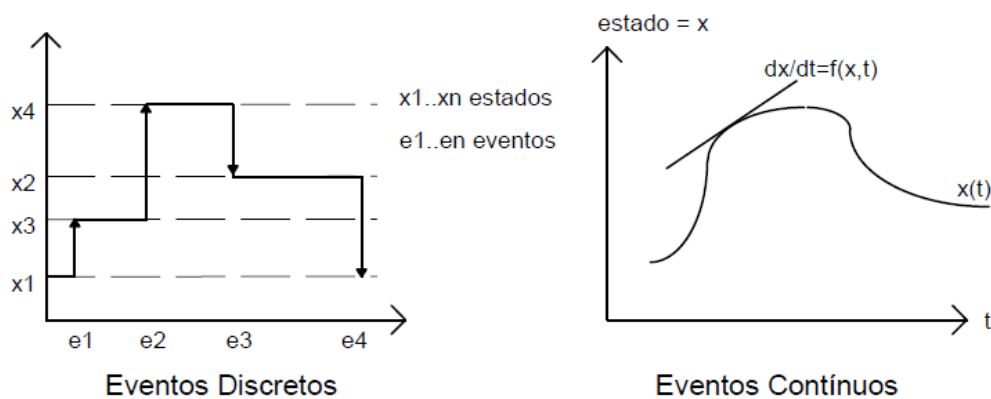


Figura 3.1:Eventos discretos x contínuos. (Chwif, 1999)

A maioria dos simuladores de redes de comunicação utiliza a simulação de eventos discretos (Alberti, 2003). Nessa categoria de simulação as execuções são agendadas na forma de eventos que são armazenados em uma fila com prioridades. Cada evento possui um tempo de execução. Um gerenciador de eventos (presente no *kernel* do simulador) retira o evento de maior prioridade (menor tempo de execução) que se encontra na fila e o executa. Vários eventos podem ser agendados para o mesmo tempo de execução e, neste caso, o *kernel* do simulador deve definir qual a ordem de execução. Toda vez que um evento é retirado da fila, o tempo global de simulação é avançado para o tempo de execução deste evento. A simulação prossegue até que não haja mais eventos para serem executados, ou até que o tempo final de simulação seja alcançado.

Modelagem é o processo de construir modelos de sistemas e de dispositivos reais de forma que estes representem o mais fielmente possível o funcionamento real diante de certo conjunto de situações. O principal desafio é simplificar o modelo suficientemente para que as simulações executem num tempo realístico, mas sem perder informações a ponto de comprometer a validade dos dados sendo observados. O modelamento das redes de comunicação é uma tarefa difícil, uma vez que é necessário não apenas o conhecimento do ambiente de simulação para o qual os modelos estão sendo desenvolvidos, mas também o conhecimento da teoria por trás das redes sendo estudadas, dos detalhes dos componentes a serem modelados e dos resultados a serem obtidos.

Outra classificação pode ser feita em relação às variáveis com que os modelos de simulação trabalham. Os modelos de simulação que possuem uma ou mais variáveis aleatórias como entrada, e que levam a saídas aleatórias são chamados de Modelos Estocásticos, enquanto que os Modelos Determinísticos trabalham somente com variáveis do tipo não probabilísticas, ou seja, para um conjunto conhecido de dados de entrada tem-se um único conjunto de resultados de saída.

Quando se fala de modelos de simulação a distinção entre simulação e emulação deve ser bem definida. A emulação é uma simulação específica em que o propósito é imitar completamente o sistema ou função original, representando na totalidade os detalhes envolvidos, em contraste com a maioria dos casos de simulação onde o propósito é modelar apenas os aspectos importantes relativos ao que se pretende observar. Ou seja, na simulação não há necessidade de se representar todas as funções envolvidas, mas somente aquelas cujos detalhes são significativos com respeito às observações de interesse (Alberti, 2003).

A emulação de um determinado componente pode ser construída com a inclusão de um elemento no sistema de modo que este altere o fluxo de dados interno, imitando o

comportamento que seria causado caso o componente real estivesse presente. Esse elemento pode ser tanto um computador de propósito geral executando um software de emulação, quanto um hardware específico de emulação. Desse modo, a emulação de uma rede de comunicação, por exemplo, pode parecer para sistemas externos como se fosse uma rede real. Sistemas externos que sejam conectados a esse ambiente vão se comportar como se estivessem conectados a uma rede real. Do ponto de vista prático, é impossível incorporar os detalhes de todas as funções executadas por um único nó de uma rede óptica e esperar que um simples computador emule tal modelo, principalmente devido às altas taxas de transferência de dados das redes ópticas.

Quanto aos níveis de modelagem e simplificação que podem ser utilizados, os sistemas de comunicação podem ser enxergados em vários níveis de abstração. Para o tipo de simulação tratada nesta dissertação a distinção entre dois níveis específicos é importante: O nível físico e o nível de rede (Favoreto, 2009).

No nível físico têm-se os componentes físicos responsáveis pela troca de dados entre dois pontos e que formam, desse jeito, os *links* de comunicação. Simulações ópticas que tratam esse nível de abstração estão preocupadas em validar o sistema de geração do sinal óptico, assim como o seu comportamento durante a *trasmissão* pela fibra. A simulação é focada na estimativa de características do enlace, como nível de ruído, queda da potência do sinal ou dispersão. Nesse nível de abstração, são necessários modelos bem detalhados de modo a retratarem os efeitos físicos de interesse. Nesse caso são usados simuladores baseados em modelos matemáticos que tentam simular o comportamento do sinal óptico o mais próximo possível do real. Com o aumento do número de comprimentos de onda presentes na fibra a representação detalhada desses sinais ópticos pode se tornar rapidamente impraticável, dado o grande número de cálculos necessários, e simplificações podem se tornar necessárias.

Já no caso das simulações no nível de rede de abstração, em geral, são utilizados simuladores orientados a eventos já que se está preocupado com informações relacionadas ao desempenho do sistema, como o *“throughput”* da rede e a utilização dos enlaces, assim como atribuição de comprimentos de onda e problemas relacionados. Nesse nível de simulação os sinais ópticos são caracterizados exclusivamente pelo comprimento de onda e potência média e não pelo formato de onda como normalmente é feito nas simulações de nível físico. Essas simplificações resultam em um grande aumento na velocidade de execução se comparados com o nível físico e permitem, portanto, que redes maiores e mais complexas possam ser simuladas.

3.2 Trabalhos relacionados

Em geral, trabalhos similares que tratam da simulação de redes ópticas estão preocupados com problemas específicos de redes ópticas de transporte e isso se reflete na escolha dos simuladores e da modelagem utilizada. Por exemplo, em (Bueno-delgado, Veiga-gontán, Pavón-mariño, & García-haro, 2005) é apresentada uma ferramenta de simulação, o oPASS, desenvolvida especificamente para a avaliação de arquiteturas “*Optical Packet Switching*” (OPS). Apesar de ser voltada para ambientes ópticos, a ferramenta permite a avaliação de parâmetros clássicos de redes de pacotes como probabilidade de perda e distribuição do atraso de pacotes. Em (Kozlovsky, Berceci, & Kozlovsky, 2007) é apresentado o NSOSS (*Non-synchronized Optical Switch Simulator*), um simulador que acompanha o comportamento de um modelo de *switch* DWDM multi-portas/multi-comprimentos de ondas tendo como objetivo observar o desempenho de cada uma das diferentes estratégias, arquiteturas e soluções de *buffers* propostas especificamente para essa tecnologia.

Em ambos os casos o ambiente de simulação utilizado é o OMNeT++, mas os modelos desenvolvidos são tão específicos que dificilmente poderiam ser aproveitados na simulação de problemas relacionados. De um modo geral, os autores simulam a rede OTN sem considerar os padrões da ITU, já que o objetivo principal é a validação de outros componentes, como, por exemplo, esquemas de roteamento, protocolos específicos ou propostas para plano de controle e gerência (Favoreto, 2009).

Os dois principais diferenciais desta dissertação em relação a trabalhos relacionados encontrados na literatura são: primeiro, o foco do *framework* proposto nas redes OTN e, conseqüentemente, nas especificações da ITU que as definem. Segundo, a modelagem do sistema está no nível de rede e não no nível físico. As informações relacionadas ao nível físico são usadas como entrada para a simulação e devem ser coletadas num passo de simulação complementar e anterior, realizado em ferramentas específicas para esse propósito.

Um dos principais objetivos do *framework* proposto nesta dissertação é ser utilizado na simulação e avaliação de novas tecnologias e na interação dessas com as redes OTN. Assim, o *framework* é planejado de forma a ser o mais genérico possível e nenhuma tecnologia de implementação física é utilizada, de modo a permitir que os usuários do *framework* possam escolher as tecnologias que mais se adéquem aos seus experimentos, ou utilizar as modelagens genéricas presentes tanto no *framework* quanto no OMNeT++, com a garantia de que as normas OTN especificadas pela ITU estão sendo seguidas. Dessa maneira

espera-se que a consolidação das ideias testadas tenha uma grande probabilidade de sucesso quando implementadas em equipamentos OTN reais que sejam construídos de acordo com os padrões internacionais.

3.3 Ambientes de simulação

Vários ambientes de simulação com capacidade para simulação de redes de comunicação estão disponíveis atualmente. Desse modo o *framework* não precisou ser construído do zero, e uma variedade de funcionalidades, necessárias a um cenário de simulação, foram aproveitadas do ambiente de simulação utilizado como base.

Para isso foi necessária uma avaliação dos principais ambientes para simulação de redes. Escolher o simulador adequado de acordo com as próprias necessidades não é trivial. Atualmente existe uma variedade de opções tanto comerciais quanto livres, com diferentes potencialidades, fraquezas, foco e popularidade. Entre eles ns2 (Site do ns-2), ns3 (site do ns-3), OMNeT++ (Site do OMNeT++), J-Sim (Site do J-Sim), GLASS (Site do GLASS) e OPNET (Site do OPNET).

Essa lista contempla tanto ferramentas com propósito geral, que podem em princípio ser utilizadas para simular qualquer tipo de sistema, quanto ferramentas de simulação orientadas às redes de comunicação.

O ns-2 (*Network Simulator*) é um simulador de eventos discretos voltado para pesquisas em redes de computadores. O simulador vem sendo desenvolvido desde 1989 e existem muitas contribuições de modelos de simulação disponíveis para uso. O simulador também é “*open source*” e muitos usuários contribuem para o desenvolvimento da ferramenta (Site do ns-2).

O núcleo do simulador, assim como a maior parte dos modelos de protocolo de rede, são escritos em C++, e o resto na linguagem de programação OTcl. O simulador “tenta” separar a parte de dados da parte de controle, usando o C++ para dados (processamento de pacotes, núcleo do simulador, etc.) e o OTcl para controle (configuração do cenário de simulação, ações periódicas ou sinalizadas, etc.). No entanto, na prática, muitos modelos de protocolos estão em OTcl e não existe uma divisão muito clara entre o C++ e a OTcl (Begg, Liu, Pawlikowski, Perera, & Sirisena, 2006). Além disso, os protocolos e tecnologias no ns-2, em geral, são desenvolvidos para uso isolado, para resolução de problemas específicos. Integrá-los, normalmente, é uma tarefa árdua (Dias, Dias, Abelem, & Dias, 2008).

Uma nova versão do ns, o ns-3, está em desenvolvimento. O ns-3 tem seu núcleo escrito em C++ e Python e foi desenvolvido para resolver vários dos problemas do seu

antecessor. Entre eles o problema da separação do núcleo do simulador, dos modelos de simulação e os cenários sendo simulados (site do ns-3). No momento da avaliação dos simuladores, para fins desta dissertação, o ns-3 ainda estava em versão beta e por esse motivo não pôde ser avaliado de acordo, principalmente porque os modelos, cenários e bibliotecas do ns-2 não podiam ser importados diretamente para a nova versão, o que reduz drasticamente o número de opções de protocolos e bibliotecas disponíveis para uso.

O OMNeT++ (Site do OMNeT++) é um ambiente de simulação de eventos discretos programado em C++ e desenvolvido por Andras Varga (Varga A. , The OMNeT++ Discrete Event Simulation System, 2001). O simulador é *open source*, baseado em componentes, e, assim como o ns, também conta com uma boa base de usuários que contribui ativamente com vários módulos para integração com a ferramenta. A partir da versão 4.0 o OMNeT++ tem uma IDE no Eclipse (Site do Eclipse).

Embora sua principal aplicação seja simulação de redes de comunicação, o OMNeT++ é um simulador de uso geral e pode ser aplicado, por exemplo, na simulação de sistemas distribuídos, validação de arquitetura de computadores, entre outras (Site do OMNeT++). Devido a sua composição modular o OMNeT++ apresenta uma curva de aprendizado relativamente pequena.

O OPNET (Site do OPNET) apresenta um conjunto de ferramentas para simulação, modelagem, otimização, planejamento e análise de desempenho para várias aplicações especializadas. Uma diferença relevante em relação aos outros ambientes é que o OPNET é uma ferramenta comercial.

O J-Sim (Site do J-Sim) é um ambiente de simulação de redes, baseado em componentes, *open source*, e desenvolvido totalmente em Java. O J-Sim ainda não tem tantos modelos de protocolos e bibliotecas como os simuladores discutidos anteriormente.

O GLASS (*GMPLS Lightwave Agile Switching Simulator*) (Site do GLASS) é uma ferramenta voltada para simulação de Redes Ópticas, livre, de código aberto e com interface gráfica desenvolvida e implementada em JAVA. O GLASS foi implementado utilizando como base a plataforma SSF (*Scalable Simulation Framework*) (Site do SSF) que oferece componentes de redes como hosts, roteadores, enlaces de comunicação e alguns protocolos de redes. O simulador utiliza a DML (*Domain Modeling Language*) (Site DML) para projetar a topologia e gerar os scripts de simulação.

O JiST (*Java in Simulation Time*) também permite a simulação de redes de comunicação em Java. Ele é usado na maioria das vezes em conjunto com o SWANS

(*Scalable Wireless Ad hoc Network Simulator*), um simulador para redes móveis ad hoc construído em cima do próprio JiST (site do JiST).

Na avaliação de vários simuladores apresentada em (Begg, Liu, Pawlikowski, Perera, & Sirisena, 2006), o ns2 e o OMNeT++ aparecem como os dois grandes concorrentes do mundo *open-source*. Cabe ressaltar que essa foi uma das comparações mais detalhadas entre as encontradas no esforço de pesquisa desta dissertação. Em (Orfanus, Lessmann, Janacik, & Lachev, 2008) existe um estudo com a avaliação dos simuladores J-Sim, OMNeT++, ns-2 e Shox. O interessante dessa avaliação é que não só a qualidade dos resultados da simulação foi levada em consideração, mas também a usabilidade dos simuladores e a qualidade dos modelos de simulação utilizados, pontos importantes quando se pretende desenvolver um *framework* que poderá ser utilizado por diferentes grupos e em variadas abordagens.

Em (Köksal, 2008) é apresentada uma extensa comparação entre os ambientes de simulação mais populares com suporte à simulação de redes sem fio. Na comparação entre J-Sim, OMNeT++, ns-2 e OPNET, são avaliadas as habilidades de cada um, assim como suas estruturas internas, ambiente de desenvolvimento, documentação e interface gráfica. São apontadas as vantagens e fraquezas de cada ambiente, e várias comparações são feitas baseadas em vários artigos e pesquisas. Novamente o ns-2 e o OMNeT++ são apresentados como os ambientes que mais se adéquam às necessidades dos pesquisadores.

Uma comparação de desempenho entre o ns2 e o ambiente de simulação JiST/SWANS pode ser vista em (Schoch, Feiri, Kargl, & Weber, 2008). Apesar de concluir que ambos os ambientes de simulação são similares em relação ao seu desempenho básico (dados os parâmetros usados na comparação), o artigo aponta a dificuldade de comparar resultados de diferentes implementações em diferentes ferramentas de simulação.

Entre os artigos mais recentes em (Weingartner, vom Lehn, & Wehrle, 2009) têm-se a comparação entre SimPy, ns-2, ns-3, OMNeT++ e JiST. Os três últimos são apontados como as melhores escolhas em relação à escalabilidade. Em (Garrido, Malumbres, & Calafate, 2008) é apresentado um estudo comparativo entre o ns2 e o OPNET em ambientes MANET.

Dada a necessidade de se utilizar o framework para pesquisas científicas, a adoção dos ambientes de simulação livres se apresenta como uma das decisões do projeto. De acordo com os estudos relacionados nos parágrafos anteriores, tanto o ns2 quanto o OMNeT++ seriam alternativas viáveis, disponibilizando um grande número de bibliotecas. No entanto, no início do desenvolvimento desta dissertação a nova versão do ns, o ns3, não era compatível com a versão anterior e poucas bibliotecas haviam sido portadas. Já o OMNeT++, também em nova versão, é compatível com grande parte das bibliotecas de sua versão anterior.

Além disso, o OMNeT++ apresenta interface gráfica integrada e uma linguagem de modelagem abstrata para descrição das redes enquanto as outras alternativas dependem do desenvolvimento de toda a simulação em código fonte (Weingartner, vom Lehn, & Wehrle, 2009). Por esses motivos, o OMNeT++ foi escolhido como o ambiente de simulação a ser utilizado nesta dissertação.

3.4 O ambiente de simulação Omnet ++

O OMNeT++ (*Objective Modular Network Testbed in C++*) é o ambiente de simulação utilizado no desenvolvimento das simulações desta dissertação. Ele tem uma arquitetura baseada em componentes, extensível, hierárquica e modular sendo *open-source* para uso acadêmico e não comercial.

Apesar do OMNeT++ não ser um simulador de redes, atualmente sua aplicação primária é na área da simulação de redes de comunicação, onde vem ganhando popularidade na comunidade acadêmica. Apesar disso, por conta de sua arquitetura flexível e genérica, ele também pode ser usado em muitas outras áreas como na simulação de complexos sistemas de TI, redes de filas ou arquitetura de hardware. Adicionalmente, o ambiente oferece várias extensões para áreas especializadas, como por exemplo: o MiXiN que permite a simulação de redes sem fio; ou o framework INET que suporta simulações de redes de comunicação oferecendo uma série de modelos como UDP, TCP, SCTP, IP, IPv6, Ethernet, entre outros.

O OMNeT++ funciona tanto no Linux, e sistemas “*Unix-like*”, quanto no Windows. O simulador inclui suporte à simulação paralela (para a execução mais rápida de simulações ou a modelagem de redes com milhares de estações) e conta com um manual de usuário relativamente grande e bem escrito, assim como documentação das APIs e alguns tutoriais.

A estrutura modular hierárquica do OMNeT++ ajuda a tratar a complexidade de modelagem. Qualquer componente complexo pode ser implementado como uma única unidade ou construído através de vários componentes menores. Os dois principais elementos do OMNeT++ são os módulos e as mensagens. Os módulos representam as entidades que se deseja simular, enquanto que as mensagens são os objetos trocados entre os módulos através suas portas (*gates*) e representam eventos no domínio simulado. Por exemplo, em uma rede IP típica, os módulos seriam os roteadores, computadores e *hubs*, e as mensagens representariam os pacotes trocados entre esses elementos.

Os módulos se dividem em dois tipos: os módulos simples e os módulos compostos. Os módulos simples são individuais, independentes (dito atômicos), e possuem seu comportamento programado em C++. Entende-se por comportamento as ações que o módulo

realiza ao receber uma mensagem. Um módulo composto é formado pelo agrupamento de módulos simples, ou de outros módulos compostos, conectados através de suas portas, como mostra a Figura 3.2, não existindo limite nos níveis dessa hierarquia. A conexão desses elementos é feita em uma linguagem de descrição de rede (NED – *Network Description Language*) criada especificamente para o simulador. Enquanto o comportamento dos módulos simples é descrito em C++, o comportamento do módulo composto depende de como seus elementos internos estão conectados e trabalham em conjunto.

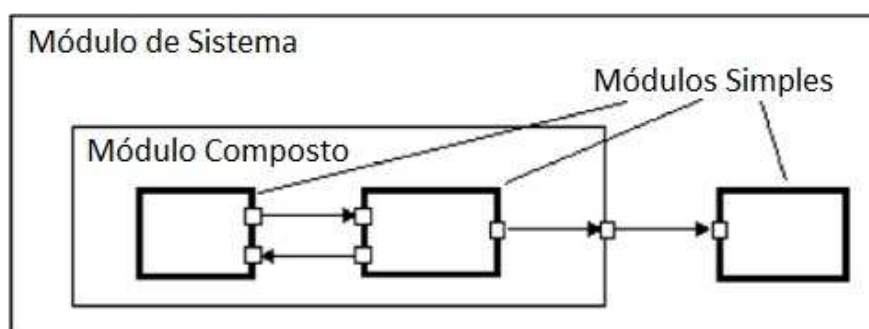


Figura 3.2: Relação hierárquica entre os módulos simples, compostos e uma rede no OMNeT++ (Varga & Hornig, *An overview of the OMNeT++ simulation environment*, 2008).

A topologia da rede simulada também é descrita na linguagem NED, de maneira similar a composição dos módulos compostos. Com isso, o OMNeT++ provê uma boa separação entre descrição da topologia da rede e o comportamento das entidades modeladas. Os modelos no OMNeT++ também são fáceis de gerenciar no sentido de que são independentes do *kernel* de simulação. No OMNeT++, o *kernel* de simulação é uma biblioteca de classes facilmente distinguível dos componentes escritos por programadores e pesquisadores. Após a escrita dos componentes esses podem ser facilmente integrados a simulação sem necessidade de modificação nas fontes do OMNeT++.

Através do envio de mensagens é que os módulos se comunicam. Nas simulações as mensagens podem representar *frames* ou pacotes e conter qualquer tipo de estrutura de dados. As mensagens podem ser recebidas tanto de outros módulos quanto do próprio módulo. Quando uma mensagem chega a partir do mesmo módulo, ela é chamada de “*self-message*” sendo usada normalmente para criar temporizadores. Uma mensagem pode ser enviada tanto diretamente quanto através de portas que estão interligadas através de conexões. As conexões podem estar associadas a parâmetros como, por exemplo, atrasos de propagação, taxas de erros e taxas de dados.

Pelo menos dois arquivos são necessários para a configuração de uma simulação OMNeT++: o arquivo `omnetpp.ini` que especifica parâmetros globais, por exemplo, qual

módulo começa a simulação; e os arquivos NED (*.ned), que descrevem a rede sendo simulada e os módulos usados. Cada módulo define suas portas, parâmetros, e submódulos em um arquivo NED separado. Finalmente, um arquivo NED global especifica a interconexão desses módulos que acabam formando a rede simulada. Além desses arquivos de configuração, o *framework* também usa: arquivos de mensagens (*.msg) que definem tipos de mensagens e pacotes e suas estruturas; e os códigos fontes dos módulos simples escritos em C++ (*.cc e *.h).

A Figura 3.3 mostra um exemplo dos arquivos necessários para uma simulação OMNeT++. O cenário exemplo consiste de três nós: um roteador (*Router*) e dois hosts (*StandardHost*) que estão interconectados por canais Ethernet (*ethernetLine*). O arquivo *omnetpp.ini* especifica o módulo inicial *MyNetwork*. Esse módulo composto é definido no arquivo NED *MyNetwork.ned* e consiste na definição dos canais, descrição dos submódulos presentes e a interconexão entre eles. Os submódulos são definidos em arquivos NED separados. O módulo composto *Router*, por exemplo, consiste nos módulos *NetworkLayer* e *RoutingTable* assim como algumas portas e interconexões desses submódulos.

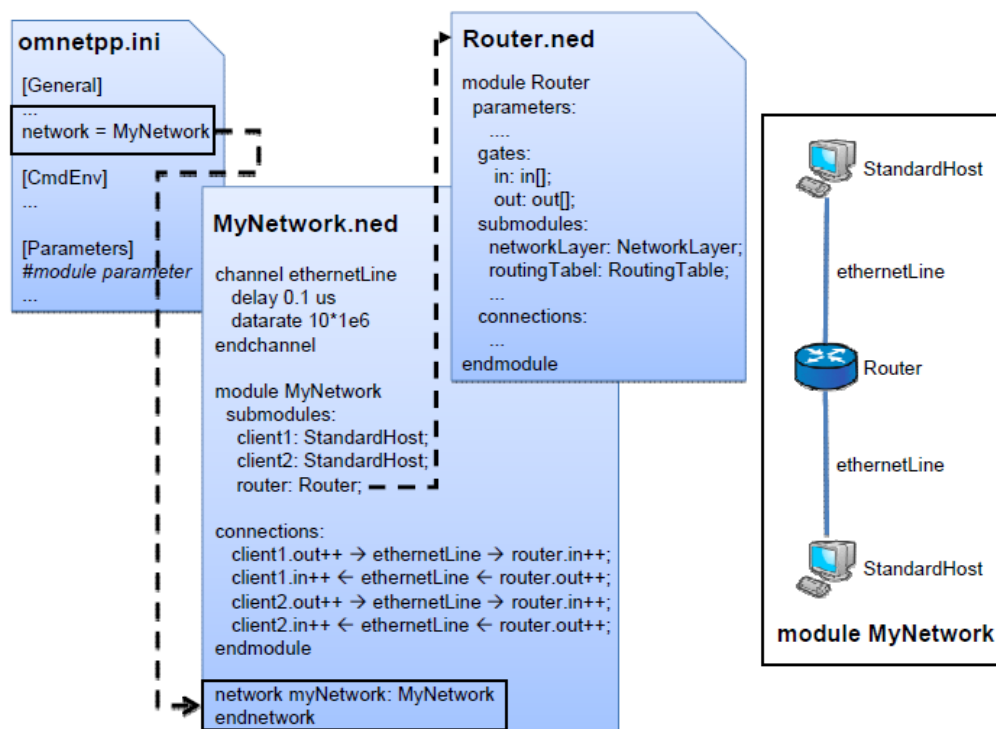


Figura 3.3: Configuração de uma simulação no OMNeT++ (Gamer & Scharf, 2008).

A partir da versão 4.0 do OMNeT++ está disponível uma IDE para o ambiente de simulação. Ela é baseada no Eclipse e permite, entre outras coisas, a criação e configuração de modelos em ambiente gráfico (arquivos .NED e .ini), execução de funções em “batch” e a

análise dos resultados de simulação. Além disso, o ambiente, por ser baseado no Eclipse, fornece a capacidade de editar, compilar e depurar arquivos C++, integração com sistemas de controle de versão, e outras características opcionais a partir de vários *plugins* tanto *open-source* quanto comerciais (modelagem UML, integração com *bugtracker*, acesso a banco de dados, etc.).

Os principais componentes dessa interface são: O editor NED, o editor de arquivos .ini, o *launcher* de simulações, o diagrama de sequências e a ferramenta de análise de resultados.

Os editores NED e ini podem ser usados tanto no modo texto quanto no modo gráfico e a troca de modos é possível a qualquer momento. Dessa maneira com o uso da IDE é possível a criação de topologias inteiramente no modo gráfico. A Figura 3.4 apresenta a interface da IDE do OMNeT++ editando um arquivo NED.

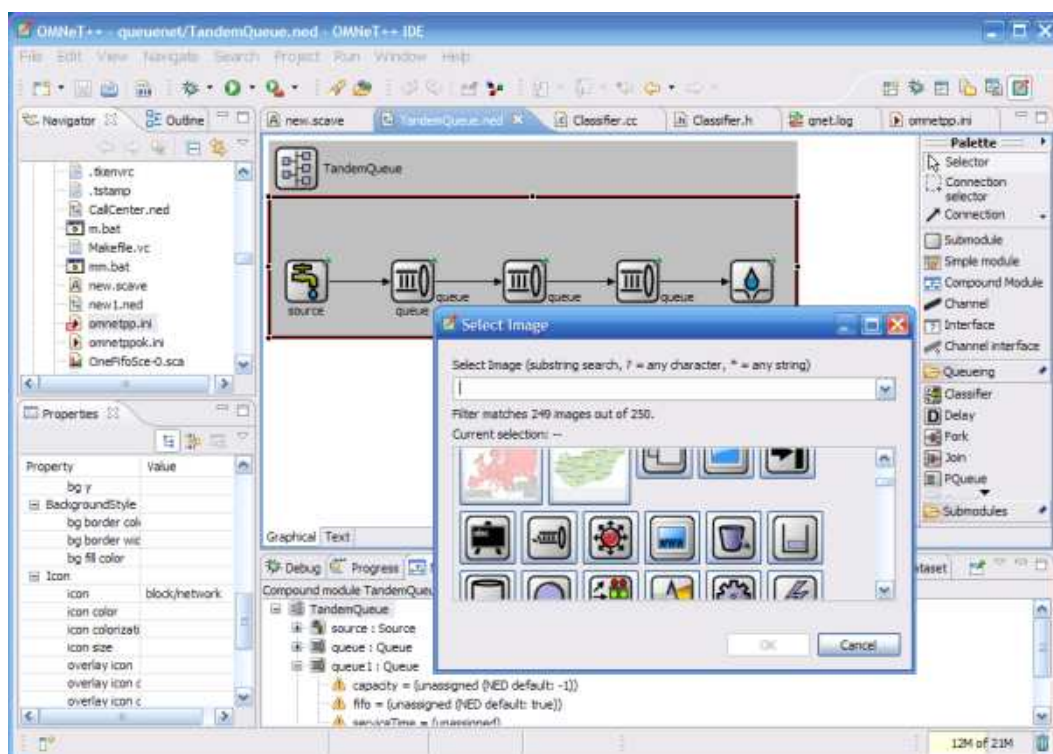


Figura 3.4: Interface da IDE do OMNeT++ 4.1 editando um arquivo NED (Site do OMNeT++).

O *launcher* é o responsável por executar as simulações diretamente da IDE. A simulação pode ser executada como uma aplicação C++ normal, ou usando uma das interfaces de execução presentes no ambiente: Tkenv (gráfica) ou Cmdenv (linha de comando). A interface gráfica de execução contida no OMNeT++ mostra as topologias das redes, os nós e as mensagens, tudo atualizado dinamicamente durante a execução. Além disso, permite que a velocidade seja ajustada para facilitar a observação, fornecendo a possibilidade de uma

inspeção detalhada dos conteúdos das mensagens e das variáveis dos nós, inclusive com a possibilidade da alteração de seus valores dinamicamente. Para concluir, a interface gráfica pode ser desabilitada para resultados de simulação mais rápidos.

O *kernel* do OMNeT++ é capaz de registrar muitos eventos gerados durante a simulação, praticamente tudo que acontece durante uma execução. O resultado é um arquivo de *log* com informações detalhadas da simulação, que pode ser usado de vários modos, sendo que o seu formato (baseado em linhas de texto) facilita o seu processamento por qualquer tipo de ferramenta. Uma dessas ferramentas é o diagrama de sequências da própria IDE. Nesse diagrama os eventos são representados como nós, as mensagens como setas e os módulos são as linhas. O diagrama permite a filtragem por módulos, tipos de mensagens, entre vários outros critérios.

O diagrama de sequências gerado pelo OMNeT++ foi uma ferramenta indispensável na elaboração do protótipo, já que permitiu a rápida avaliação e depuração de alguns algoritmos e protocolos presentes nos trabalhos que foram integrados para gerar o protótipo final. Um exemplo do diagrama de sequências gerado pelo OMNeT++ pode ser visto na figura 3.5.

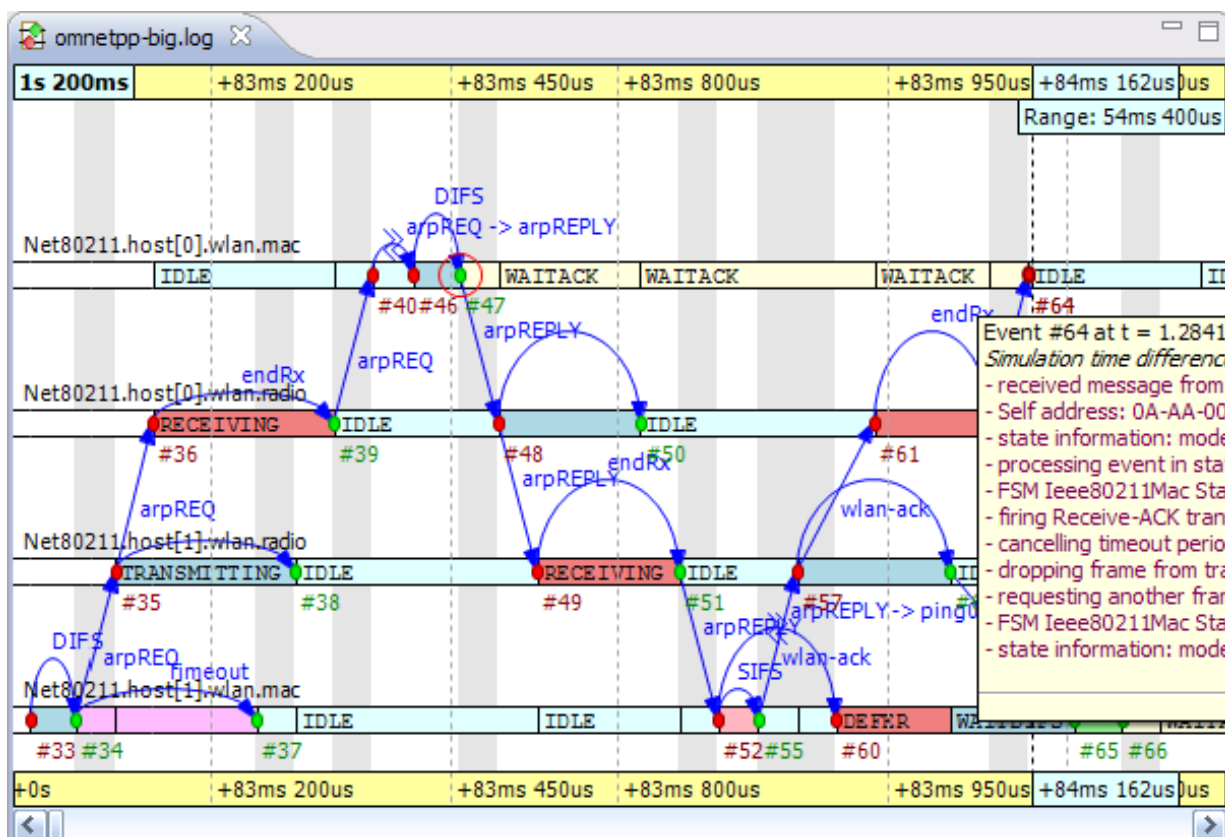


Figura 3.5: Diagrama de Sequências no OMNeT++ (Site do OMNeT++).

3.5 Conclusão

Como pode ser observado neste capítulo, vários ambientes para a simulação e estudo de sistemas de comunicação estão disponíveis. Utilizando esses ambientes de simulação uma série de projetos vem sendo desenvolvida, no entanto a maioria dos grupos de pesquisa utiliza plataformas de simulação, ou modelagens próprias, e a reutilização desses modelos se torna difícil, assim como a reprodução e comparação dos resultados obtidos.

Nesse cenário, um ambiente uniforme para a simulação de redes OTN é essencial. Isso iria fornecer aos pesquisadores de redes OTN vários benefícios, entre eles a possibilidade de reutilizar bibliotecas de protocolos e componentes, uma plataforma aberta para a criação de novos protocolos e adição de novas funcionalidades, além de uma maior facilidade para comparar resultados de diferentes frentes de pesquisa.

Para remediar esse fato, nesta dissertação é apresentada uma proposta de *framework* de simulação para redes ópticas OTN, independente de fabricantes e aberto. No intuito de aproveitar muito do trabalho desenvolvido nos ambientes de simulação, principalmente em relação à construção do *core* do simulador, o OMNeT++ foi escolhido como ambiente de simulação e o *framework* proposto é desenvolvido com base na sua plataforma.

Capítulo 4: Proposta do Framework

A presente dissertação tem como principal objetivo apresentar uma proposta de *framework*, independente de fabricantes, aberto, e voltado especificamente para as redes OTN. É esperado que o *framework*, além de modelar a camada de transporte OTN segundo as recomendações ITU-T, atenda três desafios específicos: escalabilidade, flexibilidade e realismo.

Escalabilidade: Deve-se garantir a escalabilidade da simulação de modo que modelos com uma grande quantidade de entidades de rede e com carga realística de tráfego sejam capazes de caracterizar o comportamento da rede num tempo razoável.

Flexibilidade: Configuração dinâmica e operação contínua da rede. A rede virtual simulada deve ser capaz de executar múltiplos experimentos simultaneamente ou em sucessão, com a possibilidade de mudanças nas configurações de rede com facilidade e sem interrupção do funcionamento. Além disso, a modelagem deve ser a mais genérica possível de modo que qualquer tipo de tecnologia usada para implementar as funções descritas nas recomendações possam ser simuladas.

Realismo: É necessário que os resultados das simulações sejam plausíveis e que não apresentem nenhum tipo de distorção grosseira e possam, portanto, ser usados como aproximações do que está sendo observado.

Além disso, o *framework* foi planejado com três categorias de usuários em mente. Primeiro existem os usuários que gostariam de estudar o funcionamento das redes OTN. Checar como as recomendações da ITU funcionam na prática, observar como funciona a interação entre as camadas OTN, o tratamento dos diferentes tipos de sinais clientes, o funcionamento do TCM e as temporizações das mensagens. Observar também como a rede se comporta em situações de falha de componentes específicos, a correlação de alarmes e todos os outros detalhes da arquitetura.

Um segundo grupo de usuários são os desenvolvedores de equipamentos OTN. Um ambiente de simulação como o proposto nesta dissertação pode ser muito útil na especificação de novos equipamentos. Seria muito interessante observar como equipamentos que atendem apenas parcialmente as recomendações se comportam quando colocados em uma rede OTN. Quais serviços oferecidos pelas redes OTN ainda funcionam nesse estágio de desenvolvimento do equipamento? Quais características devem ser adaptadas ou desligadas? Esses são alguns exemplos de perguntas que a simulação poderia ajudar a responder.

Na terceira categoria de usuários estão aqueles que estão desenvolvendo novas tecnologias e protocolos para as redes OTN e precisam de redes para testar e avaliar o que está sendo desenvolvido. Neste caso estariam, por exemplo, usuários desenvolvendo novas alternativas para plano de controle ou gerência de redes OTN e que poderiam usar o *framework* para avaliar suas idéias e até mesmo testar protótipos iniciais.

Outras categorias poderiam incluir os usuários interessados em mais de um desses tópicos, por exemplo, como um determinado equipamento OTN em desenvolvimento se comporta quando funciona em conjunto com um módulo de gerência também em desenvolvimento? Espera-se que com o uso do *framework* seja possível desenvolver ambos os projetos em paralelo de modo que os produtos finais tenham uma probabilidade muito maior de funcionarem como o esperado.

O *framework* proposto foi desenvolvido numa arquitetura modular, incluindo vários componentes básicos, entre eles: as camadas OTN, o canal óptico, geradores de tráfego e as mensagens ópticas e elétricas. Além disso, a implementação do *framework* permite a rápida adição de novos componentes, algoritmos e protocolos. Alguns exemplos de módulos extras que podem ser executados, utilizando-se o *framework*, são incluídos na versão atual e descritos em detalhes no próximo capítulo, como, por exemplo, o plano de controle GMPLS e o módulo de Discovery.

A Figura 4.1 ilustra os principais componentes do *framework* e a sua estrutura interna básica: O tráfego cliente é gerado em módulos geradores de tráfego e então enviado para os equipamentos OTN. Os equipamentos OTN são compostos pelas camadas OTN, e um módulo de gerência local que é responsável por toda a comunicação entre os equipamentos OTN e os outros módulos presentes no ambiente, que aqui são chamados de módulos extras. Além disso, existem os canais ópticos e as mensagens. Na próxima seção, são descritos com detalhes cada um desses módulos, como são modelados e que tipo de abstração é utilizada nesta modelagem.

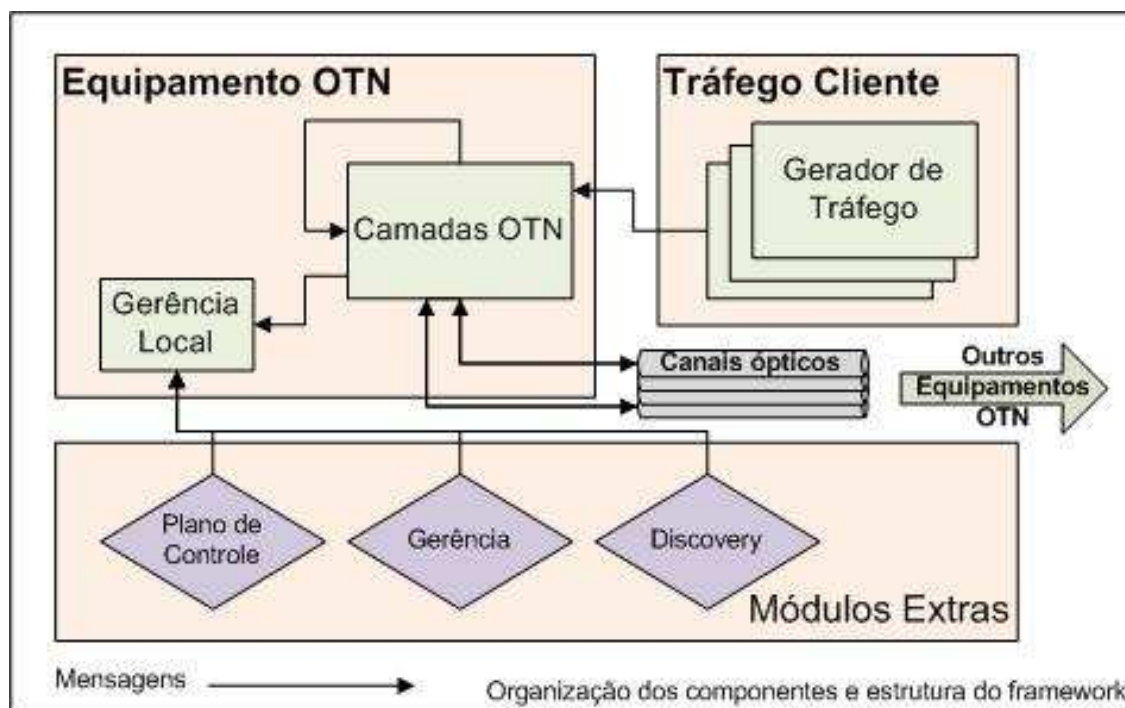


Figura 4.1: Estrutura proposta do Framework.

Essa estrutura básica foi concebida a partir da integração e adaptação dos diversos componentes sendo desenvolvidos pelo grupo de pesquisa em telecomunicações da UFES (F.F.Ferrari, 2009), (Tessinari, 2009), (Frigini, 2010), (Ferrari, Frasson, & Garcia, 2010) e utilizando como base a modelagem proposta em (Favoreto, 2009). Uma série de adaptações teve que ser feita nos componentes individuais de modo que estes se integrassem na estrutura proposta do *framework*. Essas alterações são detalhadas na modelagem que é apresentada a seguir.

A modelagem inicial do framework de simulação está apresentada em (Favoreto, 2009). Esse trabalho tem como objetivo relacionar as recomendações da ITU sobre redes OTN e as recomendações do IETF relativas ao GMPLS a fim de obter uma solução consistente e genérica de um plano de controle para as redes OTN. Para alcançar o objetivo é realizado um estudo de ambos os padrões OTN e GMPLS, identificando em que ponto há uma conexão entre eles. É analisado como o GMPLS deve representar os recursos da rede OTN e quais as extensões dos protocolos do GMPLS que devem ser utilizadas para o estabelecimento de caminhos ópticos em uma rede OTN.

O trabalho desenvolvido se propõe a responder as seguintes questões: 1) Como os recursos da rede OTN são representados pelo plano de controle? Quais entidades guardam essas informações? 2) No caminho contrário: O que é um TE link em uma rede OTN? E um LSP, como ele é implementado na rede OTN? 3) Como se dará o agrupamento de links? E o

endereçamento dos links e interfaces OTN? 4) Quem são os elementos de rede, os comutadores ou roteadores da rede OTN que são de responsabilidade dos controladores? 5) E os processos de gerência de links? Quais processos devem-se usar?

A validação da proposta do plano de controle é feita através do desenvolvimento de modelos de simulação discreta no OMNET++: a primeira versão do framework. Nessa versão é desenvolvida a primeira modelagem dos equipamentos ópticos e é também apresentada a modelagem do canal óptico DWDM. Como o foco é a validação da proposta de plano de controle, a modelagem é mais focada no canal de serviço e nos protocolos do GMPLS do que na implementação completa das camadas OTN. Por isso somente a parte das camadas ópticas OTN é modelada e mesmo assim parcialmente, já que somente a parte relativa ao *payload* está desenvolvida, sendo as partes relativas ao *overhead* e à correlação de alarmes, deixadas para trabalhos futuros.

Em (Tessinari, 2009) essa primeira versão foi expandida com a modelagem completa da maioria das camadas ópticas contidas na recomendação (ITU-T, G.709, 2009). Para cada camada são implementadas suas funções de terminação de trilha e de adaptação, tanto para o lado *source* quanto do lado *sink*. A modelagem foi realizada de forma a abranger tanto a parte de dados (*payload*) quanto a de informações de controle (*overhead*). Além disso, também foram modeladas informações de alarme e informações remotas (que são trocadas inter e intra camadas). Para a realização dos testes, dois equipamentos foram modelados utilizando-se as camadas criadas. O primeiro deles é um amplificador de linha bidirecional e o outro um host OTN capaz de enviar e receber informação em seis comprimentos de onda, cinco para dados e um para informações de overhead. As camadas digitais ainda não foram implementadas nesta versão. (Frigini, 2010) expande ainda mais o trabalho de (Tessinari, 2009), implementando as camadas digitais e integrando parcialmente com as camadas ópticas já existentes.

Em (F.F.Ferrari, 2009) é apresentado um estudo da funcionalidade de *discovery* automático para redes OTN (*Optical Transport Network*), baseado nas recomendações desenvolvidas pela ITU-T. Atualmente, busca-se a automatização das funcionalidades da gerência, tornando-a mais eficaz para o crescimento das redes. Neste cenário, o *discovery* da rede é uma importante funcionalidade da gerência de configuração que visa à descoberta dos equipamentos e conexões da rede.

O trabalho visa à exemplificação do processo de *discovery* para redes OTN com a criação de um protótipo baseado no procedimento apresentado em (ITU-T, G.7714, 2005) e (ITU-T, G.7714.1, 2003). O protótipo visa realizar a troca de mensagens descritas nas recomendações e, a partir dos dados obtidos dos TCPs, verificar os *links* bidirecionais entre as

terminações de uma camada de rede. Através do protótipo, espera-se a obtenção de um desenho da topologia, relatórios contendo o status dos *links* referentes às redes simuladas e a identificação das conexões incorretas existentes.

O protótipo de *Discovery* criado em (F.F.Ferrari, 2009) é independente e não tem nenhuma ligação com os trabalhos apresentados em (Favoreto, 2009), (Tessinari, 2009) e (Frigini, 2010) de modo que a sua integração com esses trabalhos é um dos resultados desta dissertação.

Na Tabela 4.1 um resumo do que foi desenvolvido nos trabalhos anteriores:

Tabela 4.1: Resumo do que foi desenvolvido nos trabalhos anteriores

Trabalho	Componentes desenvolvidos ou resultados obtidos
(Favoreto, 2009)	Estrutura de modelagem das camadas OTN, canal óptico DWDM, implementação de parte das camadas ópticas OTN, do canal de serviço e de um plano de controle GMPLS;
(Tessinari, 2009)	Complementação dos módulos das camadas ópticas OTN e inclusão da correlação de alarmes, além da primeira versão de equipamentos OTN: geradores de sinal e amplificadores ópticos;
(F.F.Ferrari, 2009)	Protótipo de <i>Discovery</i> ;
(Frigini, 2010)	Camadas digitais OTN, integração parcial com camadas ópticas e gerador de sinais PRBS.

4.1 Modelagem dos componentes

Como explicado no capítulo anterior os módulos do *framework* estão sendo desenvolvidos no OMNeT++. Todo o plano de transporte é modelado utilizando-se apenas as bibliotecas padrão do OMNeT++, isso inclui os equipamentos OTN, as fibras ópticas, as mensagens OTN tanto digitais quanto ópticas e os geradores de tráfego. Como o *framework* está sendo desenvolvido com base no OMNeT++ é possível aproveitar sua extensa biblioteca de protocolos e componentes de forma fácil e rápida.

A modelagem das camadas OTN forma a base do framework e tem como objetivo fornecer um modelo de plano de transporte de acordo com as normas da ITU-T. A partir dessa modelagem inicial são desenvolvidas as estruturas auxiliares necessárias ao correto funcionamento do *framework*, como: os canais ópticos, os geradores de tráfego e as mensagens OTN. Cada um desses componentes é configurável e inclui um número de parâmetros que podem ser ajustados durante a criação das redes ou durante a própria simulação.

Com isso já é possível modelar os equipamentos OTN e algumas redes de baixa complexidade. A partir deste ponto, módulos extras são desenvolvidos de modo a permitir a

construção de modelos de redes mais complexos, agregando funcionalidades como, por exemplo, a implementação de planos de controle e gerência.

4.1.1 Camadas OTN

A modelagem dos equipamentos OTN no OMNeT++ tem como base o mapeamento apresentado em (Favoreto, 2009). Neste mapeamento, equipamentos como ROADMs (*Reconfigurable Optical Add Drop Multiplex*), OXC's (*Optical Cross-connect*), amplificadores de linha e outros são modelados usando as funções de terminação de trilha e adaptação da recomendação (ITU-T, G.798, 2006).

Os equipamentos da rede OTN são módulos compostos que, internamente, são formados pela ligação dos módulos simples dos blocos funcionais de cada camada como especificados na (ITU-T, G.798, 2006). Observa-se que essa é uma abordagem funcional de modelagem dos equipamentos OTN. No mundo real, os OXC's, (R)ODMs, amplificadores de linha e outros equipamentos são compostos por um conjunto de elementos de hardware, como, por exemplo, amplificadores ópticos, (de)multiplexadores e fotodetectores.

Outra abordagem de modelagem seria construir módulos no OMNeT++ que representem esses elementos físicos e, a partir deles, construir modelos de equipamentos mais complexos como os OXC's e ROADMs. Essa abordagem não foi adotada por três motivos: Primeiro, porque em alguns equipamentos existe um grande número de elementos de *hardware*, o que tornaria a modelagem complexa e as simulações lentas. Segundo, porque é difícil garantir que um equipamento simulado está obedecendo à modelagem funcional definida nas recomendações da ITU-T, ou seja, de que o equipamento foi modelado de acordo com as normas. Isso exige uma profunda análise de comparação entre os elementos de *hardware* e a modelagem funcional da ITU-T para OTN. Terceiro, pois um dos objetivos do *framework* é ser o mais genérico possível. Caso os equipamentos OTN fossem modelados como seus elementos físicos, a abordagem de determinado fabricante poderia acabar sendo adotada e estaria se quebrando a regra da generalidade. Um exemplo de componente onde isso acontece é o canal de serviço, já que as recomendações deixam boa parte das especificações em aberto e cada fabricante o implementa da forma que mais lhe convém. No *framework* proposto o canal de serviço é implementado como um canal genérico, sem especificar taxas, tecnologia ou qualquer outra restrição de modo que qualquer tipo de tecnologia possa ser inserida quando se está modelando os equipamentos.

A seguir é mostrado um exemplo do tipo de modelagem com abordagem funcional utilizado nesta dissertação e como ela é aplicada na construção das camadas OTN presentes no *framework*.

A Figura 4.2 mostra um dos blocos funcionais da ITU-T G.798, conforme ele é apresentado na recomendação, neste caso específico o bloco *OTSn_TT_Sk*, a função de terminação de trilha da OTS no lado *sink*. Os números de 1 a 13 foram inseridos para identificar os processos internos do bloco.

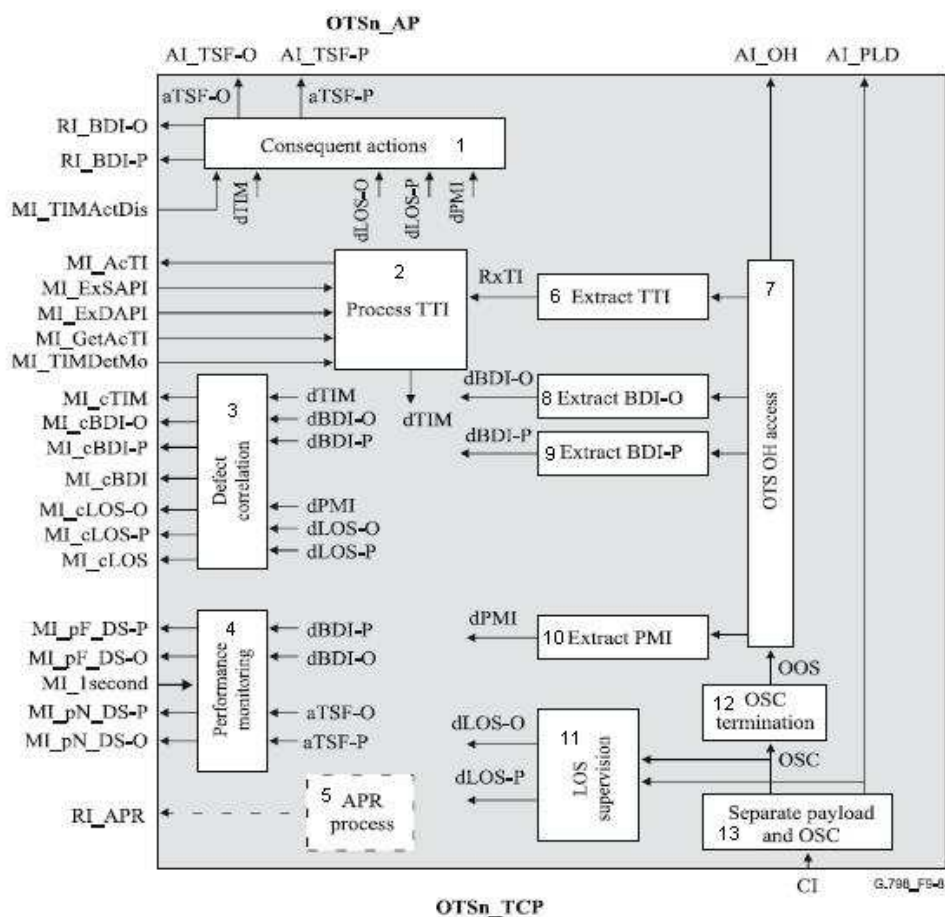


Figura 4.2: Função de terminação de trilha OTSn_TT_Sk (ITU-T, G.798, 2006).

O bloco *OTSn_TT_Sk* é a primeira terminação de trilha óptica, ela recebe o sinal óptico e faz a demultiplexação do canal de serviço, resultando em *payload* (PLD) e *overhead* (OH). Isso é realizado no bloco interno “*Separate payload and OSC*” (13 na Figura 4.2). Após a separação, ambos os sinais resultantes têm seus níveis de potência medidos (bloco 11, Figura 4.2), e caso estejam abaixo de um valor limite, um defeito é disparado (*dLOS-P* para PLD e *dLOS-O* para OH). Após a medição de potência, o PLD é transmitido à função de adaptação da camada OTS em forma de informação adaptada, *OTSn_AI_PLD*.

Diferente do payload, que na OTS são os n canais da camada OCh multiplexados, e que é simplesmente encaminhado, o *overhead* precisa ser convertido ao domínio elétrico para que seja possível o acesso às suas informações, isso acontece no bloco “OSC *termination*” (bloco 12 na Figura 4.2). O processo “OTS OH *access*” tem por função permitir a extração das informações lógicas relativas a camada OTS: TTI, BDI-O, BDI-P e PMI (*Trail Trace Identifier, Backward Defect Indication Overhead, Backward Defect Indication Payload e Payload Missing Indication*).

O bloco “*Consequent actions*” (1 na Figura 4.2) representa o processo das ações consequentes, que tem por função gerar os alarmes relativos à falhas na trilha AI_TSF-O e AI_TSF-P (*Trail Signal Fail Overhead e Trail Signal Fail Payload*) que são enviados para a função de adaptação OTSn_A_Sk. Para a camada OTSn_TT_So são enviados os alarmes RI_BDI-O e RI_BDI-P (*Backward Defect Indication Overhead e Backward Defect Indication Payload*), que devem ser de lá reenviados ao sítio anterior, conforme definido na ITU-T G.798.

Os processos restantes (2, 3 e 4 na Figura 4.2) representam processos de correlação de defeitos e correlações de alarme e têm como principal objetivo enviar informações de gerência.

A seguir são mostrados os códigos, gerados no *framework*, relativos a esse bloco funcional da OTN. A Figura 4.3(a) apresenta o código do módulo simples na linguagem NED, que consiste na declaração de seus “*gates*” (portas). Vale observar que todas as portas de entrada e saída recebem o nome dos sinais externos da função, conforme especificado na recomendação. Na Figura 4.3(b) tem-se a representação gráfica do módulo simples gerado pelo código mostrado na Figura 4.3(a).

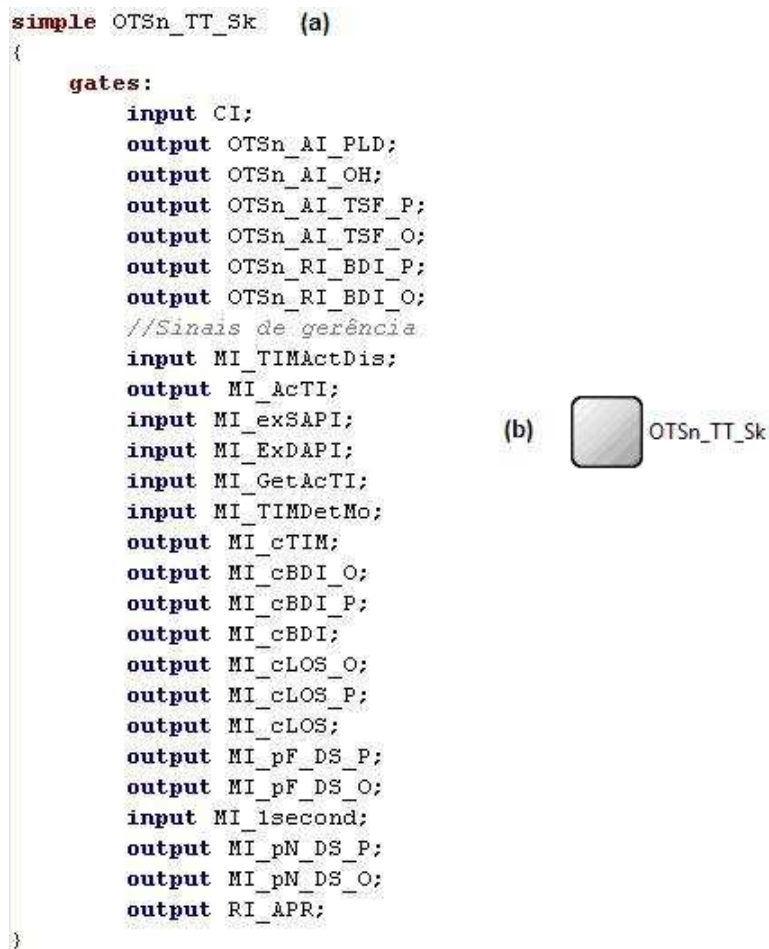


Figura 4.3: OTSn_TT_Sk. (a) código em linguagem NED. (b) Representação gráfica.

A Figura 4.4 apresenta o código da declaração da classe OTSn_TT_Sk. Os sinais internos da Figura 4.2 são modelados em forma de atributos de classe e os processos em métodos.

```

class OTSn_TT_Sk : public cSimpleModule
{
    private:
        bool dLOS_P;
        bool dLOS_O;
        bool RxTI;
        bool dBDI_P;
        bool dBDI_O;
        bool dPMI;
        bool dTIM;
        bool aTSF_P;
        bool aTSF_O;
        bool aBDI_P;
        bool aBDI_O;

    protected:
        virtual void initialize();
        virtual void handleMessage(cMessage *msg);
        virtual void process_TTI();
        virtual void OTS_OH_access(OTNMsgOH *osc);
        virtual void performance_monitoring();
        virtual void defect_correlation();
        virtual void consequent_actions();
};

```

Figura 4.4: Declaração de métodos e atributos da classe OTSn_TT_Sk.

Os processos “*Separate payload and OSC*”, “*OSC termination*” e “*LOS supervisor*” (processos 13,12 e 11 respectivamente na Figura 4.2) são implementados no método *handleMessage(cMessage *msg)*. Esse também é o método que envia o sinal OTSn_AI_PLD à camada superior.

O método *OTS_OH_access(OTNMsgOH *osc)*, é também responsável pela atualização dos atributos de classe dBDI_O, dBDI_P e dPMI (8,9 e 10 na Figura 4.2) e pelo envio do sinal OTSn_AI_OH.

No OMNeT++, seguindo este mesmo exemplo de implementação, os seguintes blocos funcionais da recomendação ITU-T G.798 são desenvolvidos como módulos simples:

- As terminações de trilha da camada OTS, lados *sink* (OTSn_TT_Sk) e *source* (OTSn_TT_So);
- As funções de adaptação da camada OTS com a OMS *sink* (OTSn/OMSn_A_Sk) e *source* (OTSn/OMSn_A_So);
- As terminações de trilha da camada OMS, lados *sink* (OMSn_TT_Sk) e *source* (OMSn_TT_So);

- As funções de adaptação da camada OMS com a OCh *sink* (OMSn/OCh_A_Sk) e *source* (OMSn/OCh_A_So);
- As terminações de trilha da camada OCh, lados *sink* (OCh_TT_Sk) e *source* (OCh_TT_So);
- As funções de adaptação da camada OCh com a OTU *sink* (OCh_OTUk_A_Sk) e *source* (OCh_OTUk_A_So);
- As terminações de trilha da camada OTU, lados *sink* (OTUk_TT_Sk) e *source* (OTUk_TT_So);
- As funções de adaptação da camada OTU com a ODU *sink* (OTUk_ODUk_A_Sk) e *source* (OTUk_ODUk_A_So);
- As terminações de trilha da camada ODU, lados *sink* (ODUkP_TT_Sk) e *source* (ODUkP_TT_So);
- As funções de adaptação da camada ODU com o cliente *sink* (ODUkP_Client_A_Sk) e *source* (ODUkP_Client_A_So);
- As funções de conexão OCh_C e ODUk_C.

Por serem módulos simples, todo o processamento e troca de sinais internos aos blocos são feitos em código C++. Já os sinais externos aos blocos funcionais são gerados pelo envio de mensagens entre módulos.

Sendo assim, qualquer equipamento OTN pode ser simulado apenas conectando e configurando os blocos funcionais corretamente. Por exemplo, um OXC sem estágio de pré-amplificação do sinal de entrada é modelado com os mesmos blocos de um OXC com pré-amplificação, já que a única diferença entre os dois está em uma subfunção de um dos blocos funcionais das camadas OTN. Porém, o módulo simples que contém a subfunção em que estão (OTSn/OMSn_A_Sk) deve ser configurado para desativar o sub-bloco opcional responsável pela pré-amplificação do sinal. Assim, não significa que os elementos de hardware não existam na simulação, mas sim que eles estão implementados de maneira implícita nos blocos funcionais da recomendação ITU-T G.798.

Na Figura 4.5 tem-se um exemplo de equipamento modelado segundo essa metodologia e com todos os blocos funcionais da recomendação. As camadas OTN digitais são agrupadas em um módulo composto “camadas_digitais” assim como as camadas ópticas (bloco “camadas_opticas”), a junção desses dois módulos, mais a configuração das camadas OTN resulta em um módulo composto que representa um equipamento OTN.

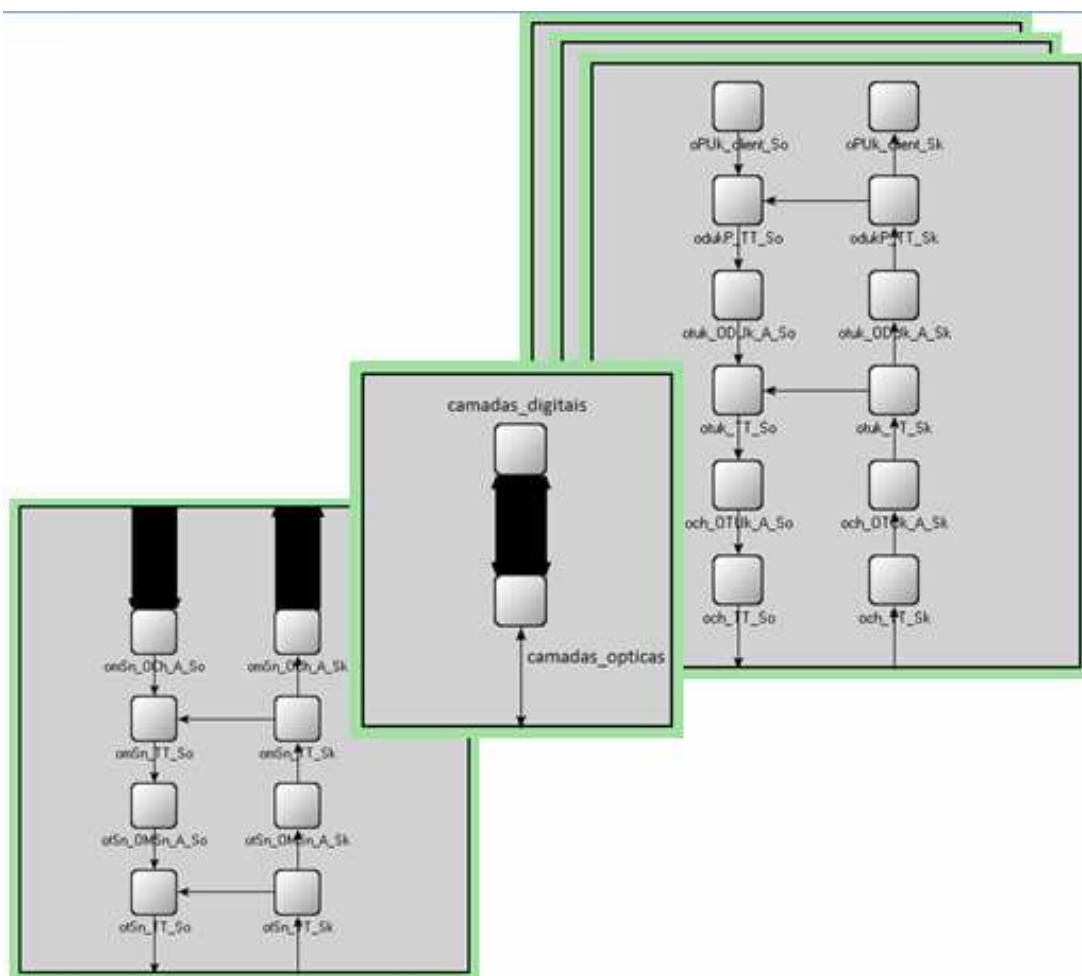


Figura 4.5: Equipamento OTN modelado no OMNeT++.

Um problema interessante identificado com essa abordagem de modelagem é na questão da representação visual dos equipamentos ópticos. Apesar de a abordagem usada no *framework* estar de acordo com a descrição funcional presente na recomendação (ITU-T, G.798, 2006), rigorosamente falando, a representação presente na Figura 4.5 está diferente das representações presentes nas recomendações ITU-T 872 e 805 já que a comunicação “vertical” entre pares *Sink* e *Source* está presente na própria camada e não em um *Access Point* entre camadas, como especificado nessas recomendações. Uma solução para esse pequeno impasse está sendo estudada.

A Figura 4.6 apresenta ambas as representações lado a lado. Na Figura 4.6 a) apresenta a representação da G.872, já na Figura 4.6 b) apresenta a representação das mesmas camadas como elas aparecem no *framework*.

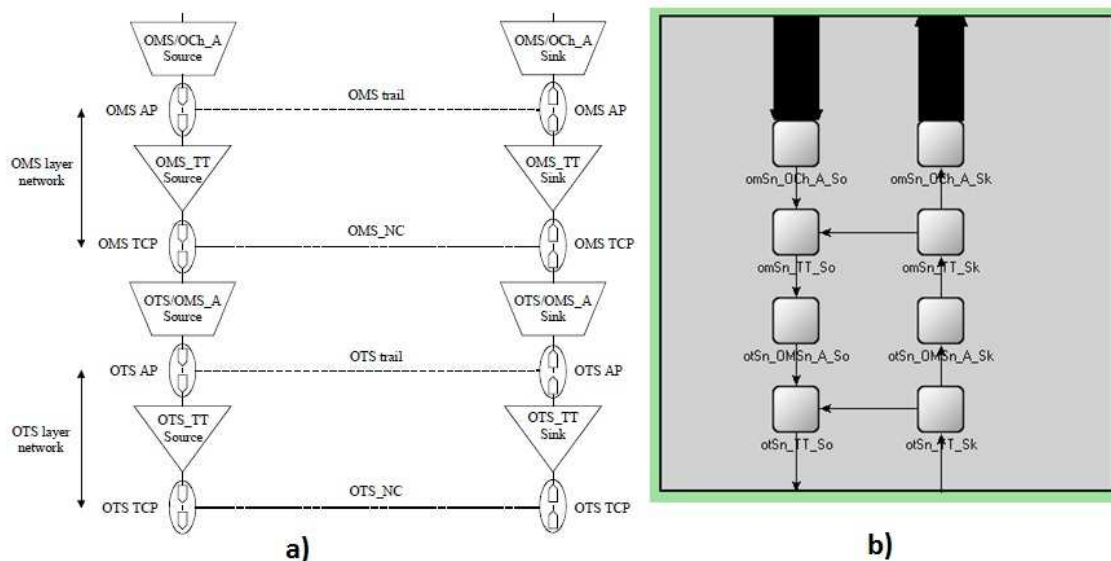


Figura 4.6: Representação visual das camadas OTN: a) (ITU-T, G.872, 2001) b)framework.

Boa parte da implementação e modelagem das camadas OTN conforme essa metodologia definida em (Favoreto, 2009) foi contruída em (Tessinari, 2009) e (Frigini, 2010). O trabalho desta dissertação é o responsável pela integração desses módulos e correção de alguns detalhes de implementação que acabavam fugindo à ideia de modelagem originalmente proposta.

Aqui cabe ressaltar que para o correto funcionamento dessa modelagem fez-se necessária também a implementação de outro módulo para ser usado em conjunto com as camadas OTN, o módulo da gerência local. Este componente é responsável por prover toda a comunicação externa e gerenciar as entradas e saídas dos sinais MI (*Management Information*) presentes nessas camadas. Quando um sistema externo precisa acessar informações relativas às camadas OTN, como, por exemplo, o plano de controle, toda a comunicação é feita através da gerência local. O acesso direto às camadas não é permitido.

Em geral os equipamentos reais têm um componente similar nas suas implementações, a placa supervisor. Além de facilitar a comunicação externa, a gerência local permite a organização dos sinais do tipo MI. Caso fosse permitida a ligação direta a esse tipo de porta a probabilidade de erro de conexão seria grande (dado o grande número de portas), além disso, a depuração desse tipo de erro é difícil já que em alguns casos essas portas só são usadas em casos específicos e raros. Além disso, estaria se violando um dos requisitos de projeto que é a modularidade do *framework*, quando uma camada fosse alterada, por exemplo, qualquer outro módulo que se utiliza dessa camada precisaria ser alterado também já que a interface poderia mudar. Desse modo, por decisão de projeto, foi criado a gerência local, e o acesso às portas MI só é permitido através de métodos públicos presentes neste módulo.

Uma ilustração do módulo de gerência local junto com as camadas digitais e ópticas pode ser visto na Figura 4.7. As setas bidirecionais representam os sinais de entrada e saída de gerenciamento. Na Figura 4.2, por exemplo, os sinais de gerenciamento são todos os sinais que iniciam por MI_*. Os sinais que iniciam por RI_* são do tipo “*Remote Information*” e trocados entre os “pares” *Sink* e *Source* de cada camada OTN.

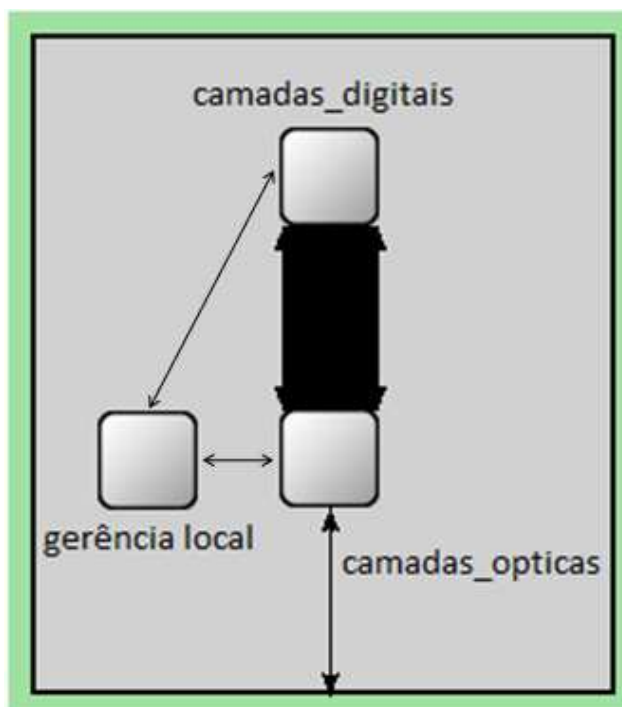


Figura 4.7: Módulo de gerência local.

Está sendo desenvolvida uma API para padronizar a comunicação com o módulo de gerência local de modo que os módulos extras possam ser desenvolvidos de forma independente à implementação das camadas OTN e demais componentes do *framework*.

4.1.2 Canal óptico e mensagens OTN

No OMNeT++ as mensagens são trocadas entre os módulos por canais, que, por padrão, podem ser de três tipos diferentes: 1) *IdealChannel*, um canal ideal sem atraso de propagação ou de transmissão, 2) *DelayChannel*, um canal apenas com atraso de propagação, e 3) *DatarateChannel*, um canal com atrasos de propagação, atrasos de transmissão e taxa de erro de *bit*. A Figura 4.8 exemplifica o tempo de envio de uma mensagem entre os nós N1 e N3, considerando a existência de um nó intermediário ao caminho utilizado, N2, utilizando cada um desses tipos de canais.

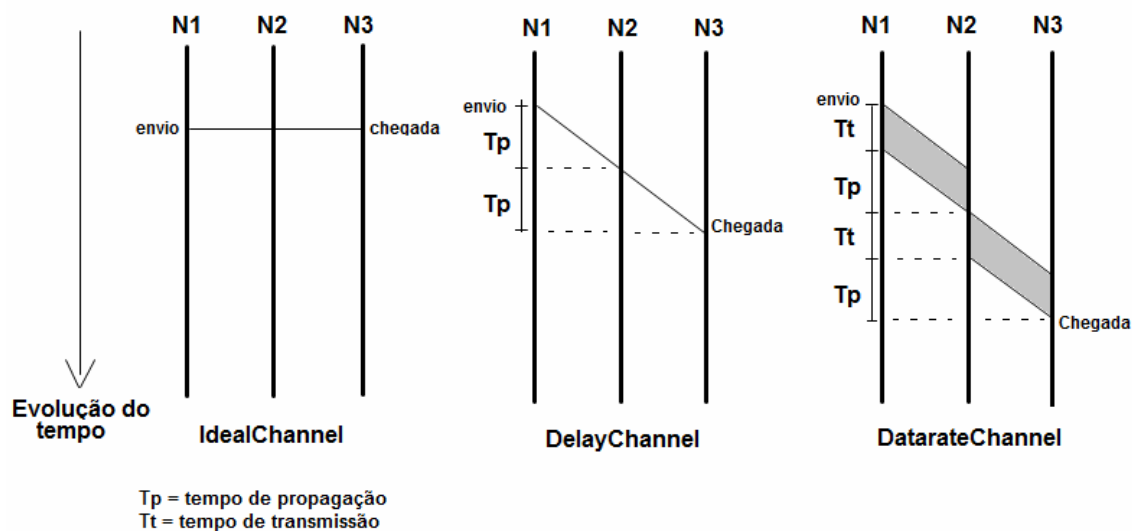


Figura 4.8: Tempo de envio nos canais do OMNeT++.

Nenhum desses canais modela a comunicação do modo que seria necessário ao funcionamento das redes OTN. Desse modo, para o desenvolvimento do framework é necessário modelar um canal que represente uma fibra por onde podem passar diversos comprimentos de onda independentes. Este tipo de canal não existe na API padrão do OMNeT++ e foi desenvolvido nesta dissertação.

Dois requisitos principais não são atendidos pelos canais já disponíveis no ambiente de simulação. Primeiro, nenhum dos canais está preparado para transmitir sinais ópticos e tratar essa transmissão em termos de potência do sinal óptico e relação sinal/ruído. Segundo, nenhum desses canais permite que os comprimentos de onda sejam modelados de forma independente e depois enviados ao mesmo tempo. Como explicado anteriormente essa dissertação está preocupada com a modelagem funcional da OTN, e trata da abstração no nível de rede, de modo que a modelagem da dispersão na fibra não faz parte do escopo deste trabalho.

Na modelagem adotada, cada sinal WDM é representado por uma mensagem do OMNeT++, a classe OTNMsg. Cada mensagem OTNMsg representa um sinal óptico de um comprimento de onda específico sendo propagado pela fibra. A OTNMsg possui os seguintes parâmetros: comprimento de onda; potência; relação sinal ruído; e tamanho. O conteúdo transportado por cada mensagem é um quadro OTN da camada OTU, conforme a recomendação (ITU-T, G.709, 2009). Assim como um sinal elétrico, o sinal óptico sofre um atraso de propagação na fibra. O atraso de transmissão vai depender da quantização adotada.

No OMNeT++, cada canal é implementado por uma classe C++, assim como os módulos. Então foi criada uma nova classe, baseada no *DataRateChannel*, o *OpticalChannel*.

O *DataRateChannel* é usado como base pois já modela os atrasos de propagação, transmissão e taxa de erros de *bit*. Este novo canal permite o envio das mensagens OTNMsg, além de simular a queda de potência, a mudança na relação sinal ruído e BER (*Bit Error Ratio*). A principal diferença do novo canal é a possibilidade de envio de mais uma mensagem ao mesmo tempo, desde que seus comprimentos de onda sejam diferentes.

O *OpticalChannel* tem como parâmetros: distância, taxa de queda de potência, taxa de transmissão e BER. A queda de potência pode ser calculada com base na distância e na taxa de queda de potência da fibra, ou de acordo com uma variável especificada dinamicamente para cada fibra sendo simulada. Quando uma *OpticalMsg* entra no canal, ela possui um valor de potência que é decrescido de um valor ao atravessar o canal. De modo similar, a relação sinal ruído diminui a cada amplificação.

Com essas novas classes disponíveis, é possível simular um elemento puramente óptico, onde o sinal transita de maneira contínua, sem atrasos devidos ao enfileiramento ou processamento de mensagens. Na Figura 4.9 o NO2 da segunda parte da figura representa, por exemplo, um ROADM no meio de um caminho óptico entre os nós NO1 e NO3.

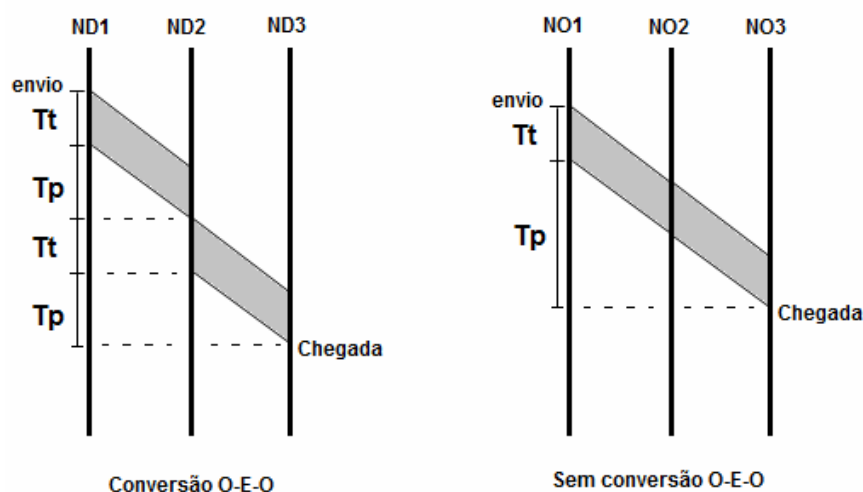


Figura 4.9: Tempos de transmissão e propagação com e sem conversão O-E-O.

Um sinal óptico, ao atravessar um ONE que implementa apenas as camadas ópticas da OTN, é conduzido da interface de entrada direto para a respectiva interface de saída como um fluxo contínuo, não existindo uma conversão O-E-O do sinal WDM. A Figura 4.9 compara a transmissão de dados em nós digitais ou puramente ópticos. Em uma rede com conversão O-E-O, ou em uma rede completamente digital como as redes de computadores, a mensagem que transporta os dados é completamente armazenada antes de ser reenviada por um nó (ND2). No caso de transmissão por nós puramente ópticos, embora haja um tempo para

transmitir o sinal WDM, não existe uma *bufferização* do sinal óptico nos nós intermediários (NO2). Assim que a mensagem é recebida, já começa a sua transmissão pela interface de saída. O final da retransmissão acontece simultaneamente com a sua chegada completa no nó que está retransmitindo.

Dado o foco na modelagem funcional das redes OTN, e ao tipo de abstração adotada, não é intuito desta dissertação gerar uma modelagem completamente fiel à transmissão de sinais ópticos por uma fibra. No entanto, dado o tipo de modelagem adotado, caso seja necessário, é possível que em trabalhos futuros esses cálculos sejam desenvolvidos e definidos com mais rigor. Isso depende apenas da necessidade de quem estiver usando o *framework* e do tipo de dados que se pretende observar.

4.1.3 Gerador de tráfego

Para que seja possível executar simulações e testes, é necessário que haja tráfego no sistema. Para tal, um gerador de tráfego foi construído, responsável por gerar os sinais clientes a serem processados nos blocos funcionais das camadas.

O gerador é descrito no arquivo *Host_OTN.ned*, onde estão contidas também, além do gerador de sinal, todas as camadas da hierarquia OTN desse *host*. O arquivo apresenta uma variável *n*, representando *n* comprimentos de onda, sendo que o valor de *n* pode ser definido dinamicamente durante os testes. Foi escolhido fazer a representação dos *hosts* OTN de forma genérica, pois dessa forma tem-se uma melhor flexibilidade na simulação. Cada *host* é capaz de enviar mensagens OTN, com *n* comprimentos de ondas distintos, representando os *n* canais ópticos suportados pela simulação.

Cada um desses *hosts* é composto por *n* conjuntos de blocos digitais e um bloco de camadas ópticas. Isso porque cada bloco de camada digital recebe um sinal cliente e após os devidos tratamentos referentes às camadas digitais gera um dos comprimentos de onda da camada óptica. A estrutura resultante pode ser observada na Figura 4.5. Inicialmente foi implementado apenas o sinal cliente do tipo PRBS (*Pseudo-Random Bit Sequence*) já que é uma das formas mais simples e genéricas de representar um sinal cliente entre os tipos de sinais clientes definidos na recomendação (ITU-T, G.798, 2006).

A interface com os sinais clientes está sendo padronizada de modo a permitir que, futuramente, qualquer tipo de sinal cliente (conforme definido nas recomendações) possa ser tratado por uma rede OTN. Desse modo os sinais clientes já modelados no OMNeT++ poderiam ser facilmente integrados à uma simulação OTN.

4.1.4 Módulos extras

Além das camadas OTN e dos módulos auxiliares, como o gerador de sinal e os canais e mensagens ópticos, que são indispensáveis à execução das simulações, existem módulos extras que são inseridos na simulação tanto para teste quanto para validação de novas idéias. Essa categoria de componente seria utilizada pelo terceiro tipo de usuário definido no início do capítulo, usuários que estão desenvolvendo novas tecnologias e protocolos para as redes OTN e precisam de modelos da rede de transporte para testar e avaliar o que está sendo desenvolvido.

Esses módulos foram planejados para interagir com a camada de transporte que o *framework* provê. Espera-se que os usuários do *framework* utilizem as camadas OTN conforme definido para modelar os equipamentos ópticos desejados e, desse modo, possam aproveitar uma camada de transporte OTN que segue as recomendações ITU para testar tecnologias, protocolos e redes utilizando-se do *framework*. Exemplos de módulos extras em desenvolvimento são apresentados no próximo capítulo.

4.2 Estrutura geral do framework

Resumindo: os cenários de simulação são construídos a partir de equipamentos OTN, módulos auxiliares (mensagens, canais e geradores), e os módulos extras. Os equipamentos OTN por sua vez podem ser tanto os equipamentos genéricos fornecidos pelo *framework* quanto os equipamentos construídos pelos próprios usuários. Em ambos os casos os equipamentos são criados a partir da junção de módulos das camadas OTN e de um módulo de gerência local. Os módulos extras interagem com os equipamentos através de uma interface da gerência local e também podem ser criados pelos usuários do *framework*.

Na Figura 4.10 a seguir é mostrada **Erro! Fonte de referência não encontrada.** a estrutura do projeto do *framework* no OMNeT++ resultado da integração dos vários projetos que estavam individualmente em desenvolvimento. A pasta do plano de transporte (*Transport_Plane*) contém os módulos relativos às camadas ópticas e digitais da OTN e aos componentes auxiliares (canal óptico, mensagem OTN, etc...). Na pasta equipamentos (*Equipments*), aparecem alguns exemplos de equipamentos modelados segundo o que foi proposto neste capítulo. Na pasta de redes (*Networks*) aparecem as redes de teste modeladas com os equipamentos e componentes auxiliares definidos anteriormente. E finalmente na pasta de plano de controle (*Management_Plane*) aparecem os módulos extras que estão em desenvolvimento no momento.

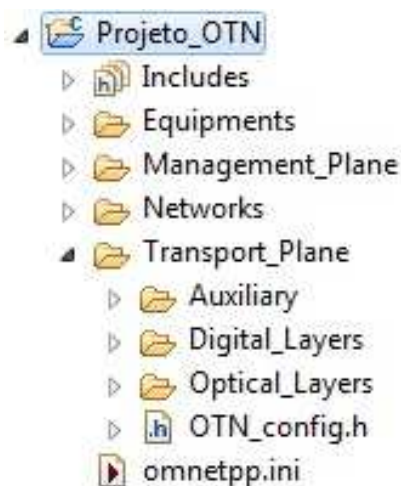


Figura 4.10: Organização dos componentes do *framework* dentro do OMNeT++.

Essa estrutura geral é resultado da adaptação de vários trabalhos conforme o que está presente na Tabela 4.1. Desse modo, uma modelagem formal do *framework* ainda não foi realizada. Sendo este um trabalho de engenharia, essa descrição formal foi postergada em favor da criação do protótipo do *framework* e da avaliação de várias decisões de projeto na prática. Essa descrição formal, no entanto, está em desenvolvimento e espera-se que utilize também da ontologia de redes OTN conforme descrito na seção de trabalhos relacionados para validar todos os componentes do *framework*.

Na Tabela 4.2, são apresentados os componentes e contribuições desta dissertação em relação ao que já havia sido desenvolvido pelo grupo de pesquisas em telecomunicações da UFES:

Tabela 4.2: Resumo do que foi desenvolvido nesta dissertação

Módulo	O que foi desenvolvido nesta dissertação
Estrutura do framework	Durante este trabalho que foi desenvolvida uma arquitetura única de modo a permitir a integração de todos os componentes sendo desenvolvidos em paralelo;
Camadas OTN	Definição do componente da gerência local, integração das camadas digitais e ópticas, ajustes de alguns pontos de implementação tanto das camadas ópticas quanto digitais;
Canais	Redefinição de como o canal óptico poderia funcionar e aperfeiçoamento da implementação referente aos canais digitais;
Mensagens ópticas	Com uma nova versão do OMNeT++ as mensagens tiveram que ser refeitas e as mensagens digitais tiveram que ser aperfeiçoadas;
Gerador de tráfego	Foi estudada e proposta um novo modo de funcionamento para o gerador de tráfego de modo a permitir o fácil desenvolvimento e integração de novos tipos de tráfego;
Equipamentos OTN	Vários equipamentos OTN foram modelados dentro do sistema e disponibilizados para uso de forma padronizada com a organização dos componentes dentro do projeto no OMNeT++;

Módulos extras	Os módulos extras que foram antes criados de forma independentes estão agora adaptados e integrados ao <i>framework</i> (<i>Discovery</i> e <i>GMPLS</i>). Um módulo de comunicação foi desenvolvido.
----------------	---

4.3 Conclusão

Concluindo o que foi apresentado neste capítulo, o usuário do *framework* terá uma série de funcionalidades presentes no ambiente de simulação dependendo do seu perfil e do que está buscando no *framework*.

Caso seja um usuário do primeiro perfil apresentado no início do capítulo, ou seja, usuários que gostariam de estudar o funcionamento das redes OTN, ele pode utilizar as modelagens genéricas de equipamentos que estão presentes no *framework* para montar a rede de teste e utilizar as ferramentas disponibilizadas pelo OMNeT++ para observar e verificar o que está acontecendo durante e após a simulação.

Caso seja um usuário do terceiro tipo, aqueles que estão desenvolvendo novas tecnologias e protocolos para as redes OTN, além de montar as redes de teste conforme o exemplo anterior, ainda tem que ficar atentos à interface de comunicação com o bloco de gerência local já que é através dessa interface que os blocos que estão sendo desenvolvidos no *framework* se comunicam com os equipamentos OTN.

E, finalmente, no caso dos usuários da segunda categoria, os desenvolvedores de equipamentos OTN, as redes de teste vão ser montadas com equipamentos OTN configurados pelo próprio usuário. Esses equipamentos devem ser construídos pela ligação dos módulos simples das camadas OTN de acordo com a metodologia apresentada na subseção 4.1.1. Dependendo dos tipos de camadas OTN presentes nesse equipamento e de quais funções estão ou não ativas dentro dessas camadas é que vai ser caracterizado o tipo de equipamento sendo modelado. Além disso, através da configuração de parâmetros internos aos módulos das camadas, o funcionamento do equipamento pode ser ajustado para espelhar aquele do equipamento sendo modelado.

Na Figura 4.11 é apresentada uma ilustração desse cenário:

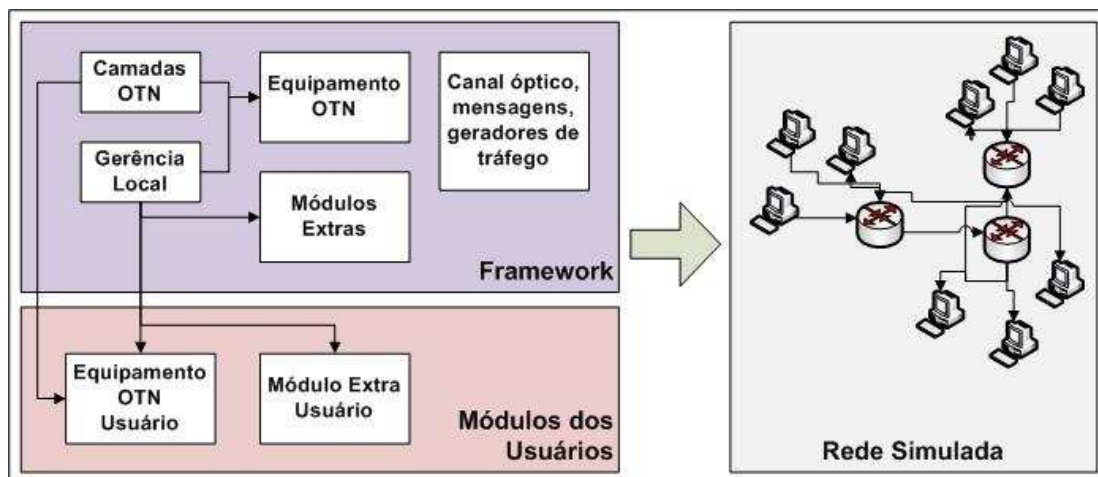


Figura 4.11: Componentes do *framework* e modos de uso.

Quanto às características propostas no início do capítulo, o seguinte foi avaliado nesta dissertação: Flexibilidade, dado o uso do ambiente de simulação OMNeT++ e da modelagem modular do *framework* apresentada neste capítulo acredita-se que neste ponto a proposta atende o que foi definido no início do projeto. O *framework* pode ser facilmente estendido e permite alto grau de customização dada a utilização de arquivos amigáveis de configuração e com um grande número de parâmetros que permitem configuração.

Quanto a escalabilidade, testes simples preliminares foram realizados para verificar o número de nós e a quantidade de comprimentos de onda que o ambiente de simulação suporta e os resultados são apresentados no próximo capítulo. No ponto de realismo, foi tomado o cuidado na modelagem para ser o mais fiel possível ao que está definido nas recomendações ITU-T. A abordagem funcional da modelagem facilitou isso ainda mais. Além disso, o OMNeT++ permite que sejam coletados vários dados estatísticos de modo que o funcionamento do *framework* pode ser sempre verificado.

Capítulo 5: Simulações e resultados preliminares

Um grande problema quando se está desenvolvendo um novo simulador é a validação do que está sendo construído. Como apontado pelas pesquisas descritas no Capítulo 3 não foi encontrada uma implementação de referência com a qual o *framework* pudesse ser validado, especificamente no quesito conformidade com as recomendações ITU-T, de modo que tanto a modelagem quanto a validação da implementação são feitas com base nas recomendações da ITU.

5.1 Testes

Num primeiro momento cada um dos blocos funcionais OTN, implementados conforme especificado na seção 3.1, são testados em relação às suas entradas e saídas. São testadas as diferentes combinações de entrada e saída dos blocos e assim verificado se o módulo está funcionando segundo especificado nas recomendações.

Para realizar essa tarefa são utilizadas ferramentas disponibilizadas pelo IDE do OMNeT++, mais especificamente pelo Eclipse. O Eclipse possui extensões que permitem o desenvolvimento de conjuntos de testes que podem então ser aplicados sobre os módulos sendo desenvolvidos no simulador. Nesses testes são configuradas as combinações de sinais de entrada que um módulo poderia receber e qual a saída esperada. Desse modo a cada alteração na implementação dos módulos esses testes são executados e é verificada a conformidade com o funcionamento esperado.

Num segundo momento são testados os equipamentos e componentes ópticos criados a partir da conexão dessas camadas ópticas OTN. Por exemplo, componentes como o presente na Figura 4.5. A partir desse teste é possível verificar se os blocos OTN estão corretamente interconectados e se o equipamento está funcionando de acordo com o tipo de sinal que deveria gerar ou processar. Como novamente o interesse é apenas nas entradas e saídas dos módulos compostos, a mesma metodologia do teste anterior é utilizada.

Além disso, é verificado se a modelagem dos equipamentos estava seguindo o que está definido nas recomendações da ITU. Um grande esforço de comparação é despendido nessa etapa já que os equipamentos reais como comentado no capítulo anterior são construídos a partir de elementos físicos e não nas camadas OTN.

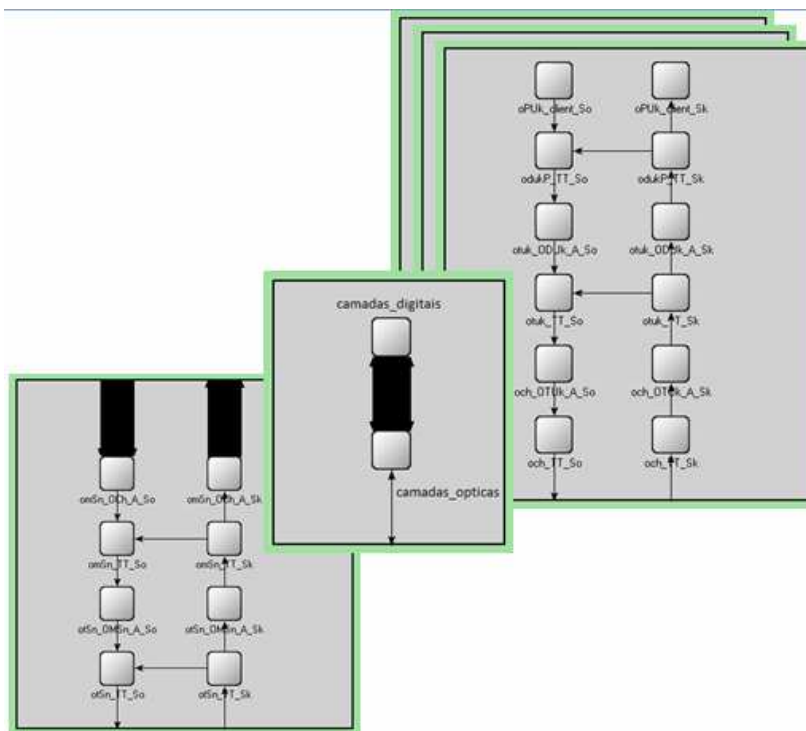


Figura 5.1: Equipamento OTN genérico modelado no OMNeT++

Num terceiro momento é usado um modelo de rede para verificar o funcionamento do simulador. Na Figura 5.2 está ilustrada a rede construída no simulador para este teste.

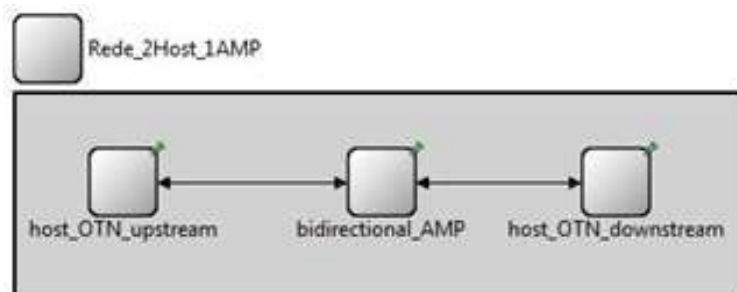


Figura 5.2: Rede OTN de teste.

Neste exemplo, o objetivo é testar a implementação de um amplificador de rede bidirecional. A rede simulada é composta então por um par de *hosts* OTN, ligados a um amplificador de linha bidirecional. Apesar de uma rede simples, atinge o objetivo de demonstrar a troca de mensagens entre as camadas e a geração de alarmes. No exemplo, o *Host Upstream* gera sinais de *n* comprimentos de onda que passam pelo amplificador de linha e são então direcionadas ao *Host Downstream* que faz um *log* do estado do sinal quando o recebe.

Usando a interface do OMNeT++ é fácil observar a troca de mensagens durante a execução. Apesar disso, esta não é a melhor maneira de observar o funcionamento dos

módulos e outra ferramenta disponibilizada pela IDE foi muito útil nesta etapa. O *kernel* de simulação do OMNeT++ é capaz de gerar arquivos de *log* de vários eventos que acontecem durante a simulação, entre eles: agendamento e cancelamento de mensagens, o envio de mensagens, criação e remoção de módulos e conexões, etc. Com esses *logs* é possível criar um diagrama de seqüências que permite observar com detalhes o que está acontecendo na simulação e se os módulos estão funcionando como deveriam.

O diagrama de seqüência permite observar em detalhes, e numa linha do tempo, o momento em que cada evento está ocorrendo, a seqüência de mensagens que foi disparada em cada situação e como essas se relacionam entre si.

A Figura 5.3 e a Figura 5.4 ilustram os diagramas de seqüências obtidos em dois testes sequenciais. Nos testes a rede da Figura 5.2 foi configurada para funcionar com cinco comprimentos de onda. Cada linha horizontal no diagrama de seqüências representa um módulo e na visualização das figuras ainda foi aplicado um filtro de modo que só os módulos de interesse estão aparecendo, nesse caso o último módulo óptico do *host_upstream* e todas as camadas do amplificador. As setas representam as mensagens e eventos.

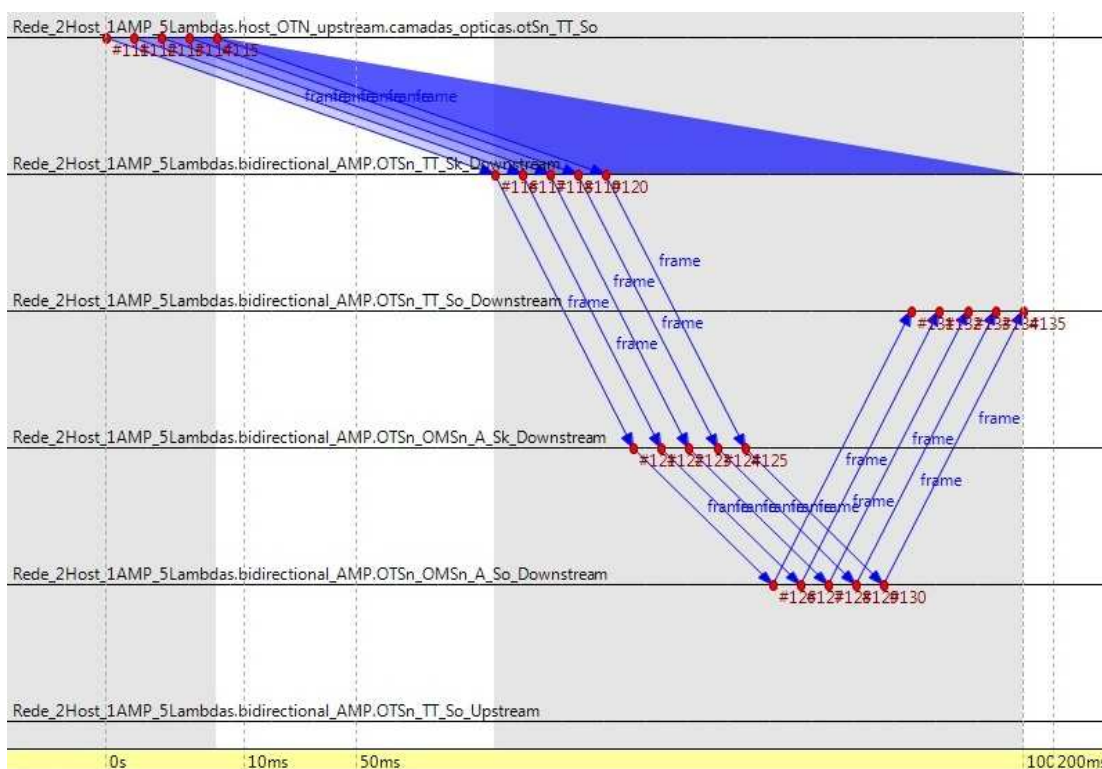


Figura 5.3: Diagrama de seqüência em teste de correlação de alarmes em situação normal.

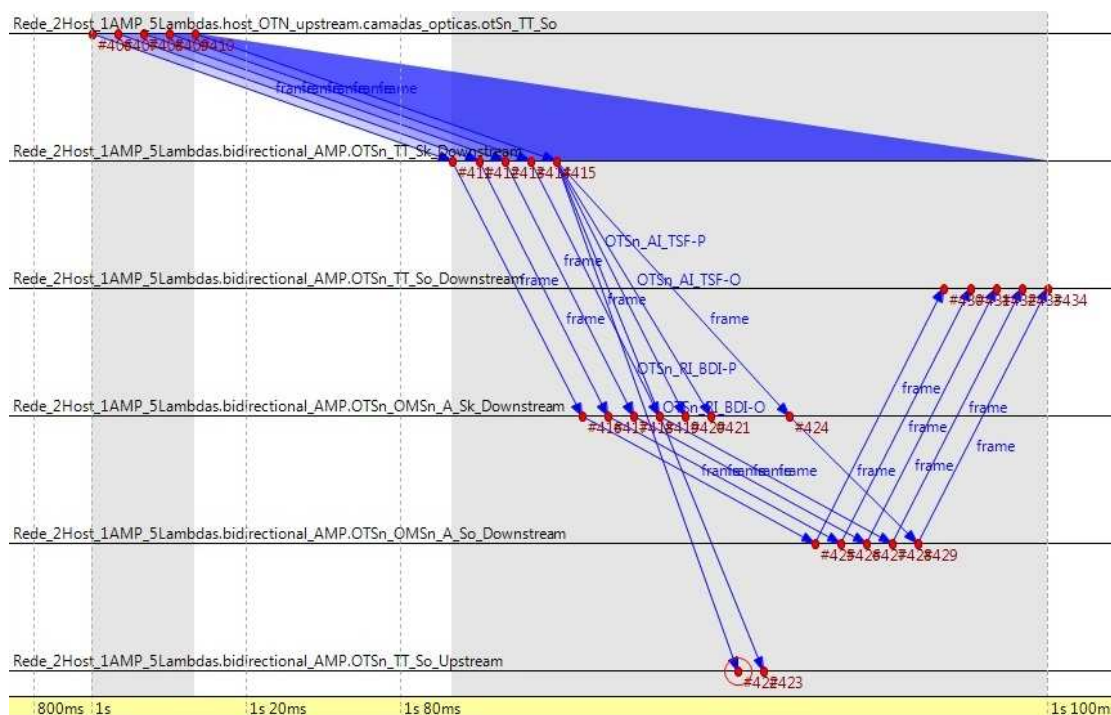


Figura 5.4: Diagrama de sequência em teste de correlação de alarmes em situação com dLOS-P na OTS.

Na Figura 5.3 vê-se uma situação normal em que o amplificador recebe os sinais ópticos, processa e repassa para o componente seguinte. Cabe ressaltar que apesar de as mensagens de cada comprimento de onda estarem deslocadas em relação umas as outras, isso é uma característica dos diagramas de sequência apresentados pelo OMNeT e servem para melhor visualizar o que está acontecendo, olhando a linha do tempo na parte de baixo da figura é possível observar que as transmissões estão acontecendo todas ao mesmo tempo.

Na Figura 5.4 o sinal de *payload* estava com a potência abaixo do valor de referência de modo que um alarme (*dLOS-P*) é disparado. Isso pode ser visualizado pelas duas setas extras que estão conectando o segundo módulo ao último. As duas mensagens são responsáveis por passar os alarmes de *payload* e os de *overhead* para o módulo *source* do módulo *sink* onde foi observado o problema (*OTSn_TT*). Além desse tipo de visualização, a interface permite clicar em cada um dos eventos e verificar exatamente o que está acontecendo diretamente no diagrama de sequência.

A partir das recomendações G.709 e G.798, um esquema de correlação de alarmes foi montado pelo grupo em planilhas. Neste esquema é possível observar todos os eventos que devem ser disparados em cada um dos cenários de falha definidos nas recomendações. Comparando o resultado do diagrama de sequência gerado pelas simulações e o que estava definido neste esquema, é possível verificar se a correlação de alarmes das camadas OTN modeladas no protótipo do *framework* estava funcionando corretamente.

5.2 Módulos Extras

Após esses testes simples e iniciais o protótipo do *framework* está em uso de modo que a todo o momento está sendo verificada a funcionalidade dos blocos básicos. Atualmente existem três módulos extras em desenvolvimento que estão utilizando o protótipo do *framework* como plataforma de desenvolvimento e testes: uma proposta de plano de controle GMPLS para redes ópticas; um módulo que agrega funcionalidades de *Discovery* para redes ópticas OTN; e um módulo para possibilitar a comunicação externa ao ambiente de simulação.

5.2.1 Plano de controle GMPLS

Este módulo pretende demonstrar a operação de uma rede OTN com controladores GMPLS baseados na proposta descrita em (F.P.Favoreto, A.S.Garcia, M.T.A.Torres, R.S.Tessinari, & Segatto, 2009). De maneira geral, esta proposta apresenta como implementar um plano de controle GMPLS com base nas RFCs do IETF para as redes OTN, baseadas nas recomendações da ITU-T.

O plano de transporte, que inclui os equipamentos OTN e as fibras WDM, é modelado conforme as descrições anteriores e fazem parte da base do *framework*. O plano de controle GMPLS é desenvolvido através de extensões do INET, que é um *framework* para o OMNET++ para simulação de redes. A simulação se preocupa principalmente com a arquitetura do GMPLS aplicada às redes OTN, abordando suas principais características, bem como os processos envolvidos na sinalização, roteamento e gerência de *links*.

5.2.2 Discovery

Atualmente, busca-se a automatização das funcionalidades da gerência, tornando-a mais eficaz para o crescimento das redes. Neste cenário, o *discovery* da rede é uma importante funcionalidade da gerência de configuração que visa à descoberta dos equipamentos e conexões da rede.

Este módulo tem como principal objetivo testar e validar o processo de Discovery apresentado nas recomendações da ITU-T e as propostas de extensão desenvolvidas em (F.F.Ferrari, 2009) e (Ferrari, Frasson, & Garcia, 2010). O protótipo visa realizar a troca de mensagens descritas nas recomendações e, a partir dos dados obtidos dos TCPs, verificar os *links* bidirecionais entre as terminações de cada camada da rede. Este protótipo realiza a simulação de uma rede e executa os procedimentos do *discovery* para a criação de grafos que representam a topologia da rede e a identificação das conexões incorretas existentes.

5.2.3 Comunicação externa

A emulação é um híbrido que aproxima aplicações reais, de ambientes de redes sintéticos ou simulações. Com uma extensão para emulação, uma simulação operando em tempo aproximadamente real pode rodar junto com aplicações reais distribuídas e serviços da rede. Essas aplicações e serviços podem gerar pacotes que são inseridos no sistema da simulação. Perdas de pacotes e atrasos são calculados como função da situação atual da rede simulada. Esse tipo de simulação não só permite que não seja necessário desenvolver modelos separados para cada aplicação nas simulações, como também aumentam o nível de confiabilidade da simulação, já que sistemas reais são incluídos no modelo da rede.

Um emulador de rede permite que *hosts* reais interajam através de uma rede virtual. Ele combina um simulador com capacidade de ser executado em tempo aproximadamente real, junto com mecanismos para captura e injeção de pacotes numa rede real. Pacotes gerados por *hosts* externos interagem com tráfego sintético dentro da simulação criando um ambiente controlado para testar aplicações reais da Internet e diversos protocolos de rede.

Quais elementos são reais e quais são parcial ou totalmente simulados normalmente diferem, dependendo das necessidades do pesquisador e dos recursos disponíveis. Uma diferença fundamental entre simulação e emulação é que a primeira é executada num tempo virtual simulado e a segunda deve ser executada em “tempo real”. Esse “tempo real” não se traduz numa temporização perfeita do que deveria acontecer realmente, mas numa abstração que, dependendo dos propósitos da simulação, tem que ter sua resolução ajustada. Outra importante diferença é que é impossível se atingir uma ordem totalmente passível de repetição nos eventos de uma emulação dado o componente real e imprevisível presente na rede. O uso de elementos reais junto com as simulações acaba permitindo:

- A validação de modelos de simulação contra carga de tráfego real;
- Validação de aplicações reais contra tráfego de fundo (“*background*”), potencialmente repetível, derivado de uma grande quantidade de modelos validados por simulação;
- Possibilidade de gerar topologias que seriam impossíveis só com elementos reais;
- Interação de aplicações reais com várias redes simuladas.

A integração da simulação com sistemas externos seria, portanto, uma característica importante para o framework de simulação sendo proposto. Para tratar este fato, está sendo desenvolvido um módulo no framework responsável por comunicação externa, capaz de passar informações relativas à simulação para ambientes externos. Inicialmente o módulo

está sendo implementado através de um agente em C++ que é executado numa *thread* do simulador e que fica escutando em um *socket* por chamadas externas. A implementação desse módulo é baseada nos estudos em (Braun, Staub, & Gantenbein, 2009), (Mayer & Gamer, 2008) e (Tüxen, Rüngeler, & Rathgeb, 2008).

Como as simulações do OMNeT++ executam em tempo discreto, um cuidado especial teve que ser tomado na implementação deste módulo, principalmente em relação à temporização do simulador, já que a simulação deve executar em tempo pseudo-real, ou seja, tentando emular o tempo real externo à simulação. Caso este cuidado não seja tomado, as informações passadas podem estar defasadas ou adiantadas e comprometer os dados sendo observados externamente.

Esse tipo de temporização é obtida pela inclusão de uma configuração nos cenários de simulação desenvolvidos no OMNeT++. O próprio simulador é responsável por verificar e manter a temporização. Quando se está simulando nesse modo não é possível utilizar a interface gráfica para visualizar a troca de mensagens nem a troca dinâmica de parâmetros da simulação. O ambiente deve ser configurado para executar no modo “*Express*”.

Dadas as aplicações planejadas inicialmente para o *framework* de simulação, principalmente a comunicação com sistemas de gerência, a solução usando conexões via *sockets* supre as demandas, tornando possível integrar a simulação com diferentes sistemas externos com o mínimo de esforço de codificação adicional. Além disso, testes simples demonstram que a simulação não só está rodando mais rápido que o tempo real, mas oferece uma resolução da ordem de 50ms. Isso é mais do que o necessário para o tipo de interação que está sendo planejado para o simulador (Canal de serviço e alarmes para serem usados na gerência).

Esses testes foram realizados para verificar como o ambiente iria se comportar com diferentes números de nós ópticos e diferentes quantidades de comprimentos de onda presentes em cada um desses nós.

O teste realizado foi simples: Foram criadas topologias aleatórias com os números de nós e comprimentos definidos, a simulação foi configurada para rodar em tempo real e então foi verificado o tempo máximo de diferença em relação ao tempo real. As topologias criadas são de dois tipos: em anel, ou em malha. Em ambos os casos o *framework* é responsável por criar a estrutura e conectar os nós a partir do número de nós e comprimentos de onda especificados em momento de execução. As três configurações de cenário usadas foram as seguintes: simples (5 nós e 40 comprimentos de onda), média (10 nós e 80 comprimentos de onda), complexo (20 nós e 160 comprimentos de onda).

Este teste não pretende ser um estudo formal do desempenho do *framework*, até porque uma parte dos componentes básicos ainda está em desenvolvimento e outra parte precisa de otimização. O estudo se apresenta mais como uma verificação da viabilidade de usar a modelagem atual na simulação de tempo real, permitindo assim a interação com componentes externos à simulação. Toda a parte de temporização é controlada pelo ambiente de simulação do OMNeT++ responsável por simulação de tempo real. Durante a execução desses testes o módulo responsável por esta funcionalidade estava em processo de atualização pelo time do OMNeT++, de modo que futuramente é esperado que não só funcione melhor, mas de forma mais otimizada, já que funcionalidade de processamento paralelo também estão sendo inseridas no módulo.

Cada uma das configurações foi testada por 10 simulações durante 300 segundos e os resultados aparecem na Tabela 5.1:

Tabela 5.1: Resultado dos testes simples de desempenho do protótipo do framework.

Cenário	<i>Maior diferença</i>	<i>Menor diferença</i>	<i>Média dos testes</i>
Simple	11ms	4ms	10ms
Médio	45ms	18ms	23ms
Complexo	129ms	32ms	48ms

Em cada um dos testes foi esperado um tempo de 10s para a simulação estabilizar em seguida foi gravado um log com a comparação entre o tempo de simulação e o tempo real. Esse log foi usado para criar a tabela 5.1. No início da simulação demora alguns segundos até que o sistema estabilize e a simulação consiga realmente seguir o tempo real. Após esse período foi anotada a maior variação entre os dois tempos em cada uma das simulações. Em geral, nos momentos de maior de processamento, a simulação acaba ficando um pouco atrasada em relação ao tempo real de modo que antes de conseguir restabelecer a sincronização as maiores diferenças alcançadas são medidas. Para cada um dos cenários a tabela 5.1 aponta a maior e a menor diferença encontradas na lista de maiores variações de cada simulação. Essas medidas mostram a maior variação obtida em cada execução, assim, quando a tabela mostra uma diferença de 4ms no cenário simples, isso significa que em todas as execuções do cenário simples teve uma execução onde a maior diferença entre o tempo real e o de simulação foi de 4ms, e além disso essa foi a menor diferença entre os dois em todas as execuções do cenário simples. A tabela 5.1 mostra também a média de maiores atrasos em todas as execuções de cada cenário.

Cabe ressaltar que os testes não foram realizados em um computador *desktop* dedicado ou especificamente montado para esse propósito. O computador de testes era um Intel Core 2 Duo de 2,66GHz, e os testes foram realizados no sistema operacional Windows que, em geral, não é a plataforma mais rápida para se rodar o OMNeT++.

Para exemplificar o uso do módulo de comunicação externa, duas pequenas aplicações foram desenvolvidas. Na primeira, informações relativas aos alarmes das camadas OTN são passadas para uma aplicação Java externa à simulação através de *sockets* em tempo real. Na segunda, um agente SNMP externo à simulação é alimentado de informações sobre a simulação. Desse modo, ele consegue prover informações, por exemplo, a uma interface web SNMP (Cacti), sobre dados relevantes da rede óptica sendo simulada.

A Figura 5.5 mostra os gráficos gerados no Cacti a partir dos dados coletados na simulação. Um agente SNMP foi criado de modo a se conectar com o módulo de comunicação externa e manter as informações atualizadas. A interface do Cacti consulta esse agente periodicamente e gera os gráficos mostrados na figura 5.3.

Para esse teste foi montada uma MIB simples que só continha três campos: número de alarmes dLOS-P disparados, potência máxima e potência mínima do sinal óptico detectado na primeira camada OTN.

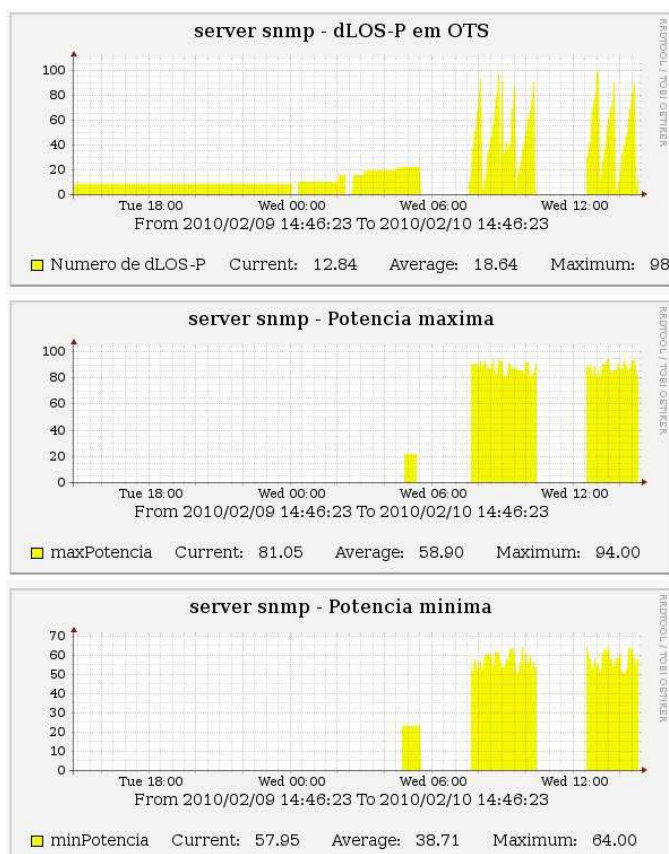


Figura 5.5: Resultados de consulta SNMP a um roteador da simulação na interface do Cacti.

Também foi criado um aplicativo Java que se conecta ao módulo de comunicação externa e é atualizado a cada alarme disparado na simulação. A Figura 5.6 mostra a interface do aplicativo.

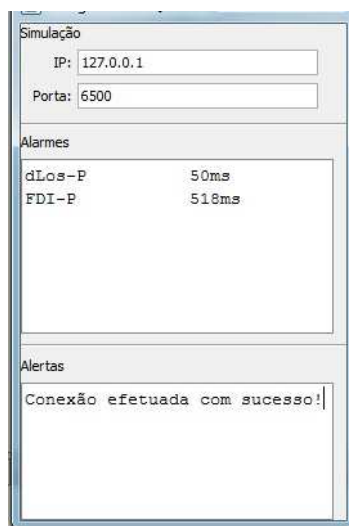


Figura 5.6: Interface Java para conexão com a simulação.

No exemplo da figura 5.4, após ser realizada a conexão com a simulação, dois alarmes foram disparados: dLOS-P e FDI-P. Os tempos em que os alarmes foram disparados também aparecem.

5.3 Conclusão

Este capítulo tem como objetivo mostrar a viabilidade da arquitetura de framework proposta no capítulo anterior. Apesar de após a integração dos componentes básicos o protótipo do *framework* ainda não estar 100% completo é possível observar pelos testes demonstrados que já atinge muitos de seus objetivos iniciais.

Além disso, é demonstrado um módulo de comunicação externa que futuramente vai agregar um número de funcionalidades ao *framework*. Entre as funcionalidades pretendidas está a possibilidade de utilizar o framework proposto para validar protótipos de módulos de planos de gerência e controle que poderiam se comunicar com a simulação como se estivessem interagindo com redes de transporte reais.

Diversos projetos estão sendo desenvolvidos utilizando como base o protótipo e à medida que a plataforma amadurece é esperado que, cada vez mais, a arquitetura proposta seja usada para a simulação de redes OTN e possa auxiliar na modelagem e análise de novas idéias e projetos.

Capítulo 6: Conclusão

Dado o funcionamento atual do *framework* e os resultados preliminares dos outros trabalhos que estão sendo realizados com base no *framework*, e que foram apresentados no capítulo anterior, pode-se afirmar que o objetivo principal do trabalho foi alcançado com sucesso.

Nesta dissertação é apresentada uma proposta de *framework* para a simulação de redes OTN, genérico, independente de fabricantes e aberto. São especificados o ambiente de simulação adotado, o tipo de abstração utilizada, a estrutura geral da arquitetura e a modelagem dos principais componentes.

A arquitetura está focada principalmente na descrição funcional e interação dos componentes ópticos OTN e as normas e especificações da ITU-T. Além disso, a estrutura proposta é flexível e extremamente modular de modo a permitir a fácil adição de novos componentes. A arquitetura proposta é resultado da integração e adaptação de vários trabalhos que estão sendo desenvolvidos pelo grupo de pesquisa, em paralelo.

Essa integração não é fácil e um trabalho extenso de comparação e análise de código faz-se necessário para validar o que já estava construído. Isso para permitir não só a interoperabilidade dos módulos que já estavam prontos, mas a conformidade com a estrutura modular da arquitetura proposta de modo que, futuramente, módulos adicionais possam ser facilmente integrados ao *framework*.

Um estudo aprofundado dos ambientes de simulação disponíveis e de trabalhos relacionados foi feito para que as soluções propostas fossem as mais adequadas, dados os objetivos do trabalho. Durante as pesquisas de trabalhos relacionados também foi verificado que poucos trabalhos na área de simulações ópticas levam em consideração as recomendações da ITU e a maioria das soluções encontradas é genérica demais para ser utilizada na simulação de redes OTN.

Nos capítulos 4 e 5 um protótipo do *framework* é apresentado e alguns testes preliminares são feitos para demonstrar a sua conformidade com o proposto no início do desenvolvimento do projeto. São apresentados também alguns resultados preliminares de pesquisas que estão em desenvolvimento e que foram possíveis graças à utilização do *framework*.

A partir dos testes é possível observar que o protótipo do *framework* é escalável, pelo menos nas situações propostas até o momento. A estrutura modular do *framework*, além do

uso do ambiente de simulações do OMNeT++ e de suas ferramentas permitem afirmar que o sistema é flexível. Além disso, com a modelagem funcional totalmente baseada nas especificações da ITU é esperado que o realismo da simulação não esteja comprometido. Deste modo, os três desafios propostos no início do desenvolvimento do framework são atingidos.

Com o uso do *framework* está sendo possível validar uma série de propostas de novas tecnologias que futuramente podem vir a ser utilizadas com equipamentos OTN reais. Conforme a plataforma amadurece é esperado que daqui em diante o *framework* possa ser utilizado, cada vez mais, como um ambiente centralizado e único que permita aos pesquisadores modelar, analisar e validar projetos de redes OTN.

Na Tabela 6.1 é apresentado um resumo do estado atual dos módulos do *framework* conforme o que foi apresentado no Capítulo 4:

Tabela 6.1: Estado atual dos módulos do framework.

Módulo	Estado atual
Estrutura do <i>framework</i>	A estrutura em desenvolvimento tem passado por alguns ajustes, mas até o momento está seguindo a base da estrutura proposta nesta dissertação;
Camadas OTN	As camadas OTN estão praticamente completas. Uma nova interface para a montagem dos equipamentos está em estudo e os blocos usados para representar as camadas foram alterados para refletir o que está definido nas recomendações;
Gerador de tráfego	Geradores de tráfego estão em desenvolvimento;
Módulos extras	Novos módulos estão em desenvolvimento e com a padronização da API da gerência local se espera que a integração seja direta.

6.1 Contribuições e Resultados

As principais contribuições desta dissertação estão relacionadas com a descrição da arquitetura do *framework*, integração dos seus componentes e definição das relações e interfaces entre esses componentes. A estrutura modular proposta permite a rápida integração de novos módulos.

Tomando como base os estudos realizados para a confecção desta dissertação podem ser listados como principais resultados:

- Definição de uma arquitetura modular baseada em componentes e focada principalmente na descrição funcional dos componentes ópticos. Apresentação da modelagem e interação dos seus principais componentes.
- Desenvolvimento de um primeiro protótipo do *framework* de modo a verificar o que foi proposto.

- Integração de diversos trabalhos relacionados às redes OTN que estavam em desenvolvimento pelo grupo de pesquisa em telecomunicações da UFES e que, até o momento, estavam funcionando de forma separada.
- Construção de um módulo extra que permite a comunicação com sistemas externos à simulação. Deste modo é esperado que seja possível testar protótipos de sistemas que de outro modo precisariam de redes físicas reais para serem avaliados.

6.2 Dificuldades

Como apontado no Capítulo 4: uma das dificuldades encontradas na definição e implementação do *framework* é como manter a representação gráfica do *framework* consistente com o que está definido nas recomendações da ITU. Além disso, como garantir que os equipamentos OTN construídos pelos usuários continuam válidos e seguindo as recomendações, ou seja, que as camadas OTN foram interconectadas corretamente. Para remediar esse fato, está em desenvolvimento uma solução que pretende usar uma ontologia de redes OTN desenvolvida pelo grupo para validar tanto os equipamentos presentes no *framework* quanto os cenários de simulação que podem ser criados (Monteiro M. , Garcia, Barcelos, & Guizzardi, 2010), (Monteiro M. , Garcia, Segatto, Barcelos, & Simões, 2009), (Barcelos, 2009).

Outra dificuldade encontrada é com relação à validação do que está sendo proposto nesta dissertação. Embora o *framework* e a modelagem dos equipamentos tenham sido construídos com base nas recomendações da ITU, não existem implementações de referência que poderiam ser usadas para verificar a validade do que está sendo proposto. No Capítulo 5 é apresentada a forma usada no contexto desta dissertação para contornar esse problema.

6.3 Trabalhos Futuros

No momento está sendo trabalhada a integração à estrutura do *framework* de novos módulos extras sendo desenvolvidos pelo grupo de pesquisa, de modo a inserir novas funcionalidades aos cenários de simulação que podem ser construídos no *framework*. Adicionalmente, as implementações dos módulos extras existentes (*Discovery*, plano de controle GMPLS, comunicação externa) estão sendo otimizadas para trabalhar com a nova arquitetura do *framework* proposta nesta dissertação.

As camadas OTN foram modeladas conforme os blocos funcionais definidos na G.798, desse modo tanto o payload quanto o overhead são tratados no mesmo módulo em cada camada. Está sendo verificada a possibilidade de se dividir cada um desses módulos em dois. Assim, cada um desses novos módulos, vai tratar ou do *payload* ou do *overhead* associado àquela camada. Devido à constatação que na maioria dos equipamentos ópticos reais os caminhos de dados dentro do equipamento também são divididos em *payload* e *overhead*, essa alteração é sugerida. Desse modo, espera-se que a modelagem dos equipamentos reais seja facilitada.

Outro ponto que está em desenvolvimento é a interação com os sinais clientes. Inicialmente estão sendo criados mais tipos de sinais clientes no gerador de tráfego. Além disso, a interface que recebe o sinal cliente está sendo padronizada, conforme comentado na seção 4.1.3 de modo que futuramente é esperado que qualquer sinal cliente gerado pelo OMNeT++ possa ser facilmente integrado às simulações OTN.

No momento o *framework* ainda não está trabalhando com a comutação dinâmica de pacotes ópticos de modo que o tráfego de dados é praticamente constante. Numa versão futura onde este tipo de comutação se torne uma realidade é necessário que as fontes de tráfego sejam mais bem modeladas. Isso não é difícil já que o OMNeT++ fornece fontes de tráfego como CBR, Exponencial e Pareto. Essas só precisam ser adaptadas para o cenário óptico WDM.

Finalmente está sendo verificada a possibilidade de se criar uma *OpticalMsg*, que seria responsável por encapsular as *OTNmsg* de cada comprimento de onda e representar um sinal óptico multiplexado. Teoricamente, essa alteração iria diminuir o número de mensagens trafegadas durante a simulação e desse modo melhorar o desempenho dos cenários sendo simulados.

Referências Bibliográficas

- Alberti, A. (24 de Abril de 2003). Desenvolvimento de Modelos de Simulação para a Análise de Qualidade de Serviço em Redes ATM. Campinas, São Paulo, Brasil: Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas.
- Aloia, E. J. (2009). Contribuições para a análise e simulação de redes ópticas: Aspectos de engenharia de tráfego, restauração dinâmica e conversão de comprimentos de onda. São Carlos, São Paulo.
- Barcelos, P. (2009). *Uma Proposta de Modelo de Ontologia para Arquitetura de Redes Ópticas de Transporte*. Vitória: Universidade Federal do Espírito Santo.
- Begg, L., Liu, W., Pawlikowski, S., Perera, S., & Sirisena, H. (2006). *Survey of simulators of Next Generation Networks for Studying Service Availability and Resilience*. Department of Computer Science and Software Engineering, University of Canterbury.
- Binh, L. N., Binh, L. H., & Tu, V. T. (2009). Routing and Wavelength Assignment and Survivability of Optical Channels in Ultra-high Speed IP over DWDM Networks Under Constraints of Residual Dispersion and Nonlinear Effects. *International Journal of Computer Science and Network Security (IJCSNS)* , 12.
- Braun, T., Staub, T., & Gantenbein, R. (2009). VirtualMesh: An Emulation Framework for Wireless Mesh Networks in OMNeT++. *Proceedings of the 2nd International Workshop on OMNeT++* .
- Bueno-delgado, M. V., Veiga-gontán, J. A., Pavón-mariño, P., & García-haro, J. (Outubro de 2005). oPASS: A simulation tool for the performance evaluation of Optical Packet Switching architectures. *Proc. of the European Symposium on Simulation Tools for Research and Education in Optical Network (STREON 2005)* .
- C. Santos, G., D. Santos, F., & C. R. F. Oliveira, J. (02 de setembro de 2008). Aprovisionamento Automático de Circuitos Ópticos via Plano de Controle GMPLS aplicado a uma Rede Óptica Reconfigurável Baseada em ROADMs. *Simpósio Brasileiro de Telecomunicações (SBRT)* , p. 4.
- Chwif, L. (1999). Redução de modelos de simulação de eventos discretos na sua concepção: Uma abordagem casual. São Paulo.

- Cieutat, L., & Binh, L. (2003). *Routing and Wavelength Assignment in GMPLS-based DWDM Optical Networks: An OMNeT++ modelling Platform*. Melbourne: MONASH UNIVERSITY.
- Dias, K. L., Dias, K. L., Abelem, A. J., & Dias, K. L. (2008). Uma Proposta de Implementação de um Módulo de Re-roteamento Dinâmico de Túneis LSPs para a Ferramenta de Simulação GLASS. *IX Workshop de Software Livre*. Porto Alegre.
- ECI Telecom. (Julho de 2008). White Paper. *Next Generation OTN*.
- EXFO. (2006). Application Note. *The G.709 Optical Transport Network - An Overview*. Canadá.
- F.F.Ferrari. (2009). *Simulação da Funcionalidade de Descoberta Automática Aplicada às Redes OTN*. Vitória: Universidade Federal do Espírito Santo.
- F.P.Favoreto, A.S.Garcia, M.T.A.Torres, R.S.Tessinari, & Segatto, M. E. (Setembro de 2009). Plano de controle GMPLS para redes ópticas de transporte. *SBrT 2009*.
- Farrel, A., & Bryskin, I. (2006). *GMPLS: Architecture and Applications*. San Francisco: Elsevier.
- Favoreto, F. (Setembro de 2009). Plano de Controle GMPLS para redes Ópticas de Transporte. *Dissertação de Mestrado*. Vitória, ES, Brasil: Programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo.
- Ferrari, F., Frasson, A., & Garcia, A. (2010). Descoberta automática em redes ópticas de transporte. *MOMAG 2010*. Vitória.
- Frigini, F. (2010). *Modelagem das camadas digitais de uma rede óptica de transporte no simulador OMNeT++ de acordo com a recomendação ITU-T G.798*. Vitória: Universidade Federal do Espírito Santo.
- Gamer, T., & Scharf, M. (2008). Realistic simulation environments for IP-based networks. *Simutools '08: Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems*, pp. 1-7.
- Garrido, P., Malumbres, M., & Calafate, C. (3-7 de Março de 2008). ns-2 vs. OPNET: a comparative study of the IEEE 802.11e technology on MANET environments. *First*

International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (SIMUTools'08) .

Ilyas, M., & Mouftah, H. T. (2003). *The Handbook of Optical Communication Networks*. Boca Raton: CRS Press.

Iniewski, K., McCrosky, C., & Minoli, D. (2008). *Network Infrastructure And Architecture*. New Jersey: Wiley Interscience.

ITU-T, T. S. (Novembro de 2009). G.695. *Optical interfaces for coarse wavelength division multiplexing applications* .

ITU-T, T. S. (16 de Março de 2003). G.709. *Interfaces for the Optical Transport Network (OTN)* .

ITU-T, T. S. (Dezembro de 2009). G.709. *Interfaces for the Optical Transport Network (OTN)* .

ITU-T, T. S. (2005, Agosto). G.7714. *Generalized Automatic Discovery for Transport Entities* .

ITU-T, T. S. (2003, Abril). G.7714.1. *Protocol for Automatic Discovery in SDH and OTN Networks* .

ITU-T, T. S. (14 de Dezembro de 2006). G.798. *Characteristics of optical transport network hierarchy equipment functional blocks* .

ITU-T, T. S. (2000, março). G.805. *Generic functional architecture of transport networks* .

ITU-T, T. S. (2001, Novembro). G.872. *Architecture of Optical Transport Networks* .

ITU-T, T. S. (29 de Março de 2008). G.874. *Management aspects of the optical transport network element* .

ITU-T, T. S. (Novembro de 2009). G.959.1. *Optical transport network physical layer interfaces* .

ITU-T, T. S. (Dezembro de 2008). G-Sup.43. *Transport of IEEE 10GBASE-R in optical transport networks (OTN)* .

Janota, V. (28 de julho de 2005). *Traffic engineering in modern switched networks*. Praga, República Tcheca.

- Jeruchim, M. C., Balaban, P., & Shanmugan, K. S. (2002). *Simulation of Communication Systems: Modeling, Methodology and Techniques*. New York, Boston, Dordrecht, London, Moscow: Kluwer Academic Publishers.
- Kazi, K. (2006). *Optical Networking Standards - A Comprehensive Guide for Professionals*. Springer.
- Köksal, M. (2008). *A Survey of Network Simulators Supporting Wireless Networks*. Ankara: Department of Computer Science, Middle East Technical University.
- Kozlovsky, M., Berceci, T., & Kozlovsky, V. (2007). NSOSS: the non-synchronized optical switch simulator. *ValueTools '07: Proceedings of the 2nd international conference on Performance evaluation methodologies and tools* , pp. 1-6.
- Kumar, U. (2002). *Optical Transport Networks*. China.
- Mayer, C., & Gamer, T. (2008). *Integrating real world applications into OMNeT++*. Institute of Telematics, Universitat Karlsruhe.
- Monteiro, M., Garcia, A. S., Barcelos, P. P., & Guizzardi, G. (2010). Ontology based model for the ITU-T recommendation G.805: towards the self-management of transport networks. *International Journal of Computer Science and Information Technology* , pp. 155-170.
- Monteiro, M., Garcia, A., Segatto, M., Barcelos, P., & Simões, R. (2009). OOTN -An Ontology Proposal for Optical Transport Networks. *International Workshop on Management of Emerging Networks and Services, MENS 2009* (pp. 1-7). Saint-Petersburg: ICUMT 2009 and Workshops Proceedings.
- Nance, R. (1993). A history of discrete event simulation programming languages. *HOPL-II: The second ACM SIGPLAN conference on History of programming languages* , pp. 149--175.
- Orfanus, D., Lessmann, J., Janacik, P., & Lachev, L. (2008). Performance of wireless network simulators: a case study. *PM2HW2N '08: Proceedings of the 3rd ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks* , pp. 59-66.
- Perez, J. (2005). *MQTT Performance Analysis with OMNeT++*. Institut Eurécom, Zurich: IBM Zurich Research Laboratory.

Perros, H. G. (2005). *Connection-oriented networks*. John Wiley & Sons Ltd.,

Ramaswami, R., & Sivarajan, K. N. (2002). *Optical Networks - A Practical Perspective*. San Francisco: Morgan Kaufmann.

Schoch, E., Feiri, M., Kargl, F., & Weber, M. (2008). Simulation of ad hoc networks: ns-2 compared to JiST/SWANS. *Simutools '08: Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems*, pp. 1-8.

Site DML. (s.d.). Acesso em 24 de 08 de 2010, disponível em Domain Modeling Language (DML) Reference Manual: <http://www.ssfnet.org/SSFdocs/dmlReference.html>

Site do Eclipse. (s.d.). Acesso em 28 de 08 de 2010, disponível em Eclipse.org home: <http://www.eclipse.org/>

Site do GLASS. (s.d.). Acesso em 24 de 08 de 2010, disponível em GMPLS Lightwave Agile Switching Simulator: <http://www-x.antd.nist.gov/glass/Main.htm>

site do JiST. (s.d.). Acesso em 24 de 08 de 2010, disponível em JiST - Java in Simulation Time / Scalable Wireless Ad hoc Network Simulator : <http://jist.ece.cornell.edu/>

Site do J-Sim. (s.d.). Acesso em 24 de 08 de 2010, disponível em J-Sim Official: <http://sites.google.com/site/jsimofficial/>

Site do ns-2. (s.d.). Acesso em 24 de 08 de 2010, disponível em The Network Simulator - ns-2: <http://www.isi.edu/nsnam/ns/>

site do ns-3. (s.d.). Acesso em 24 de 08 de 2010, disponível em The ns-3 network simulator: <http://www.nsnam.org/>

Site do OMNeT++. (s.d.). Acesso em 24 de 08 de 2010, disponível em OMNeT++ Community Site: <http://www.omnetpp.org/>

Site do OPNET. (s.d.). Acesso em 24 de 08 de 2010, disponível em Application and Network Performance with OPNET | Monitoring, Troubleshooting, Auditing, and Prediction: <http://www.opnet.com/>

Site do SSF. (s.d.). Acesso em 24 de 08 de 2010, disponível em Scalable Simulation Framework: <http://www.ssfnet.org/homePage.html>

- Tessinari, R. (2009). *Mapeamento de Equipamentos Ópticos e Modelagem de Redes OTN no Simulador OMNeT++ de acordo com a Recomendação ITU-T G.798*. Vitória: Universidade Federal do Espírito Santo.
- TPACK A/S. (Junho de 2008). *P-OTN: Packet Optical Network Transformation*. Fonte: www.tpack.com
- Tüxen, M., Rüngeler, I., & Rathgeb, E. (2008). Interface connecting the INET simulation framework with the real world. *Simutools '08: Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops* , pp. 1-6.
- Varga, A. (6 de Junho de 2001). The OMNeT++ Discrete Event Simulation System. *Proceedings of the European Simulation Multiconference* , pp. 319-324.
- Varga, A., & Acadêmica, C. (s.d.). *INET Framework main page*. Acesso em 06 de agosto de 2009, disponível em Site oficial do INET.
- Varga, A., & Hornig, R. (3-7 de Março de 2008). An overview of the OMNeT++ simulation environment. *SIMUTools* .
- Vásquez Vásquez, D. E. (2009). *Study of optical transport networks (OTN) as a multiservice networking platform*.
- Weingartner, E., vom Lehn, H., & Wehrle, K. (14-18 de Junho de 2009). A Performance Comparison of Recent Network Simulators. *IEEE International Conference on Communications, ICC '09* . , pp. 1-5.