



Marcos Antônio de Siqueira

# **Redes Ópticas de Transporte Definidas por Software com Suporte à Virtualização e Operação Autônoma com Base em Políticas**

Campinas

2015





UNIVERSIDADE ESTADUAL DE CAMPINAS  
Faculdade de Engenharia Elétrica e de Computação

Marcos Antônio de Siqueira

# **Redes Ópticas de Transporte Definidas por Software com Suporte à Virtualização e Operação Autônoma com Base em Políticas**

Tese apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Doutor em Engenharia Elétrica, na Área de Engenharia da Computação

Orientador: Prof. Dr. Christian Rodolfo Esteve Rothenberg

Este exemplar corresponde à versão da tese defendida pelo aluno Marcos Antônio de Siqueira, e orientada pelo Prof. Dr. Christian Rodolfo Esteve Rothenberg

---

Campinas

2015

Ficha catalográfica  
Universidade Estadual de Campinas  
Biblioteca da Área de Engenharia e Arquitetura  
Elizangela Aparecida dos Santos Souza - CRB 8/8098

Si75r Siqueira, Marcos Antônio de, 1978-  
Redes ópticas de transporte definidas por software com suporte à virtualização e operação autônoma com base em políticas / Marcos Antônio de Siqueira. – Campinas, SP : [s.n.], 2015.

Orientador: Christian Rodolfo Esteve Rothenberg.  
Tese (doutorado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Redes ópticas elásticas. 2. Arquitetura de redes. 3. Virtualização de redes. 4. Políticas. I. Esteve Rothenberg, Christian Rodolfo, 1982-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Informações para Biblioteca Digital

**Título em outro idioma:** Software defined transport optical networks with support for virtualization and policy-based autonomic operation

**Palavras-chave em inglês:**

Elastic optical networks

Network architecture

Network virtualization

Policies

**Área de concentração:** Engenharia de Computação

**Titulação:** Doutor em Engenharia Elétrica

**Banca examinadora:**

Christian Rodolfo Esteve Rothenberg [Orientador]

Fábio Luciano Verdi

Moisés Renato Nunes Ribeiro

Eleri Cardozo

Darli Augusto de Arruda Mello

**Data de defesa:** 10-04-2015

**Programa de Pós-Graduação:** Engenharia Elétrica

## COMISSÃO JULGADORA - TESE DE DOUTORADO

**Candidato:** Marcos Antônio de Siqueira

**Data da Defesa:** 10 de abril de 2015

**Título da Tese:** "Redes Ópticas de Transporte Definidas por Software com Suporte à Virtualização e Operação Autônoma com Base em Políticas"

Prof. Dr. Christian Rodolfo Esteve Bothenberg (Presidente): \_\_\_\_\_

Prof. Dr. Fábio Luciano Verdi: \_\_\_\_\_

Prof. Dr. Moisés Renato Nunes Ribeiro: \_\_\_\_\_

Prof. Dr. Eleri Cardozo: \_\_\_\_\_

Prof. Dr. Darli Augusto de Arruda Mello: \_\_\_\_\_



# Resumo

Esta tese apresenta uma proposta de arquitetura para controle de redes ópticas de transporte que utiliza o paradigma de redes definidas por software, com suporte a operação autônoma com base em políticas. A arquitetura é constituída pelos seguintes pilares: (i) modelagem dos elementos de rede, incluindo suas interconexões, restrições e capacidades, utilizando a linguagem YANG; (ii) composição dos modelos dos elementos de rede e suas relações em um modelo que representa a rede, suportando transformações para representação da rede como grafos de propriedades; e (iii) um modelo de políticas baseado em objetos associados ao grafo de propriedades da rede que viabiliza a operação autônoma do controlador. A proposta foi validada através de provas de conceito realizadas por simulações, protótipos e experimentos, incluindo casos de uso de segmentação e virtualização da rede óptica de transporte, aplicações SDN para ajuste de parâmetros operacionais da rede com base em políticas, bem como a operação autônoma do controlador SDN com auxílio de ferramentas de simulação com rotinas de planejamento automatizado.

**Palavras-chaves:** SDN; Redes Ópticas de Transporte; Políticas.

# Abstract

This thesis proposes an architecture for optical transport networks control, using the software defined networking paradigm, with support for policy-based autonomic operation. The architecture is composed of three pillars: (i) modeling of network elements, its interconnections, constraints and capabilities using the YANG language; (ii) composition of the network element models and its interconnections forming a network model, supporting transformations for representing the network as property graphs; and (iii) a policy model based on objects associated to the network graph designed for allowing autonomic operation of the network controller. The proposal has been validated through a set of proofs of concept performed via simulations, prototypes and experiments, including use cases for optical transport network slicing and virtualization, SDN applications for policy-based operational parameters adjustment, and autonomic operation of the SDN controller assisted by simulation tools with routines for automated planning.

**Keywords:** SDN; Optical Transport Networks; Policies.



# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Desafios e definição do problema	3
1.2	Contribuições	4
1.3	Trabalhos relacionados	5
1.3.1	SDN em redes ópticas de transporte	5
1.3.2	Virtualização de redes ópticas de transporte	7
1.3.3	Políticas em SDN	7
1.3.4	Planejamento de redes ópticas multicamadas	8
1.4	Organização da Tese	9
<b>2</b>	<b>Redes Ópticas de Transporte de Nova Geração (NG-OTN)</b>	<b>11</b>
2.1	Evolução das redes ópticas de transporte	11
2.1.1	Equipamentos de comutação em redes ópticas	11
2.1.1.1	Comutação óptica sem conversão <i>Optical-Electrical-Optical</i> (OEO)	11
2.1.1.2	Comutação óptica com conversão OEO parcial	16
2.1.1.3	Comutação em redes ópticas de transporte com conversão OEO	17
2.1.2	Plano de controle GMPLS	18
2.1.2.1	Proteção e restauração em redes GMPLS	19
2.1.2.2	GMPLS em redes multicamadas e multi regiões	21
2.1.3	PCE ( <i>Path Computation Element</i> )	23
2.1.3.1	Descoberta de PCEs e balanceamento de carga	25
2.1.3.2	Sincronização da TED	25
2.1.3.3	Manutenção de estado pelo PCE	26
2.1.3.4	<i>PCE Communication Protocol</i> (PCEP)	26
2.1.3.5	Características e requisitos específicos para redes GMPLS	28
<b>3</b>	<b>Novos Paradigmas em Controle de Redes</b>	<b>29</b>
3.1	Redes autonômicas	29
3.1.1	Computação autonômica	29
3.1.2	Redes autonômicas	30
3.1.3	Redes ópticas cognitivas	31
3.2	Gerência de redes com base em políticas	32
3.2.1	Arquitetura de referência para sistemas PBNM	33
3.2.2	Linguagens para representação de políticas	34

3.3	Redes definidas por software (SDN) . . . . .	37
3.3.1	Controladores SDN . . . . .	39
3.3.1.1	OpenDayLight . . . . .	41
3.3.1.2	Projeto TeraStream . . . . .	41
3.3.2	<i>Network Configuration Protocol</i> (NETCONF) como protocolo em redes SDN . . . . .	43
3.3.3	Atividades relacionadas à T-SDN no OIF . . . . .	45
3.3.3.1	Arquitetura de políticas em SDN com mecanismo declarativo . . . . .	46
3.4	Virtualização de redes . . . . .	47
3.4.1	Virtualização de redes ópticas de transporte . . . . .	50
3.4.2	Virtualização de funções de redes . . . . .	51
3.4.2.1	Arquitetura de referência para NFV . . . . .	52
3.4.2.2	Composição de serviços de redes . . . . .	54
3.4.3	Virtualização de funções de redes de transporte . . . . .	54
<b>4</b>	<b>Arquitetura SDN para NG-OTN com Base em Políticas . . . . .</b>	<b>57</b>
4.1	Objetivos . . . . .	57
4.2	Requisitos e decisões de projeto . . . . .	58
4.2.1	Requisitos da solução . . . . .	58
4.2.2	Decisões de projeto . . . . .	60
4.3	Arquitetura proposta . . . . .	62
4.3.1	Arquitetura de implementação do T-SDNC . . . . .	63
4.3.2	Posicionamento da arquitetura proposta frente a esforços de padronização . . . . .	66
4.4	Aplicações SDN . . . . .	66
4.4.1	Plano de controle GMPLS virtualizado como aplicação SDN . . . . .	66
4.4.2	Plano de controle baseado em SDN . . . . .	67
4.5	Modelo de políticas para <i>Transport SDN</i> (T-SDN) . . . . .	68
4.5.1	Arquitetura de políticas proposta . . . . .	69
4.5.1.1	Modelo de políticas . . . . .	69
4.5.1.2	<i>Policy flow</i> . . . . .	71
4.5.1.3	Modelo de políticas para SDN baseado em grafos . . . . .	74
4.5.2	Planejamento automatizado como suporte à análise de políticas . . . . .	76
<b>5</b>	<b>Aplicações e Casos de Uso . . . . .</b>	<b>79</b>
5.1	Ambiente experimental . . . . .	79
5.2	UC01 – Virtualização da rede com suporte a GMPLS e instanciação automática de <i>Virtual Optical Networks</i> (VONs) . . . . .	80
5.3	UC02 – Reconfiguração dinâmica de VON com base em políticas . . . . .	87
5.4	UC03 – Aplicação SDN para reconfiguração dinâmica do formato de modulação . . . . .	89

5.5	UC04 – Plano de controle SDN com suporte a NEs flexíveis . . . . .	91
5.6	UC05 – Ajuste dinâmico de políticas de sobrevivência através de planejamento em operação . . . . .	98
5.7	UC06 – Ajuste fino de políticas locais via planejamento automatizado . . . . .	106
	<b>Conclusão . . . . .</b>	<b>111</b>
	<b>Referências . . . . .</b>	<b>113</b>
	<b>Anexos . . . . .</b>	<b>125</b>
	<b>ANEXO A Publicações do autor relevantes para esta tese . . . . .</b>	<b>127</b>
	<b>ANEXO B Logs de execução da ferramenta de planejamento de redes multicamadas . . . . .</b>	<b>131</b>
B.1	Fase 1 - Análise da Topologia . . . . .	131
B.2	Fase 2 - Dimensionamento da Rede . . . . .	131



*À minha amada esposa, filhos e pais.*



# Agradecimentos

Agradeço a meus familiares pelo apoio incondicional.

Agradeço aos professores Dr. Maurício Ferreira Magalhães e Dr. Christian Rodolfo Esteve Rothenberg, do Mestrado e Doutorado respectivamente, pela atuação como mentores e especialmente pela amizade e confiança.

Agradeço aos autores e co-autores que participaram em parceria das publicações que serviram como base para esta tese: Adelmo Avancini, Ademilson Silva, Alberto Hirata, Aldario Bordonalli, Antonio Sachs, Christian Rothenberg, Diego Macedo, Dimitra Simeonidou, Eduardo Magalhães, Edmundo Madeira, Eleri Cardozo, Fabian Hooft, Fabio Verdi, Fabricio Figueiredo, Fabrizio Lacerda, Flavia Rocha, Georgios Zervas, Giovanni Curiel, Heitor Carvalho, João Januário, José Martins, Joselis Oliveira, Juliano Oliveira, Julio Magro, Julio Oliveira, Lavoisier Farias, Leonardo Marioti, Luiz Carvalho, Mamede Lima-Marques, Marcel Castro, Marcelo Nascimento, Marcio Colazza, Marcos Salvador, Maurício Magalhães, Miquel Garrich, Nadia Nassif, Neil Guerrero, Norberto Amaya, Rafael Duarte, Rafael Pasquini, Rafael Scaraficci, Raulison Resende e Sandro Rossi.

Finalmente agradeço a todos que, de alguma forma, contribuíram para com este trabalho, incluindo os membros da banca de qualificação e defesa, os professores das disciplinas que cursei na FEEC, colegas de trabalho e do laboratório.



*“Simplicity is a great virtue but it requires hard work to achieve it and education to appreciate it. And to make matters worse: complexity sells better.”*  
*(Edsger W. Dijkstra)*



# Lista de ilustrações

Figura 1 – Visão de alto nível da arquitetura proposta para o controlador SDN para redes de transporte . . . . .	4
Figura 2 – ROADM WSS de primeira geração . . . . .	13
Figura 3 – ROADM WSS CDC . . . . .	14
Figura 4 – Arquiteturas de ROADM WSS do tipo CDC . . . . .	15
Figura 5 – ROADM WSS CDC com rotas alternativas para resolução de contenção .	16
Figura 6 – Modelo de referência para comutador OTN baseado em matriz de células	18
Figura 7 – Possíveis arquiteturas para solução PCE . . . . .	24
Figura 8 – Arquitetura de referência para sistema PBNM . . . . .	34
Figura 9 – Modelo PCIM . . . . .	36
Figura 10 – Uma visão simplificada da arquitetura SDN . . . . .	38
Figura 11 – Principais primitivas do OpenFlow 1.0 . . . . .	39
Figura 12 – Arquitetura de referência para controlador SDN, adaptado de Nadeau e Gray (2013) . . . . .	40
Figura 13 – Arquitetura do projeto OpenDaylight, versão Helium (OPENDAYLIGHT, 2015) . . . . .	42
Figura 14 – Arquitetura SDN do projeto TeraStream . . . . .	43
Figura 15 – Arquitetura SDN para redes de Transporte, OIF . . . . .	45
Figura 16 – Arquitetura de Referência para NFV. Fonte: [R]. . . . .	53
Figura 17 – Exemplo de um serviço de rede fim-a-fim com VNFs e VNF-FGs aninhados. Fonte: [R]. . . . .	55
Figura 18 – Arquitetura do NCA . . . . .	61
Figura 19 – Arquitetura do Sistema. Adaptado de [A] . . . . .	63
Figura 20 – Arquitetura de controle do ROADM baseada no NCA . . . . .	64
Figura 21 – Arquitetura de implementação do controlador T-SDN baseada no NCA .	65
Figura 22 – Fluxo de avaliação de política. Fonte: [C]. . . . .	72
Figura 23 – Exemplo de operação do <i>policy flow</i> . Fonte: [C]. . . . .	73
Figura 24 – Exemplo de um modelo de políticas para equalização. Fonte: [A]. . . . .	75
Figura 25 – <i>Testbed</i> composto por cinco ROADMs WSS. Fonte: [B]. . . . .	80
Figura 26 – Pacotes LLDP e modelo YANG. Fonte: [B]. . . . .	82
Figura 27 – VONs principal e HS, seus respectivos NEs, TE-Links e mapeamento. Fonte: [B]. . . . .	83
Figura 28 – Mensagens RSVP durante estabelecimento de <i>lightpath</i> . Fonte: [B]. . . . .	84

Figura 29 – Cross-conexões no ROADM por VON/NE. Fonte: [B]. . . . .	85
Figura 30 – Espectro óptico nos NEs 01, 02 e 04. Fonte: [B]. . . . .	86
Figura 31 – Reconfiguração da VON após falha em enlace na rede física. Fonte: [A]. . . . .	88
Figura 32 – Experimento com Transponder Flexível. Fonte: [E]. . . . .	90
Figura 33 – Arquitetura proposta para elemento de rede SD-POSS. Fonte: [D]. . . . .	93
Figura 34 – Grafo completo ilustrando a rede de referência e o caminho calculado entre SPO e SSA. Fonte: [D]. . . . .	95
Figura 35 – Demandas fictícias (Ex.: entre <i>data centers</i> ou tráfego Internet) . . . . .	97
Figura 36 – Topologia de sítios de cabos ópticos de referência . . . . .	100
Figura 37 – Topologia do <i>Backbone</i> IP (demandas) . . . . .	101
Figura 38 – Topologia da rede de OTN <i>Switches</i> . . . . .	102
Figura 39 – Topologia de cabos ópticos readequada . . . . .	103
Figura 40 – Topologia da rede de OTN <i>Switches</i> readequada . . . . .	104
Figura 41 – Grafo da rede gerado pela ferramenta <i>Network Planner</i> . . . . .	105
Figura 42 – Resultados de simulação de equalização. Fonte: [A]. . . . .	108
Figura 43 – Artigos publicados pelo autor relacionados com os blocos de construção do modelo SDN . . . . .	127

# Lista de tabelas

Tabela 1 – Parâmetros do experimento com transponder flexível . . . . .	89
Tabela 2 – Análise de desempenho para o cálculo de caminhos . . . . .	96
Tabela 3 – Comparação de custos relativos entre a arquitetura SD-POSS, com soluções baseadas em IP e/ou OTN . . . . .	98



# Lista de Acrônimos e Abreviações

3GPP	3rd Generation Partnership Project
AAA	Authentication, Authorization, and Accounting
AC	Autonomic Computing
ACS	Ambient Control Space
AFI	ETSI ISG - Autonomic network engineering for the self-managing Future Internet
AGC	Automatic Gain Control
AGCN	Agente de Configuração NETCONF
AIS	Alarm Indication Signal
AML	Autonomic Management Layer
AN	Ambient Networks
ANI	Ambient Network Interface
APA	Autonomic Policy Agents
API	Application Programming Interface
APS	Autonomic Policy Server
AS	Autonomous Systems
ASE	Amplified Spontaneous Emission
ASI	Ambient Services Interface
ASN	Autonomous System Number
ASON	Automatic Switched Optical Network
AWG	Arrayed Waveguide Grating
BER	Bit Error Rate
BGP	Border Gateway Protocol

BVT Bandwidth-Variable Transponders

CAPEX Capital Expenditure

CD Colorless e Directionless

CDC Colorless, Directionless e Contentionless

CDC-F Colorless, Directionless, Contentionless e Flex-grid

CDL Cognition Definition Language

CDN Content Distribution Network

CHRON Cognitive Heterogeneous Reconfigurable Optical Network

CIM Common Information Model

COPS Common Open Policy Service

CORBA Common Object Request Broker Architecture

COTS Commercial Off-The-Shelf

CP Connection point

CPE Customer Premises Equipment

CPqD Centro de Pesquisa e Desenvolvimento em Telecomunicações

CPU Central Processing Unit

CW Continuous Wave

DA Discovery Agent

DCOM Distributed Component Object Model

DDOS Distributed Denial of Service

DEMUX Demultiplexador

DEN-ng Directory Enabled Network-new generation

DFZ Default-free zone

DHCP Dynamic Host Configuration Protocol

DLUX OpenDayLight User Experience

DMTF Distributed Management Task Force

DOCSIS Data Over Cable Service Interface Specification

DP-QPSK Dual Polarization Quadrature Phase Shift Keying

DPI Deep Packet Inspection

DSP Digital Signal Processor

DTD Document Type Definition

DWDM Dense Wavelength Division Multiplexing

E-NNI External Network-to-Network Interface

ECA Event-Condition-Action

ECMP Equal Cost Multipath

EDFA Erbium Doped Fibre Amplifier

ELC Explicit Label Control

EMS Element Management System

EON Elastic Optical Networks

EQUAL-APP Aplicação SDN para equalização óptica

ETSI European Telecommunication Standards Institute

FA Forwarding Adjacency

FA-LSP Forwarding Adjacency LSP

FC Fibre Channel

FCAPS Fault, Configuration, Accounting, Performance and Security

FEC Forward Error Correction

FIB Forwarding Information Base

FIC Fabric Interface

FPGA Field-Programmable Gate Array

FRM Forwarding Rules Manager

FSC Fiber-Switch Capable

GANA Generic Autonomic Network Architecture

GC Global Constraints

GCC General Communications Channel

GCO Global Concurrent Optimization

GCV Global Concurrent Vector

GLL Generic Link Layer

GMPLS Generalized Multi-Protocol Label Switching

GOF Global Objective Function

GPB Group Based Policy

HA High Availability

HTTP Hypertext Transfer Protocol

IA Inteligência Artificial

IAB Internet Architecture Board

IBM International Business Machines Corporation

IDEALIST Industry-Driven Elastic and Adaptive Lambda Infrastructure for Service and Transport Networks

IETF Internet Engineering Task Force

IGP Interior Gateway Protocol

ILP Integer Linear Programming

IM-NENS Information Model for NE, Network and Services

IntServ Integrated Services

IP Internet Protocol

IPS Intrusion Prevention System

IS-IS Intermediate System to Intermediate System

ISC Interface Switching Capability

ISCD Interface Switching Capability Descriptor

ISG Industry Specification Group

ISO International Organization for Standardization

ITU-T International Telecommunication Union - Telecommunication Standardization Sector

JSON JavaScript Object Notation

JUNG Java Universal Network/Graph Framework

JVM Java Virtual Machine

LAN Local Area Network

LCoS Liquid Crystal on Silicon

LDAP Lightweight Directory Access Protocol

LISP Locator/Identifier Separation Protocol

LLDP Link Layer Discovery Protocol

LMP Link Management Protocol

LPDL Logic-based Policy Definition Language

LPDP Local Policy Decision Point

LRM Link Resource Manager

LSA Link State Advertisement

LSC Lambda Switch Capable

LSP Label Switched Path

LXC Linux Containers

MBC Minimize aggregate Bandwidth Consumption

MBP Maximum residual Bandwidth Path

MCC Minimize the Cumulative Cost of a set of paths

MCP Minimum Cost Path

MCS Multicast Switch

MEMS Micro-Electro-Mechanical Systems

MIT Management Information Tree

MLL Minimize the Load of the most loaded Link

MLN MultiLayer Network

MLP Minimum Load Path

MME Mobility Management Entity

MO Managed Object

MPLS Multiprotocol Label Switching

MPLS-TE MPLS Traffic Engineering

MRN MultiRegion Network

MTOSI Multi-Technology Operations System Interface

MUX Multiplexador

NaaS Network as a Service

NAT Network Address Translation

NCS Network Control System

NE Network Element

NETCONF Network Configuration Protocol

NF Network Functions

NFV Network Functions Virtualization

NFVI NFV Infrastructure

NFVI-PoP NFV Infrastructure Point of Presence

NG-OTN Next Generation Optical Transport Networks

NIST National Institute of Standards and Technology

NMS Network Management System

NOS Network Operating System

NP Network Processor

NS2 Network Simulator 2

NVGRE Network Virtualization using Generic Routing Encapsulation

NVP Nicira Network Virtualization Platform

O-NE Optical Network Element

O-OFDM Optical Orthogonal Frequency Division Multiplexing

OAM Operations and Management

OBS Optical Burst Switching

OCL Object Constraint Language

OCM Optical Channel Monitor

ODU Optical Channel Data Unit

OEO Optical-Electrical-Optical

OIF Optical Internetworking Forum

ONF Open Networking Foundation

OPEX Operational Expenditure

OPFP OTN over Packet Fabric Protocol

OPS Optical Packet Switching

OSGi Open Service Gateway initiative

OSI Open Systems Interconnection

OSNR Optical Signal Noise Ratio

OSPF Open Shortest Path First

OSPF-TE Open Shortest Path First – Traffic Engineering

OSS Operations Support System

OTN Optical Transport Network

OTP Optical Transport Platforms

OTS Open Transport Switch

OTT Over The Top

OTU Optical channel Transport Unit

OVSDB Open vSwitch DataBase Protocol

OXC Optical Cross-Connect

P-OTP Packet Optical Transport Platforms

P-OTS Packet-Optical Transport System

PAN Personal Area Networks

PB-TSDN Policy-based Transport SDN

PBNM Policy-Based Network Management

PCC Path Computation Client

PCE Path Computation Element

PCEP PCE Communication Protocol

PCIM Policy Core Information Model

PCMM Packet Cable MultiMedia

PDP Policy Decision Point

PE Policy Element

PEG Policy Engine

PEP Policy Enforcement Point

PFDL Policy Framework Definition Language

PGW Packet Data Network Gateway

PID Proportional, Integral, Derivative

PLC Planar Lightwave Circuit

PLI Physical Layer Impairments

Plugin2OC Plugging To OpenContrail

PR Policy Repository

PSC Packet-Switch Capable

PTT Ponto de Troca de Tráfego

QoS Quality of Service

R+DWA Routing + Distributed Wavelength Assignment

REST Representational State Transfer

RESTCONF Protocolo equivalente ao NETCONF, baseado em REST

RIB Routing Information Base

RM-ODP Reference Model of Open Distributed Processing

RMI Remote Method Invocation

RNP Rede Nacional de Ensino de Pesquisa

ROADM Reconfigurable Optical Add-Drop Multiplexer

ROADM-CD ROADM Colorless e Directionless

RPC Remote Procedure Call

RSA Routing and Spectrum Assignment

RSVP-TE Resource Reservation Protocol – Traffic Engineering

RWA Routing and Wavelength Assignment

S-IONP SDN - In Operation Network Planning

SaaS Software as a Service

SAL Service Abstraction Layer

SD Signal Degrade

SD-POSS Software-defined Packet-optical Switching System

SDH Synchronous Digital Hierarchy

SDN Software Defined Networking

SDN-VE SDN for Virtual Environments

SDNI SDN Interface (Cross-Controller Federation)

SDNRG Software Defined Networking Research Group

SF Signal Failure

SFC Service Function Chaining

SFP Small Form-factor Pluggable

SLA Service Level Agreement

SNBI Secure Network Bootstrapping Infrastructure

SNMP Simple Network Management Protocol

SNP Subnetwork Point

SNPP Subnetwork Point Pool

SO Sistema Operacional

Sonet Synchronous Optical Network

SPARC Split Architecture Carrier Grade Networks

SRLG Shared Risk Link Group

SSH Secure Shell

STP Spanning Tree Protocol

SUPA Shared Unified Policy Automation

T-NOS Transport Network Operating System

T-NVF Transport - Network Functions Virtualization

T-SDN Transport SDN

T-SDNC Transport SDN Controller

TAP Termination and Adaptation Performer

TCP Transmission Control Protocol

TDM Time-Division Multiplexing

TED Traffic Engineering Database

TLV Type Length Value

TTI Trail Trace Identifier

TTP Table Type Patterns

UCP Unified Control Plane

UNI User to Network Interface

VLAN LAN virtual

VLAN Virtual Local Area Network

VM Virtual Machine

VNF Virtualised Network Function

VNF-FG VNF Forwarding Graph

VNI VXLAN Network Identifier

VNT Virtual Network Topology

VON Virtual Optical Networks

VONCR-APP Virtual Optical Network Configuration and Restoration SDN Application

VPN Virtual Private Network

VTEP VXLAN Tunnel Endpoint

VTN Virtual Tenant Network

VXLAN Virtual eXtensible Local Area Network

WAN Wide Area Network

WBEM Web-Based Enterprise Management

WSON Wavelength Switched Optical Networks

WSS Wavelength Selective Switch

XCP eXtended Control Plane

XCP-Mgr eXtended Control Plane Manager

XML Extensible Markup Language

XSLT XML Stylesheet Language for Transformations

YANG Data Modeling Language for the Network Configuration Protocol

YMCA Yang Models for Configuration and topology Abstraction

# 1 Introdução

A crescente demanda por velocidades de transmissão mais altas, devido ao surgimento de novas aplicações como computação em nuvem, transmissão de vídeo de alta definição e serviços *Over The Top* (OTT), tem gerado grandes desafios às operadoras de telecomunicações (*carriers*) relativos a como projetar, implementar, operar e integrar suas redes. Um dos principais desafios é a manutenção ou redução de custos *Capital Expenditure* (CAPEX) e *Operational Expenditure* (OPEX); ao mesmo tempo que as redes tornam-se maiores e mais complexas, o valor recebido por bit transportado é cada vez menor e os clientes demandam novos tipos de serviços que devem ser criados e provisionados rapidamente.

A fertilização cruzada entre tecnologias de transporte baseadas em *Time-Division Multiplexing* (TDM) e *Dense Wavelength Division Multiplexing* (DWDM) com as tecnologias de redes de pacotes está permitindo a evolução de uma arquitetura ossificada, caracterizada por camadas separadas, para um cenário mais integrado e flexível, de forma a minimizar duplicação de funções, permitindo otimização de recursos, implementação mais flexível de novos serviços e melhor coordenação na realização das funções da rede. Desta forma, a composição das redes ópticas de transporte está mudando rapidamente de sistemas ponto a ponto para topologias em malha, com diversidade de formatos de modulação, diferentes granularidades de comutação, suporte a várias estratégias de sobrevivência a falhas, entre outros. Tal evolução é viabilizada também pelo surgimento de novas tecnologias de comutação em redes ópticas como *Reconfigurable Optical Add-Drop Multiplexers* (ROADMs) (PERRIN, 2010), equipamentos com suporte a comutação multicamadas, incluindo *Optical Channel Data Unit* (ODU) e pacotes (GLINGENER, 2011), os quais têm incorporado suporte a um maior grau de flexibilidade e dinamismo nas redes de transporte, bem como amplo crescimento de capacidade a menores custos, permitido o crescimento contínuo na capacidade das conexões no acesso, redes metropolitanas e de longa distância a elevadas taxas (CISCO, 2012). No âmbito desta tese, o termo *Next Generation Optical Transport Networks* (NG-OTN) será utilizado para redes ópticas de transporte que utilizam tais tecnologias.

Atualmente, os principais fornecedores de equipamentos de telecomunicações estão instalando equipamentos de transmissão com interfaces de 100 Gbps com tecnologia coerente, ROADMs com suporte a *Colorless, Directionless e Contentionless* (CDC) e grade flexível (INFINERA, 2013; ALCATEL-LUCENT, 2013), bem como comutadores OTN. Esta revolução nas redes de transporte tem sido chamada de *reboot* óptico (SCHMITT, 2011). Adicionalmente, fornecedores de roteadores *Internet Protocol* (IP) de grande porte estão promovendo

o projeto de redes com interfaces *Optical Transport Network* (OTN) e DWDM integradas nos roteadores, bem como suporte a plano de controle integrado (IP-Óptico) (CISCO, 2013; ADVA, 2013), permitindo otimização das funções de rede, como o suporte a estratégias de projeto com arquiteturas com núcleo puramente óptico.

Atualmente há um processo contínuo de implementação de novos equipamentos nas redes, os quais utilizam em sua maioria hardware e software proprietários, não oferecendo flexibilidade e escalabilidade necessárias para acompanhar a evolução tecnológica introduzida para atender aos novos serviços. Desta forma, a academia, as operadoras e os fabricantes de equipamentos estão em busca de novas abordagens para a concepção de redes, as quais são inspiradas em parte no modelo de desenvolvimento dos computadores pessoais e dos *data centers* modernos que implementam o conceito de computação em nuvem, utilizando princípios de virtualização, abertura de interfaces e protocolos.

O conceito de *Software Defined Networking* (SDN) (KREUTZ *et al.*, 2014) (em português – Redes Definidas por Software) vem sendo proposto como uma abordagem arquitetural para a separação dos planos de controle e de encaminhamento, habilitando as redes a suportarem novas funcionalidades como virtualização, automação nas tarefas de operação de forma programável, visão global através da centralização lógica e unificação dos planos de controle e facilidade na implementação de novos tipos de serviços. Desta forma, os princípios de SDN vêm sendo estendidos a redes ópticas de transporte (GRINGERI *et al.*, 2013) como uma forma de habilitar a integração transparente de redes multicamadas, respondendo a desafios não atendidos por completo atualmente.

Outra iniciativa, liderada pelas principais *carriers* do mundo, objetiva construir a definição de um novo conceito: a *Network Functions Virtualization* (NFV) através de um grupo de trabalho formado no *European Telecommunication Standards Institute* (ETSI) (CHIOSI, 2012). O conceito de NFV é complementar à virtualização de redes, permitindo que diversos elementos de rede como firewalls, roteadores, switches e *Deep Packet Inspections* (DPIs) possam operar como máquinas virtuais em uma plataforma de hardware padronizada.

Adicionalmente, na última década muito esforço foi realizado em pesquisa e desenvolvimento na área de *Policy-Based Network Management* (PBNM) (STRASSNER, 2003), com o objetivo principal de permitir que políticas de alto nível sejam mapeadas de forma automática para configurações apropriadas nos dispositivos de rede, reduzindo a complexidade nas operação das redes independentemente do tipo, fornecedor ou sistema operacional utilizado, bem como para permitir a operação autônoma da rede, com base nas políticas preestabelecidas. A incorporação do conceito de política é fundamental para o estabelecimento de SDN e esforços recentes têm proposto linguagens, abstrações e protocolos de políticas adequados

ao ecossistema SDN (FOSTER *et al.*, 2013; SMITH *et al.*, 2014).

As próximas seções apresentam, de forma sucinta, a definição do problema e as contribuições desta tese que resultaram da busca por possíveis soluções.

## 1.1 Desafios e definição do problema

O desenvolvimento tecnológico dos equipamentos que compõem as redes ópticas de transporte NG-OTN, incluindo tecnologias de comutação e transmissão como ROADM, comutadores mistos de pacotes e OTN, receptores coerentes, entre outros, bem como a implementação de plano de controle *Generalized Multi-Protocol Label Switching* (GMPLS), principalmente para a implementação de funções de sobrevivência, tem desempenhado um papel fundamental para o atendimento da crescente demanda de tráfego.

No entanto, há diversas lacunas para que as redes possam atender às demandas futuras de forma sustentável, atendendo aos requisitos de flexibilidade, custo, facilidade de operação e integração multi-fornecedores. Neste contexto, são apresentados a seguir alguns dos desafios, para os quais soluções são investigadas no escopo desta tese.

» **Técnicas de planejamento, controle automático e gerência das redes ópticas de transporte, incluindo cenários multi-camada, não evoluíram no mesmo ritmo das tecnologias do plano de dados**, de forma que tais redes, mesmo que sejam cada vez mais complexas, continuam a ser planejadas e operadas da mesma forma que as redes *Synchronous Digital Hierarchy* (SDH) implementadas há mais de vinte anos. Adicionalmente, com relação ao plano de controle, há deficiências relacionadas a não efetividade e imprevisibilidade em cenários de sobrevivência a falhas e interoperabilidade. Desta forma, com o grande crescimento das redes e na complexidade tecnológica, os operadores enfrentam cada vez mais dificuldades no planejamento, otimização e até mesmo na operação das redes.

» **Atualmente, as redes ópticas de transporte não atendem plenamente aos requisitos impostos para interconexão de *data centers***. Grandes empresas como Google, Facebook e Amazon têm buscado desenvolver soluções tecnológicas abertas, porém personalizadas para suas necessidades, com alto nível de integração entre diferentes camadas, realizando orquestração de recursos virtualizados conforme as demandas. Em um cenário onde a demanda por recursos de rede é muito alta (eventualmente da ordem de Tbps), podendo mudar muito rapidamente, as redes NG-OTN deverão incorporar características já presentes nas camadas superiores, como suporte a virtualização, com interfaces para a gerência dos recursos virtualizados.

## 1.2 Contribuições

Esta tese aborda os problemas apresentados na seção anterior de forma parcial, trazendo para o cenário das redes NG-OTN inovações que têm sido realizadas no âmbito de redes IP e de *data centers*, incluindo SDN, virtualização de redes, redes autonômicas e PBNM. Através destes mecanismos, as redes ópticas de transporte poderão incorporar funcionalidades e características como maior flexibilidade para implementação de serviços, controle autônomo e melhor integração com as camadas superiores.

A Figura 1 apresenta uma visão de alto nível da arquitetura do controlador SDN para redes de transporte proposta, a qual consiste na principal contribuição desta tese.

As contribuições desta tese podem ser divididas em três grupos:

- **Arquitetura para a implementação de redes NG-OTN definidas por software, ou T-SDN**, permitindo a implementação de funções de automação e virtualização destas redes através do controlador SDN. Uma camada de abstração que atua como um sistema operacional de redes de transporte, ou *Transport Network Operating System* (T-NOS) suporta a execução de aplicações SDN que podem ter acesso, através de

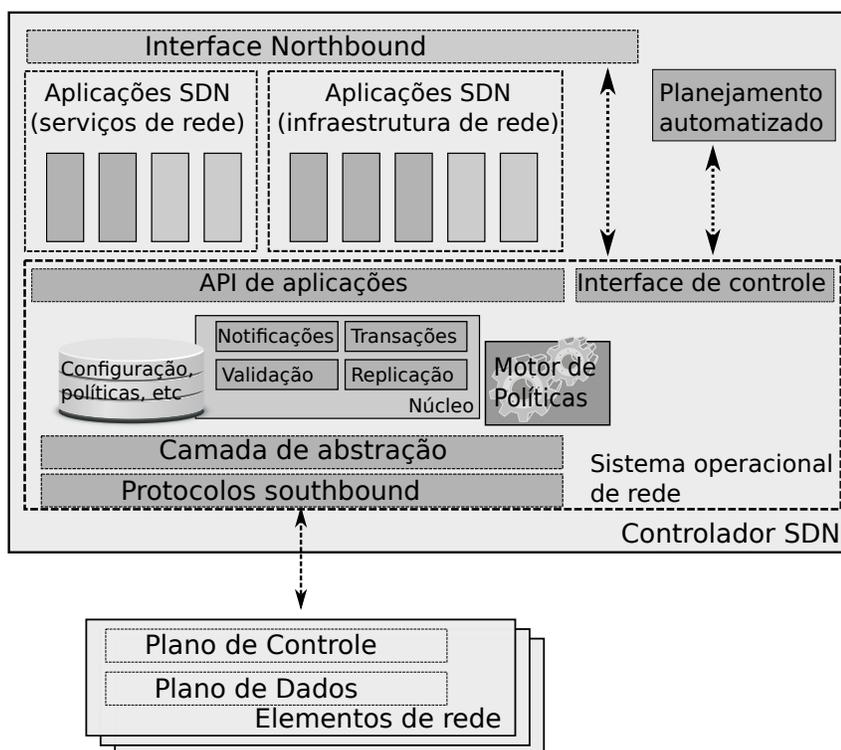


Figura 1 – Visão de alto nível da arquitetura proposta para o controlador SDN para redes de transporte

*Application Programming Interface* (API), a informações topológicas, de configuração e desempenho, de forma que possam ter visão global da rede e local dos elementos de rede, permitindo realizar variadas funções.

- **Modelo de políticas aplicável ao ecossistema SDN**, no qual é introduzido o conceito de motor de políticas para aplicações SDN, utilizando o conceito de *policy flow*. Os principais objetivos da introdução do conceito de políticas em SDN são que o motor de políticas seja capaz de governar: *i*) o comportamento das aplicações SDN, *ii*) o controle de acesso aos recursos da rede e *iii*) o modelo de concorrência e conflitos. Na proposta realizada, a avaliação de políticas utiliza uma abordagem baseada em análise de grafos, na qual os grafos são gerados e atualizados automaticamente com base em modelos *Data Modeling Language for the Network Configuration Protocol* (YANG) (BJORKLUND, 2010), bem como pelas próprias definições das políticas.
- **Implementação e validação experimental de aplicações SDN** incluindo os seguintes casos de uso:
  - UC01 – Virtualização da rede com suporte a GMPLS e instanciação automática de VONs.
  - UC02 – Reconfiguração dinâmica de VON com base em políticas.
  - UC03 – Aplicação SDN para reconfiguração dinâmica do formato de modulação.
  - UC04 – Plano de controle SDN com suporte a NEs flexíveis.
  - UC05 – Ajuste dinâmico de políticas de sobrevivência através de planejamento em operação.
  - UC06 – Ajuste fino de políticas locais via planejamento automatizado

## 1.3 Trabalhos relacionados

Esta seção apresenta uma amostra dos trabalhos mais relevantes relacionados aos temas explorados no contexto desta tese, os quais são organizados em quatro áreas: (1) SDN em redes ópticas de transporte, (2) virtualização de redes ópticas de transporte, (3) aplicação de políticas em SDN, e (4) planejamento de redes multicamadas.

### 1.3.1 SDN em redes ópticas de transporte

O conceito de SDN vem sendo proposto como uma abordagem arquitetural para a separação dos planos de controle e de encaminhamento, habilitando as redes a suportarem

novas funcionalidades como virtualização, automação nas tarefas de operação de forma programável, visão global através da centralização lógica e unificação dos planos de controle e facilidade na implementação de novos tipos de serviços. Desta forma, os princípios de SDN vêm sendo estendidos para redes ópticas de transporte (GRINGERI *et al.*, 2013) como uma forma de habilitar a integração transparente de redes multicamadas, respondendo a desafios não atendidos por completo atualmente. Os autores apresentam benefícios e desafios para estender os conceitos de SDN a redes de transporte, destacando as vantagens da abordagem SDN para redes multicamadas; no entanto, comparando-se com o trabalho realizado no contexto desta tese, somente é realizada uma abordagem conceitual, sem análise ou proposta de soluções específicas. Em um trabalho similar, Liu *et al.* (2012) demonstram a configuração de redes ópticas reconfiguráveis através do protocolo OpenFlow, incluindo suporte a provisionamento dinâmico de caminhos fim a fim. No entanto, o trabalho não resolve questões relacionadas à arquitetura SDN, modelagem e abstração da rede, políticas e virtualização de redes.

Uma das propostas do paradigma SDN é que o plano de controle dos equipamentos de rede seja simplificado, de forma que funções mais complexas sejam realizadas pelo controlador. Neste contexto, Das (2012) propõe uma arquitetura baseada em SDN para a implementação de um *Unified Control Plane* (UCP), usando o conceito de sistema operacional de redes, ou *Network Operating System* (NOS), controlando comutadores de pacotes e de circuitos através de uma API única (e.g., OpenFlow), de forma a abstrair interfaces específicas de fabricantes ou camadas específicas, provendo configuração flexível do plano de encaminhamento por um plano de controle único, baseado em SDN. No entanto, o autor não aborda questões como políticas, aspectos mais associados às redes ópticas de transporte, como restrições de camada física, mecanismos de proteção e restauração e compatibilidade com mecanismos legados.

O conceito de *Open Transport Switch* (OTS) (LIOU, 2013) foi introduzido através do desenvolvimento de uma interface OpenFlow para uma comutador OTN comercial, que combina funcionalidades de comutação de pacotes *Multiprotocol Label Switching* (MPLS), tributários OTN e comprimentos de onda na mesma plataforma. Os autores argumentam que o paradigma das redes de transporte atuais já é baseado em um controle centralizado, auxiliado por funções distribuídas implementadas pela pilha GMPLS, incluindo descoberta de topologia, atualizações de estado de enlaces e restauração dinâmica. Desta forma, os autores afirmam que o desenvolvimento de arquitetura SDN para redes de transporte deverá considerar, além da configuração de fluxos, estratégias de migração, bem como a melhor localização de funções específicas de rede, como descoberta de topologia, cálculo de caminhos, proteção e restauração de serviços. Desta forma, são propostas diferentes abordagens de

abstração para o suporte das múltiplas camadas pelo controlador SDN, incluindo cenários que mantêm o plano de controle atual, disponibilizando interfaces OpenFlow somente nas bordas da rede, bem como possibilidade de controle mais granular, com interfaces OpenFlow em todos os elementos de rede. O trabalho possui uma abordagem mais prática, contribuindo com requisitos para a implementação de SDN em redes de transporte, porém não propõe uma arquitetura, nem aborda aspectos importantes como políticas.

### 1.3.2 Virtualização de redes ópticas de transporte

O fabricante Ciena apresenta um *whitepaper* (CIENA, 2013) com motivadores para o provimento de serviços *Network as a Service* (NaaS) em redes ópticas. Já no projeto *Split Architecture Carrier Grade Networks* (SPARC) (JOHN, 2012) é realizado um levantamento de requisitos e soluções para virtualização de redes, com foco em redes de operadoras, incluindo a proposta de arquiteturas com controladores SDN hierárquicos, permitindo o particionamento de redes (*slicing*). No entanto, somente é realizada uma abordagem conceitual, sem análise, proposta ou validação de soluções específicas.

Os autores Perello *et al.* (2012) propõem um mecanismo para alocação de VONs de forma ótima sobre uma rede puramente óptica através de planejamento *offline*. Já Peng *et al.* (2013) propõem modelos para alocação de VONs, considerando *Physical Layer Impairments* (PLI) para cenários intra e inter-VON. Ambos os trabalhos possuem foco específico na realização de formulações *Integer Linear Programming* (ILP) para a alocação ótima de redes virtuais em redes ópticas.

Já no contexto de redes SDN híbridas (SDN-IP), *RouteFlow* (ROTHENBERG *et al.*, 2012) utiliza uma arquitetura de roteamento, combinando o desempenho de plataformas comerciais com a flexibilidade de plataformas de código aberto, rodando em hardware de propósito geral, permitindo a implementação de NaaS em ambientes de redes de pacotes. O trabalho é focado em roteamento em redes IP, não considerando aspectos como redes ópticas e políticas.

Os autores Esteves *et al.* (2013) apresentam os desafios para o gerenciamento de redes virtualizadas, bem como soluções prominentes e questões em aberto. No entanto, somente é realizada uma abordagem conceitual, sem análise, proposta ou validação de soluções específicas.

### 1.3.3 Políticas em SDN

Trabalhos recentes propõem linguagens de programação para SDN (FOSTER *et al.*, 2013), incluindo *Pyretic* (MONSANTO *et al.*, 2013), uma linguagem e ambiente de execu-

ção para de aplicações SDN como foco em políticas de encaminhamento de pacotes. *Pyretic* introduz um modelo abstrato de pacotes, uma álgebra de políticas de alto nível, bem como “objetos de rede”, permitindo a implementação de aplicações SDN com suporte a políticas e virtualização. Estes trabalhos propõem operações de composição de políticas de encaminhamento para a criação de regras para a instalação nos dispositivos de rede. No entanto, comparando à abordagem proposta nesta tese, os autores utilizam o conceito de política de forma muito específica, considerando política como sinônimo de “regra de encaminhamento de pacotes”.

Uma proposta recente de grupo de trabalho no *Internet Engineering Task Force* (IETF) denominado *Shared Unified Policy Automation* (SUPA) (KARAGIANNIS *et al.*, 2014), possui objetivos muito alinhados com o trabalho desenvolvido no escopo desta tese. O principal objetivo do grupo é prover uma forma para que aplicações e serviços de rede possam especificar políticas para a rede a partir de uma visão simplificada desta. A proposta é que isto seja possível através da modelagem da infraestrutura de rede como um grafo utilizando YANG, expondo esta visão para as aplicações através de NETCONF ou *Representational State Transfer* (REST), bem como soluções para o mapeamento automático de políticas em atributos do grafo da rede.

### 1.3.4 Planejamento de redes ópticas multicamadas

Arquiteturas de redes multicamadas, com foco na integração IP/óptica, vêm evoluindo há quase duas décadas. As contribuições incluem aspectos como coordenação de mecanismos de sobrevivência a falhas, otimização de recursos e facilidade de operação (WU, 1995; AYANOGLU, 1995; YE *et al.*, 2000). Adicionalmente Kerivin e Mahjoub (2005) apresentam um levantamento de abordagens para o projeto de redes resilientes. O artigo mostra que, mesmo que somente uma camada de rede seja considerada, o projeto de uma rede com suporte a sobrevivência a múltiplas falhas já é um problema NP-difícil. No escopo desta tese, é utilizada uma abordagem diferente da utilizada nos trabalhos em questão, na qual as diferentes camadas são integradas em um único grafo, o qual inclui em sua abrangência as conexões internas dos elementos de redes multi-camadas, permitindo a unificação do controle multicamadas de forma mais simplificada através de aplicações SDN.

Ferramentas de planejamento de redes de transporte (TANG *et al.*, 2011; TANG *et al.*, 2009) permitem a realização de dimensionamento de redes multicamadas de forma *offline*, além de análises do tipo *what-if*. No entanto, comparando-se ao trabalho realizado no contexto desta tese, os trabalhos referenciados não utilizam o conceito de “planejamento em operação”, o qual permite alimentar a “camada de inteligência” do motor de políticas com resultados de simulações *online*.

## 1.4 Organização da Tese

Esta tese está organizada da seguinte forma: o Capítulo 2 apresenta fundamentos tecnológicos e o estado da arte de redes ópticas de transporte. O Capítulo 3 apresenta uma revisão e estado de desenvolvimento e tendências de mecanismos e paradigmas de gerência e controle das redes, incluindo mecanismos para controle autônomo de redes, redes definidas por software e virtualização. O Capítulo 4 apresenta a proposta de uma arquitetura SDN para redes NG-OTN com suporte a políticas e virtualização. O Capítulo 5 apresenta casos de uso e aplicações. Finalmente, o Capítulo 6 apresenta uma discussão final e conclusões. Adicionalmente, a tese inclui o Apêndice A que apresenta a lista de publicações do autor que formam os pilares das contribuições apresentadas, as quais são referenciadas no texto no formato [A-R]. Já o Apêndice B apresenta *logs* da ferramenta desenvolvida para a realização de planejamento automatizado de ações do controlador SDN proposto.



## 2 Redes Ópticas de Transporte de Nova Geração (NG-OTN)

Este capítulo apresenta o estado da arte de NG-OTN, incluindo: redes ópticas elásticas, tecnologias de comutadores ópticos, digitais e de pacotes, bem como plano de controle e redes multicamadas.

### 2.1 Evolução das redes ópticas de transporte

A limitação imposta atualmente pela grade DWDM com canais de 50GHz de largura de espectro impõe limitações ao desenvolvimento de transmissores com taxas acima de 100 Gbps. Desta forma, a *International Telecommunication Union - Telecommunication Standardization Sector* (ITU-T) padronizou uma grade de frequências flexível (ITU-T, 2012), denominada *flex-grid*, que permite o estabelecimento de conexões com diferentes larguras de espectro. Esta flexibilização permitiu a criação do conceito de redes ópticas elásticas, o qual, além de permitir o estabelecimento de conexões com maior largura de banda, potencializa a melhor utilização do espectro. Um recente relatório do projeto *Industry-Driven Elastic and Adaptive Lambda Infrastructure for Service and Transport Networks* (IDEALIST) (BOHN *et al.*, 2013) lista as tecnologias *flex-grid* e *Bandwidth-Variable Transponders* (BVT) (GERS-TEL *et al.*, 2012) como chaves para o desenvolvimento de redes ópticas de nova geração. Por sua vez, os transponders BVT permitem a utilização mais eficiente do espectro, permitindo a criação de fluxos dinâmicos, que ocupam o espectro conforme a demanda, permitindo melhor eficiência energética, além de permitir novas possibilidades na integração de IP e óptica. A implementação de redes ópticas elásticas apresenta muitos desafios aos elementos de rede bem como ao plano de controle, já que a rede óptica passa a ser mais dinâmica, apresentando novos desafios como fragmentação do espectro e maior dinamicidade.

#### 2.1.1 Equipamentos de comutação em redes ópticas

##### 2.1.1.1 Comutação óptica sem conversão OEO

Equipamentos de comutação óptica com suporte a roteamento de comprimentos de onda são denominados ROADMs. Em sua primeira geração estes equipamentos foram desenvolvidos utilizando-se tecnologia *Planar Lightwave Circuit* (PLC) ou *Wavelength Selective Switch* (WSS) (MAGALHAES *et al.*, 2013). Por sua vez, os comutadores WSS podem ser

construídos utilizando-se tecnologia *Micro-Electro-Mechanical Systems* (MEMS) ou *Liquid Crystal on Silicon* (LCoS). Tipicamente, ROADMs construídos com tecnologia PLC limitam-se a duas direções, podendo ser aplicados em redes com topologia em anel. Já os ROADMs desenvolvidos com tecnologia WSS podem ser construídos utilizando-se comutadores WSS com suporte a  $1 \times N$  portas, tendo suporte a  $N$  direções, permitindo a construção de redes ópticas em malha. Tipicamente, para os ROADMs de primeira geração, o valor máximo de  $N$  é 8.

Tantos os módulos WSS implementados com tecnologia MEMS como os implementados com LCoS possuem as seguintes características principais:

- Funcionam de forma bidirecional.
- Na direção  $1 \Rightarrow N$ , permitem a seleção de quais comprimentos de onda entrantes serão encaminhados por cada uma das  $N$  portas.
- Na direção  $N \Rightarrow 1$ , permitem a seleção de quais comprimentos de onda entrantes por cada uma das  $N$  portas serão encaminhados pela porta de saída.
- Apresentam alto tempo de comutação e/ou reconfiguração, da ordem de centenas de milissegundos.
- O valor máximo de  $N$  é limitado entre 20 e 23, dependendo do fabricante, sendo que há *roadmaps* para WSSs de  $1 \times 40$ .

A Figura 2 ilustra a arquitetura típica de um ROADM de primeira geração, conforme classificado por (MAGALHAES *et al.*, 2013). Tal ROADM pode ser construído utilizando-se arquitetura *Route-and-Select* ou *Broadcast-and-Select*. A primeira utiliza módulos WSS tanto na entrada como na saída do ROADM, já a segunda utiliza *splitters* na entrada e WSS na saída. A maior vantagem da arquitetura *Route-and-Select* é a menor perda de inserção, seguida pela diminuição no *crosstalk*, porém apresenta custos mais elevados, devido à utilização do dobro de módulos WSS. Outra desvantagem da arquitetura *Route-and-Select* é o estreitamento do espectro devido ao aumento do efeito de filtragem em cascata.

Conforme mostrado na Figura 2, os ROADMs de primeira geração possuem portas de ADD/DROP distintas para cada direção do ROADM, sendo esta característica definida como “ROADM direcionado”. Outra característica que pode ser observada é que nas portas de ADD/DROP são utilizados Multiplexador / Demultiplexador do tipo *Arrayed Waveguide Grating* (AWG), os quais possuem portas cliente para conexão de transponders, sendo que cada porta é associada a um comprimento de onda da grade DWDM do ITU-T. Desta

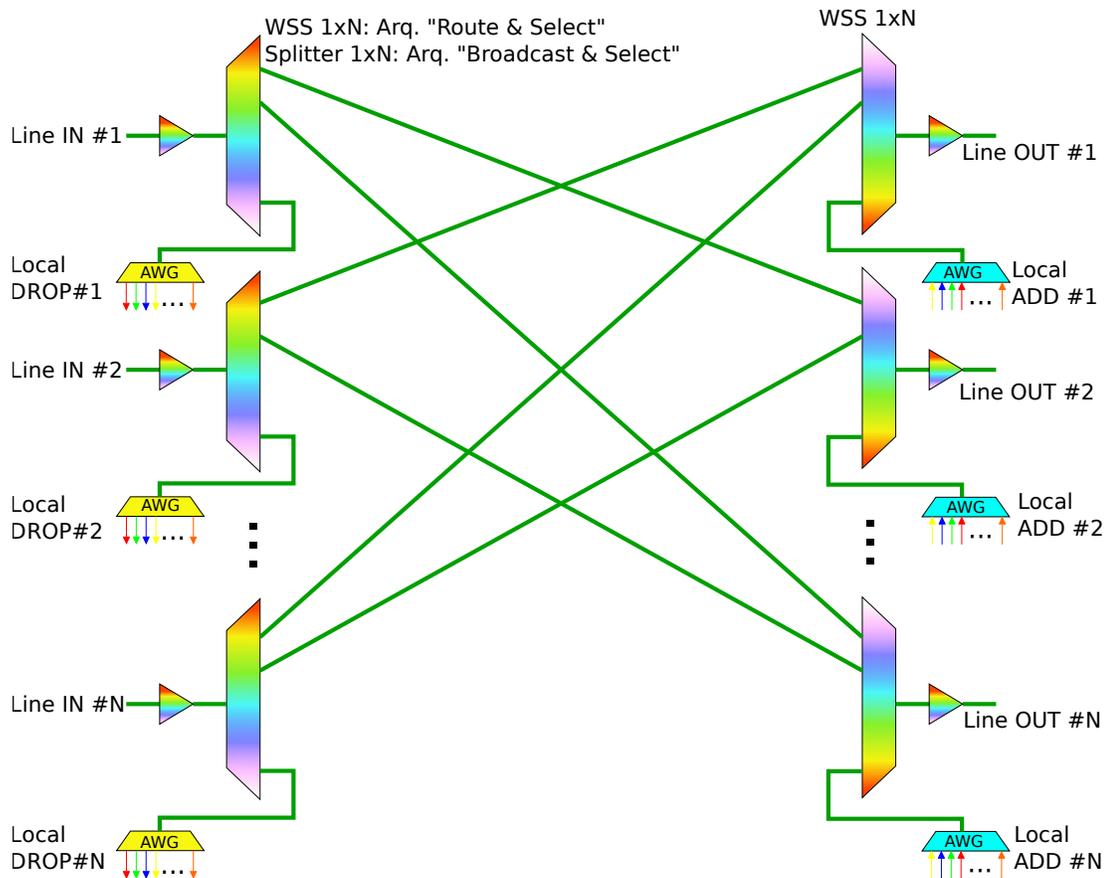


Figura 2 – ROADM WSS de primeira geração

forma, os ROADMs de primeira geração são denominados coloridos, ou seja, cada porta de ADD/DROP tem um comprimento de onda (cor) pré-definido.

As restrições de direção e cor são indesejáveis em redes reconfiguráveis, já que dificultam o re-roteamento automático e a alteração de comprimento de onda sem a intervenção humana no sítio (e.g., alteração de porta do cordão óptico do transponder). Desta forma, foi proposta uma solução denominada *ROADM Colorless e Directionless* (CD), na qual o ROADM é equipado com um conjunto de portas ADD/DROP globais, que podem ser encaminhadas por qualquer das direções, bem como podem utilizar qualquer comprimento de onda. A Figura 3 apresenta o modelo conceitual de um ROADM CD, o qual utiliza módulos WSS do tipo  $M \times N$  nos bancos de ADD/DROP. No entanto, tais módulos, ainda no *roadmap* de alguns fabricantes, apresentam contenção interna, o que significa que em cada WSS  $N \times M$  o mesmo comprimento de onda não pode ser roteado duas vezes, mesmo que sejam roteados de/para portas diferentes. Módulos WSS sem contenção não estão disponíveis comercialmente e não há informação de *roadmap* para tais módulos. ROADMs CD sem contenção são denominados ROADM CDC. No contexto desta tese, os ROADMs do tipo CDC são considerados

de segunda geração.

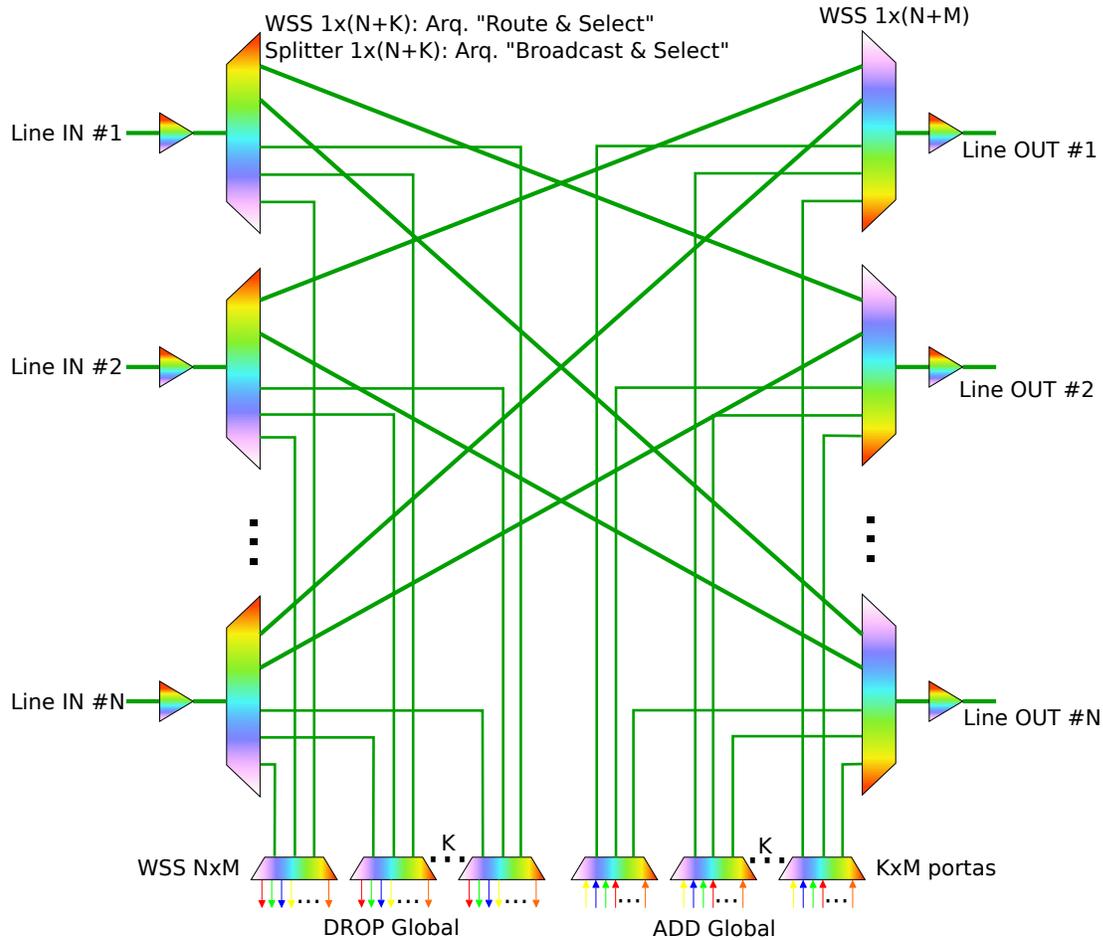


Figura 3 – ROADM WSS CDC

Como os WSS NxM não estão disponíveis comercialmente, a literatura (MAGALHAES *et al.*, 2013) descreve dois tipos de soluções para a construção de ROADMs-CDC. A primeira, mostrada na Figura 4a, é baseada em um módulo formado por *splitters*, chaves ópticas e *array* de amplificadores, denominada *Multicast Switch* (MCS). A segunda, mostrada na Figura 4b apresenta solução utilizando *Optical Cross-Connect* (OXC). Ambas soluções possuem vantagens e desvantagens. A primeira, apresenta a vantagem de ser *flex-grid*, ou seja, não é amarrada à grade DWDM do ITU-T, suportando canais com diferentes larguras. Porém, tal solução apresenta alta perda de inserção, requerendo grande quantidade de amplificadores ópticos, de forma a ocupar muito espaço físico, possuir alto consumo e custo. Já a segunda ocupa menos espaço e possui menor consumo, porém, apresenta altos custos, além de não suportar grade flexível.

Com a dificuldade em obter componentes com custo-benefício adequado, incluindo requisitos de espaço e consumo, diversas propostas têm surgido na literatura, seja para a

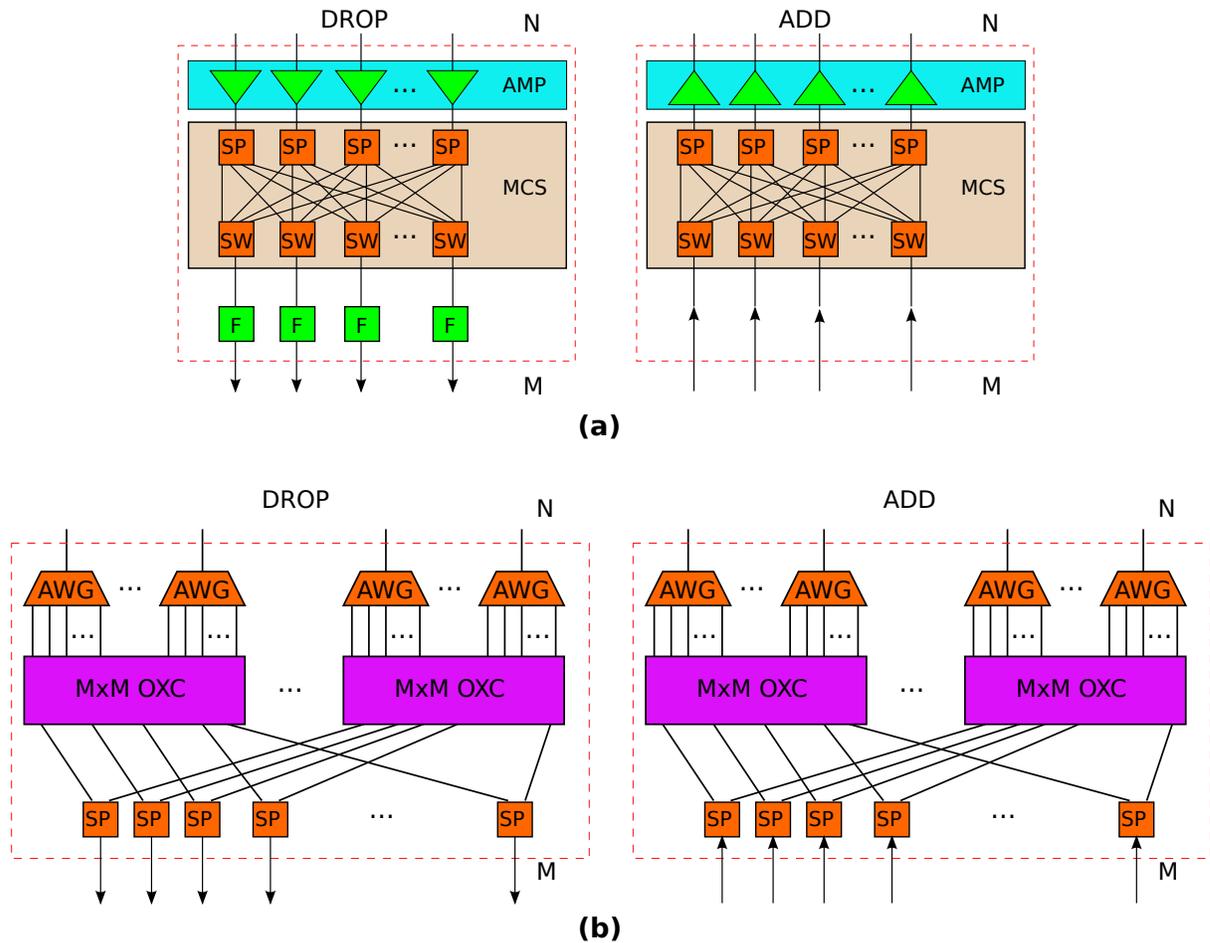


Figura 4 – Arquiteturas de ROADMs WSS do tipo CDC

reorganização de blocos existentes, seja para a utilização de novos componentes. Uma proposta com potencial utiliza *splitters* variáveis (WAY *et al.*, 2013) para a construção de um nó ROADMs parcialmente CDC, que permite a diminuição da contenção pela adição de caminhos alternativos que permitem a reutilização de comprimentos de onda em um mesmo banco de ADD ou DROP, conforme ilustrado na Figura 5. Os autores apresentam simulações do tipo Monte Carlo, mostrando que, para determinadas configurações de carga e quantidade de ADD/DROP, a adição de quatro caminhos alternativos permite a obtenção de uma taxa de bloqueio devido à contenção praticamente nula.

Vários estudos têm sido publicados sobre o impacto da contenção dos ROADMs do tipo CD e parcialmente CDC na taxa de bloqueio dos mecanismos de *Routing and Wavelength Assignment* (RWA) ou *Routing and Spectrum Assignment* (RSA) das redes. Um estudo realizado sobre a topologia da rede NFSNET (KIM *et al.*, 2012) demonstrou que a probabilidade de bloqueio do algoritmo de RWA para um ROADMs que utiliza OXCs modulares de 16x16 do lado cliente para resolução de contenção tem probabilidade de bloqueio similar

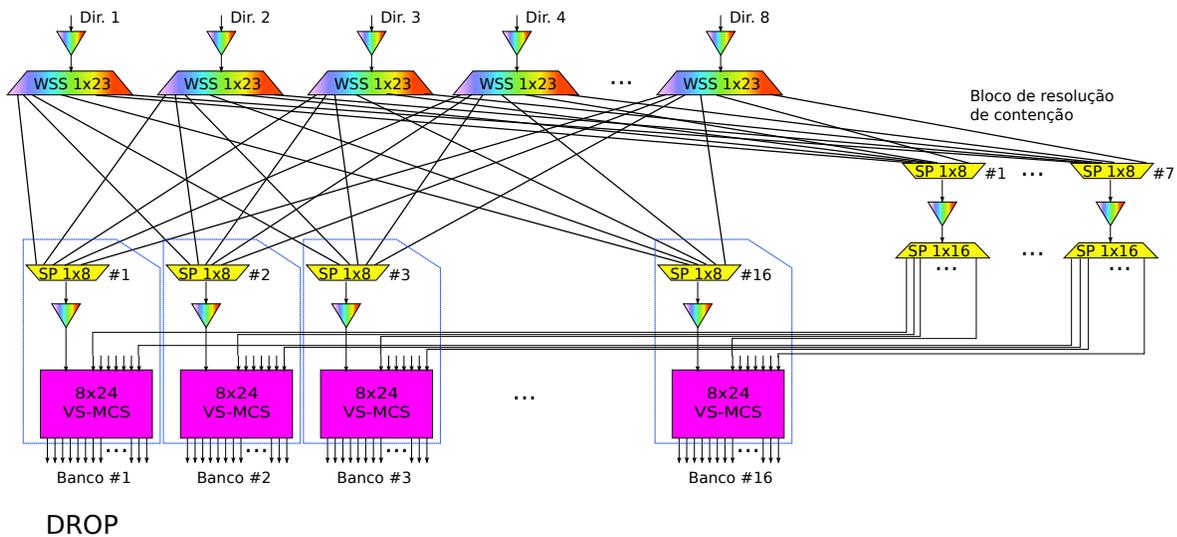


Figura 5 – ROADM WSS CDC com rotas alternativas para resolução de contenção

a um ROADM completamente CDC quando o algoritmo de RWA é otimizado para minimizar contenções. Outro estudo (SHEN *et al.*, 2012) demonstrou que a funcionalidade que mais contribui para a diminuição da probabilidade de bloqueio dos mecanismos de RWA é o suporte a *colorless*.

### 2.1.1.2 Comutação óptica com conversão OEO parcial

O conceito de comutação de pacotes ópticos *Optical Packet Switching* (OPS) foi introduzido e proposto como solução para atender às demandas de grande crescimento de tráfego há mais de uma década (O'MAHONY *et al.*, 2001). Nesta estratégia, os cabeçalhos dos pacotes ópticos podem ser processados tanto de forma elétrica como óptica, enquanto o *payload* deve ser armazenado em *buffers* ópticos (linhas de atraso) antes de ser encaminhado ao próximo elemento de rede.

Outra estratégia, também proposta há aproximadamente uma década, denomina-se *Optical Burst Switching* (OBS) (CHEN *et al.*, 2004), na qual somente há necessidade de conversão OEO de pacotes de controle, sendo que os pacotes de clientes são encaminhados de forma óptica. Diferentemente de redes DWDM com ROADM, OPS e OBS permitem a obtenção de multiplexação estatística. Ambas as estratégias possuem limitações quanto à dificuldade e custo de implementação, contenção, compatibilidade com sinais clientes, entre outras, de forma que não obtiveram sucesso comercialmente, e atualmente não há indícios de que obterão a curto ou médio prazo, já que o constante aumento nas velocidades de transmissão aumenta os desafios de implementação.

### 2.1.1.3 Comutação em redes ópticas de transporte com conversão OEO

Uma das tecnologias que mais tem se destacado em redes ópticas de transporte nos últimos anos é OTN (ITU-T, 2009), que pode ser dividida em três funções principais: transporte, multiplexação e comutação. O transporte OTN já é utilizado há mais de uma década em implementações de transponders DWDM, em que o sinal é encapsulado (*Digital Wrapper*) em quadros OTN, com suporte a variados cabeçalhos, algoritmos de *Forward Error Correction* (FEC) e funcionalidades de *Operations and Management* (OAM), suportando mapeamento de variados tipos de sinais cliente de forma transparente, incluindo Sonet/SDH, Ethernet, Fibre Channel, e Infiniband. A Multiplexação OTN também já é largamente utilizada em redes de transporte, permitindo a combinação de diferentes sinais cliente de variadas velocidades em um agregado OTN de maior velocidade. Em 2009, foi definida uma nova versão da recomendação G.709 (ITU-T, 2009) que permitiu maior flexibilidade na hierarquia de multiplexação OTN, incluindo por exemplo o conceito de ODU-Flex, que permite a multiplexação de sinais cliente de taxas variáveis e/ou não padronizadas. Mais recentemente, foi introduzido o conceito de comutação OTN, que, de forma simplista, permite que os transponders e muxponders DWDM sejam integrados através de uma matriz de comutação, permitindo que sinais ODU sejam comutados entre diferentes interfaces DWDM.

Em termos de análise de arquiteturas de redes, enquanto (MELLE *et al.*, 2013) apresenta resultados que apontam benefícios em termos de quantidade de interfaces e OPEX na utilização de comutadores OTN com interfaces DWDM integradas em redes de longa distância, comparando-se com arquiteturas DWDM com ROADMs e transponders *standalone*, (MEZHOUDI; HU, 2010) acrescentam que a utilização de comutadores OTN, juntamente com soluções com integração IP/Óptica em roteadores, pode reduzir a quantidade de roteadores de núcleo, bem como interfaces nestes, permitindo redução em CAPEX, enquanto simplifica a arquitetura da rede.

Atualmente, há dois tipos de arquiteturas de equipamentos com suporte a comutação OTN: *Packet Optical Transport Platforms* (P-OTP) e *Optical Transport Platforms* (OTP) (MRAZE; WAKELIN, 2011). Enquanto a primeira combina comutação de OTN e pacotes na mesma plataforma, a segunda é focada em transporte OTN. A convergência possibilitada pelas plataformas P-OTP é viabilizada pela utilização de matrizes de comutação baseadas em células, suportando tanto a comutação de tributários OTN como de pacotes. Para isso, o *Optical Internetworking Forum* (OIF) padronizou a interface *OTN over Packet Fabric Protocol* (OPFP) (MOK; GASS, 2011) de forma que dispositivos de processamento OTN comunicam-se com as matrizes de células, incluindo um método para o transporte de informações de sincronismo através da matriz. Outros tipos de matrizes de comutação que podem ser utilizadas em plataformas OTN, mas que não são compatíveis com comutação de

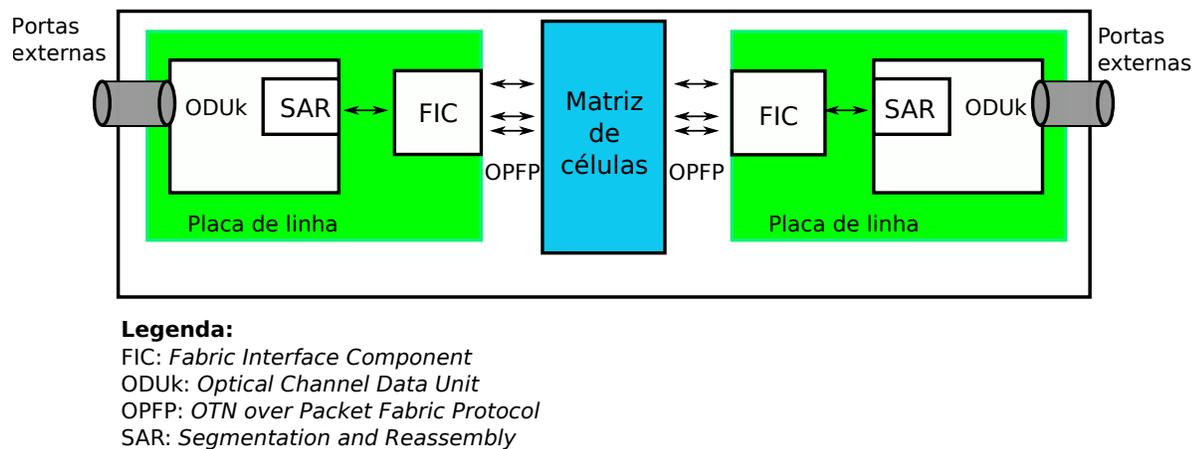


Figura 6 – Modelo de referência para comutador OTN baseado em matriz de células

pacotes, incluem as matrizes baseadas em TDM e os *cross-points* digitais.

A Figura 6 ilustra um modelo de referência para a implementação de um comutador OTN baseado em matriz de células. Neste modelo, o processador OTN extrai os tributários ODU do sinal de linha, segmenta e mapeia em células através do dispositivo *Fabric Interface* (FIC). Essas células são transferidas para outra placa de linha que realiza o processo inverso, mapeando novamente o sinal em outro tributário ODU, ou transmitindo através de uma interface cliente (e.g., Ethernet, FC).

### 2.1.2 Plano de controle GMPLS

A evolução das tecnologias de transmissão óptica tem permitido o contínuo crescimento na capacidade das redes de acesso, metropolitanas e longa distância, de forma a atender a crescente demanda da Internet e das redes privadas. Os principais fornecedores de equipamentos de transmissão já estão fornecendo sistemas ópticos reconfiguráveis com ROADMs e transponders coerentes de 100Gbps em larga escala. Entre os anos de 2014 a 2016, serão implementados em maior escala equipamentos do tipo *Packet-Optical Transport System* (P-OTS) com funcionalidades de comutação na camada elétrica, bem como ROADMs com características CDC. A evolução das redes para taxas mais elevadas (200Gbps e mais) deverá ser realizada com a utilização de transmissores com formatos de modulação flexíveis e adaptáveis às condições da rede, de forma que a rede óptica deverá suportar grade flexível (*flexgrid*).

Desta forma, tais redes apresentam muitos desafios ao plano de controle, que deve ser multicamadas, com suporte serviços dinâmicos, diferentes esquemas de proteção e restauração, estratégias de otimização de rede através de *bypass* óptico (*cross-layer optimizations*)

e suporte a mecanismos de RSA. Órgãos de padronização como o IETF e OIF têm trabalhado em extensões aos protocolos GMPLS para atender a tais requisitos, incluindo extensões para suporte a comutação na camada OTN, bem como suporte a grades flexíveis na camada óptica. Desta forma, é fundamental que o plano de controle GMPLS evolua para permitir a introdução das novas linhas de equipamentos com funcionalidades mais complexas, mantendo a simplicidade percebida pelas equipes de implantação e operação das redes, garantindo a confiabilidade e flexibilidade na configuração e prestação dos serviços.

A pilha de protocolos GMPLS é composta por três protocolos base: *Resource Reservation Protocol – Traffic Engineering* (RSVP-TE) para sinalização, *Open Shortest Path First – Traffic Engineering* (OSPF-TE) para roteamento e *Link Management Protocol* (LMP) para gerenciamento de enlaces.

#### 2.1.2.1 Proteção e restauração em redes GMPLS

O grande motivador para a implementação de plano de controle GMPLS em redes de transporte é a capacidade de implementação de mecanismos de sobrevivência (restauração de circuitos) que permitam aumentar o percentual de utilização dos enlaces. Conforme definido na RFC 4427 (MANNIE; PAPADIMITRIOU, 2006), os esquemas de proteção listados a seguir são aplicáveis a *Label Switched Paths* (LSPs) ou a segmentos em redes de transporte com plano de controle GMPLS:

**Proteção 1 + 1**, em que o tráfego é duplicado na origem e enviado através de dois caminhos, denominados LSP/segmento de “trabalho” e de “proteção”. Na recepção, o tráfego é selecionado de um dos dois LSPs ou segmentos.

**Proteção 1 :  $N(N \geq 1)$** , na qual há  $N$  LSPs/segmentos que transportam o tráfego normal e um LSP/segmento que pode transportar tráfego extra (e.g., tráfego de baixa prioridade que pode ser preemptado). No caso mais comum, o tráfego será enviado através de somente um LSP/segmento (ou o de trabalho ou o de proteção).

**Proteção  $M : N(M, N \geq 1, N \geq M)$** , em que há  $N$  LSPs/segmentos de trabalho transportando o tráfego normal e  $M$  LSPs/segmentos de proteção que podem transportar tráfego extra. No caso mais comum, o tráfego será enviado através de somente um LSP/segmento (ou o de trabalho ou o de proteção).

Os esquemas de proteção podem ser uni- ou bidirecionais. No segundo caso, é necessário coordenação entre ingresso e egresso durante a comutação para o LSP/segmento de proteção.

Conforme definido na RFC 4427 (MANNIE; PAPADIMITRIOU, 2006), os esquemas de restauração listados a seguir são aplicáveis a LSPs em redes de transporte com plano de controle GMPLS:

**Restauração pré-planejada**, em que um ou mais LSPs de restauração são instanciados entre o mesmo par ingresso/egresso que o LSP de trabalho. Neste caso, os recursos de restauração devem ser calculados e sinalizados a priori, mas não necessitam ser cross-conectados. Desta forma, não é possível o transporte de tráfego extra. A cross-conexão somente é realizada quando há sinalização de falha no LSP de trabalho, sendo necessária sinalização de restauração. Quando o LSP de trabalho é protegido por somente um LSP pré-planejado, este tipo de proteção é referido como re-roteamento 1 : 1 pré-planejado, sem tráfego extra.

**Restauração *Shared-Mesh***, a qual é definida como um caso particular de re-roteamento através de LSP pré-planejado que reduz a quantidade de recursos alocados para restauração, permitindo que múltiplos LSPs de restauração, que iniciam em nós de ingresso distintos compartilhem recursos.

**Restauração de LSP**, também referido como re-roteamento de LSP, na qual o LSP é sinalizado e cross-conectado após a ocorrência da falha no LSP de trabalho. Se adicionalmente o roteamento for realizado após a falha, o esquema pode ser referido como “*Full LSP Re-routing*”. O LSP é re-roteado na origem, podendo utilizar recursos não afetados pela falha do LSP de trabalho.

Conforme definido nas RFCs 4427 (MANNIE; PAPADIMITRIOU, 2006) e 4428 (PAPADIMITRIOU; MANNIE, 2006), o processo de detecção e recuperação (restauração e/ou proteção) pode ser dividido em cinco fases:

1. Detecção da falha.
2. Localização e isolamento da falha.
3. Notificação da falha.
4. Recuperação (proteção ou restauração).
5. Reversão (normalização).

Detecção da falha é uma tarefa que deve ser realizada pelo plano de transporte, de forma que deve haver algum mecanismo que permita que a falha seja informada ao plano de

controle. Independentemente da tecnologia de transporte (e.g., SDH ou OTN), é importante que o processo de detecção de falha que resultará na atuação do mecanismo de recuperação resulte em uma das seguintes indicações:

- *Signal Degrade (SD)*: indica que o canal de dados sendo transportado está sofrendo degradação (como por exemplo, no caso em que a *Bit Error Rate (BER)* está acima do limite.
- *Signal Failure (SF)*: Indica que o sinal está interrompido.

A localização da falha permite que o processo de restauração possa ser realizado de forma mais otimizada, permitindo, por exemplo, a realização de restauração de segmento. A localização da falha pode ser realizada no plano de dados, como por exemplo, através dos mecanismos de OAM das redes OTN, ou através do protocolo LMP no plano de controle.

Da mesma forma, a notificação da falha pode ser realizada pelo plano de dados, como, por exemplo, através de mensagens *Alarm Indication Signal (AIS)*, através dos mecanismos de OAM das redes OTN, ou através do plano de controle e é utilizada para informar às entidades decisórias (e.g., nó de egresso) para que seja realizado o processo de comutação para a proteção ou a restauração. No entanto, um único evento de falha pode gerar múltiplos alarmes, de forma que é desejável que seja implementado algum mecanismo de correlação de falhas.

Recuperação de falhas pode ser realizada através de proteção, restauração, ou uma combinação de ambos. De modo geral, para o caso de proteção de LSPs, a detecção da falha e comutação são de responsabilidade do plano de transporte, sendo que o plano de controle também é responsável em alguns casos, (por exemplo, no caso de comutação bidirecional). Já no caso de restauração, o plano de transporte está envolvido na detecção da falha, porém, o plano de controle é responsável por realizar a restauração. Desta forma, a recuperação de falhas envolve aspectos dependentes da tecnologia, como uni- ou bi-direcionalidade, separação do canal de controle do canal de dados, suporte a restauração de segmento, entre outros.

### 2.1.2.2 GMPLS em redes multicamadas e multi regiões

Conforme detalhado na RFC 5212 (SHIOMOTO *et al.*, 2008), GMPLS estende MPLS para suportar múltiplas tecnologias de comutação, incluindo comutação de: pacotes, frames (camada 2), TDM, lambdas (e.g., via ROADMs) e fibra (e.g., via OXC). Desta forma, o plano de controle pode ser dividido em regiões, sendo que cada região é definida por uma tecnologia de comutação. Uma região pode ser dividida em mais de uma camada, por exemplo,

em redes OTN, em que há duas camadas de comutação de ODU (ODU<sub>k</sub>, ODU<sub>j</sub>), onde ODU<sub>k</sub> representa um ODU de mais alta ordem e um ODU<sub>j</sub> representa o de mais baixa ordem.

Do ponto de vista do GMPLS, uma *MultiLayer Network* (MLN) consiste em uma rede com uma único domínio de engenharia de tráfego (*TE Domain*), com múltiplas camadas dentro de um mesmo tipo de comutação, ou uma *MultiRegion Network* (MRN), suportando pelo menos dois tipos de comutação (e.g., Lambda e ODU). Em um mesmo domínio TE pode haver *single-switching-type-capable* LSRs e *multi-switching-type-capable* LSRs, que podem operar com somente uma instância de plano de controle suportando as múltiplas regiões ou tecnologias de comutação, com uma única *Traffic Engineering Database* (TED) consolidada com informações sobre todas as regiões e camadas. A utilização de uma única instância de GMPLS por *Network Element* (NE) para o controle de múltiplas camadas/regiões possui as seguintes vantagens:

- Evita múltiplas instâncias de plano de controle em um mesmo NE com suporte a múltiplas camadas de comutação no plano de dados, diminuindo complexidade nas interações entre estes planos e diminuindo requisitos de capacidade de processamento e memória.
- Unificação de espaços de endereçamento.
- Facilita colaboração entre camadas co ponto de vista de canais de controle entre diferentes NEs.
- Facilita o gerenciamento de recursos e aplicação de políticas, já que podem ser realizados de forma integrada entre camadas.
- Aumenta a flexibilidade de técnicas de engenharia de tráfego, por exemplo, aumentando as potencialidades de otimização de recursos.

O grau de integração provido permite, por exemplo, o estabelecimento dinâmico de *Forwarding Adjacency LSPs* (FA-LSPs) na camada servidora para o estabelecimento de uma topologia virtual na camada cliente mais adequada aos requisitos de engenharia de tráfego (e.g., TE *auto-mesh* (VASSEUR *et al.*, 2007)). Adicionalmente, é possível o estabelecimento de LSPs multicamada fim-a-fim através do disparo automático da criação de FA-LSPs na camada servidora. O *Type Length Value* (TLV) do tipo *Interface Switching Capability Descriptor* (ISCD) (KOMPELLA; REKHETER, 2005), identifica a *Interface Switching Capability* (ISC), o tipo de codificação, a granularidade de comutação, permitindo a caracterização das camadas associadas. Um TE-Link pode conter mais de uma ISCD, de forma a pertencer a mais de uma camada. Um FA-LSP pode ser divulgado como um TE-Link para a camada

superior, de forma a prover uma *Virtual Network Topology* (VNT) para a camada superior. No entanto, informações sobre *Shared Risk Link Group* (SRLG) devem ser disponibilizadas para a camada superior, de forma que rotas de proteção na VNT não compartilhem os mesmos recursos físicos do mesmo grupo de risco (como por exemplo a mesma fibra óptica em um determinado trecho). Neste caso, o TE-Link criado na camada cliente não é multicamadas do ponto de vista da camada servidora, já que é estabelecido através de uma adjacência de encaminhamento. As motivações para a utilização de VNTs incluem maior separação entre camadas, incluindo separação do plano de controle entre camadas.

A RFC 6001 (PAPADIMITRIOU *et al.*, 2010) define extensões de roteamento e sinalização ao GMPLS para atender aos requisitos para operação em redes MLN/MRN. No entanto, observa-se que, conforme definido na RFC 5339 (ROUX; PAPADIMITRIOU, 2008), a maioria dos requisitos para a implementação de redes com plano de controle MLN/MRN envolve a implementação de mecanismos, procedimentos e algoritmos que estão fora do escopo dos protocolos GMPLS, como, por exemplo, a definição de algoritmos e políticas para a criação automática de *Forwarding Adjacencys* (FAs).

### 2.1.3 PCE (*Path Computation Element*)

O cálculo de caminhos baseado em restrições é uma funcionalidade fundamental em redes GMPLS. Tal funcionalidade pode tornar-se complexa em redes multicamada, com múltiplos domínios e de grande escala, podendo requerer componentes computacionais com requisitos diferenciados. Desta forma, a RFC 4655 (FARREL *et al.*, 2006) especifica a arquitetura que utiliza um elemento para cálculo de caminhos *Path Computation Element* (PCE) para tratar deste problema. A arquitetura PCE é composta por dois componentes principais:

- *Path Computation Client* (PCC): representa uma aplicação ou elemento que solicita cálculo de rotas ao PCE. Comumente, esta aplicação está localizada no nó de ingresso do LSP a ser calculado.
- PCE: consiste em uma entidade capaz de calcular rotas com restrições em um grafo de rede. Tal entidade é uma aplicação que pode ser executada em um ou mais elementos de rede, em um ou mais servidores de gerência ou em servidores independentes, dedicados ou não.

Diferentemente do que pode parecer, o objetivo principal da arquitetura PCE não é centralizar funções de rede que atualmente são distribuídas. O PCE pode ser visto como uma entidade que permite a realização de operações cooperativas de funcionalidades distribuídas,

de forma a viabilizar principalmente o cálculo de caminhos com restrições com altos requisitos computacionais, bem como para o cálculo de caminhos inter-domínios.

A Figura 7 ilustra quatro diferentes arquiteturas para a utilização de PCE.

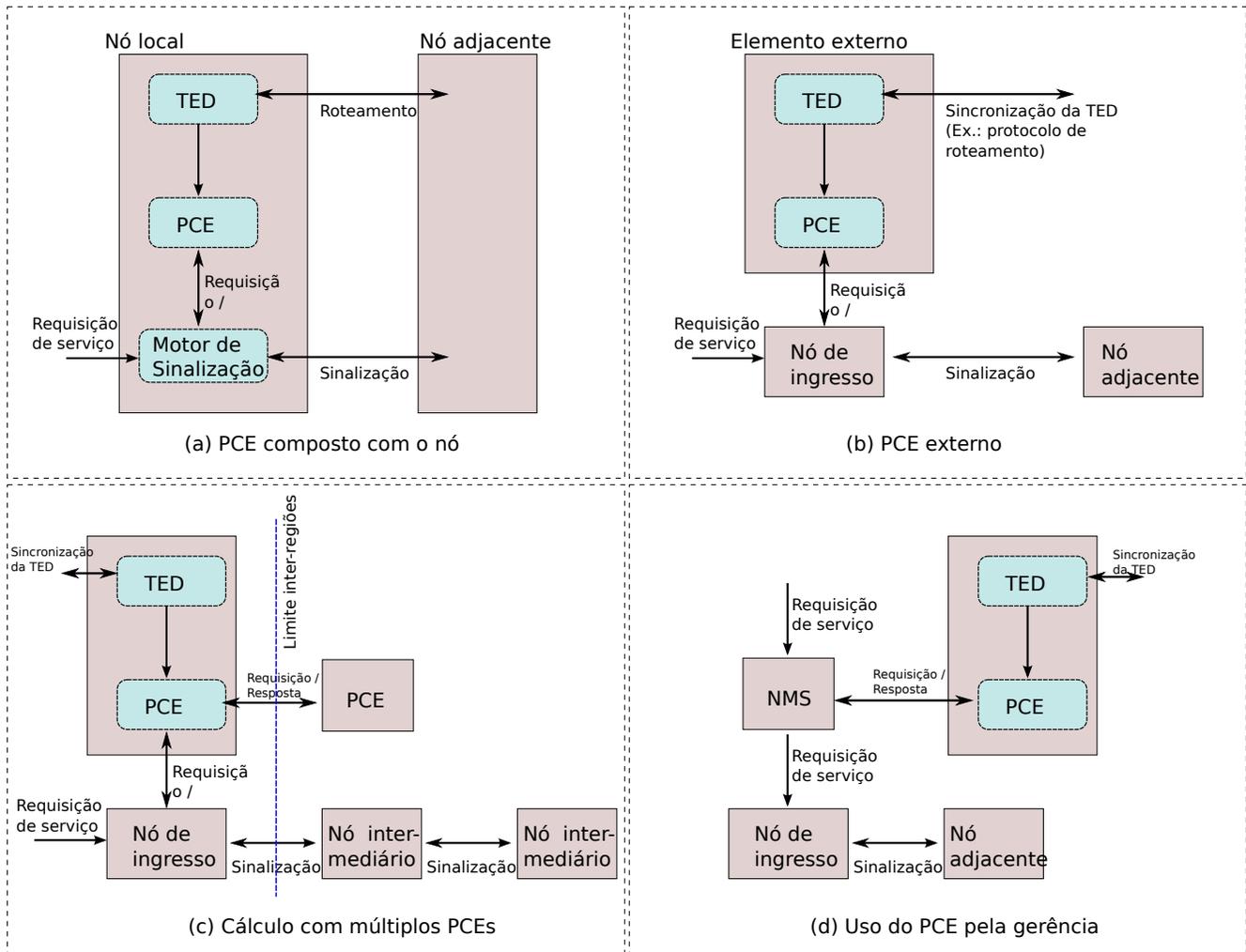


Figura 7 – Possíveis arquiteturas para solução PCE

1. PCE composto com o nó de ingresso da rede: o PCE utiliza a TED do nó, mantida pelo protocolo de roteamento.
2. PCE externo ao nó: neste caso, o PCE é compartilhado, podendo ser acessado por diferentes nós da rede. As formas de manutenção da TED do PCE serão discutidas nas próximas seções.
3. Cálculo de rotas envolvendo múltiplos PCEs: em cenários com múltiplas regiões, o PCE local não tem visão sobre a topologia detalhada de todas as regiões. Desta forma, pode-

se utilizar comunicação entre PCEs para cálculo da rota. Observe que esta estratégia pode ser utilizada para aumentar a escalabilidade de um domínio ou mesmo entre domínios confiáveis (nos quais pode-se expor a topologia).

4. Utilização do PCE pela gerência de redes: no caso de criação de conexões do tipo soft-permanentes, solicitadas pela ferramenta de gerência de redes e/ou serviços, pode ser necessário que a gerência informe ao nó de ingresso o caminho explícito do LSP. Neste caso, a ferramenta de gerência pode solicitar ao PCE o cálculo da rota. Outro cenário possível é aquele em que o provisionamento é feito nó a nó pela gerência.

Um domínio pode adotar arquitetura de implementação de PCE centralizado ou distribuído. No modelo centralizado, somente um PCE responde às requisições dos clientes, apesar de poderem haver um ou mais PCEs reserva no domínio. Já no modelo de PCEs distribuídos, um determinado PCC pode utilizar um ou mais PCEs para o cálculo de rotas, ou diferentes PCCs podem utilizar diferentes PCEs no mesmo domínio.

#### 2.1.3.1 Descoberta de PCEs e balanceamento de carga

O PCC deve conhecer o endereço do PCE para realizar solicitações de cálculo de caminhos. Para isso, pode ser realizada configuração manual ou descoberta automática através de algum protocolo. Pode haver configuração ou descoberta de mais de um PCE para fins de balanceamento de carga. A RFC 5088 (ROUX *et al.*, 2008) define extensões ao protocolo *Open Shortest Path First* (OSPF) para permitir descoberta automática do PCE pelos PCCs em um mesmo domínio *Interior Gateway Protocol* (IGP). Para isso, é definida uma nova TLV denominada “*PCE Discovery TLV*” a ser transportada pelos *Link State Advertisements* (LSAs) do tipo “*OSPF Router Information*”. As informações divulgadas incluem: o endereço IP do PCE, o escopo do PCE (i.e., intra-area, inter-area, inter-AS, or inter-camadas), domínio(s) para o(s) qual(is) os PCE pode calcular rotas, bem como capacidades adicionais, como restrições suportadas no cálculo de rotas e suporte a priorização no tratamento de requisições.

#### 2.1.3.2 Sincronização da TED

A RFC 4655 (FARREL *et al.*, 2006) especifica diferentes opções para a sincronização da TED pelo PCE:

- Participação no IGP, através da criação de uma adjacência com pelo menos um nó da rede. Observe que o PCE pode não estar fisicamente adjacente ao nó da rede, mantendo a adjacência IGP através de um túnel.

- Sincronização *out-of-band*: através de mecanismo a ser definido, o qual pode ser incremental ou utilizar transferência completa.
- Obtenção de informações através de fontes externas, como, por exemplo, configuração de LSPs pelo operador.

### 2.1.3.3 Manutenção de estado pelo PCE

O PCE pode ser *stateful* ou *stateless*. No primeiro caso, além de manter uma sincronização da TED com a rede, o PCE mantém sincronizadas informações sobre o conjunto dos caminhos já computados e estabelecidos, incluindo relações de caminhos de proteção e restauração pré-planejada, e utilização de recursos da rede. Desta forma, tais informações podem ser utilizadas para o cálculo de novos caminhos. Observe que a manutenção de estados pode melhorar a capacidade do PCE de realizar cálculos mais otimizados, porém com o custo de um maior *overhead* de sincronização e manutenção de estados, o que, em uma rede muito grande, pode ter questões de escalabilidade.

Por outro lado, um PCE *stateless* não precisa manter uma base com informações sobre o caminho calculado para cada requisição, podendo realizar os cálculos de rota utilizando somente a requisição e a TED obtida da rede. No entanto, é possível que, neste caso, o PCC em questão envie juntamente com a requisição a lista de caminhos pré-computados para que o PCE possa, por exemplo, fazer distribuição dos diferentes caminhos deste PCC por diferentes rotas para diminuir impacto de falhas na rede para caminhos de um mesmo NE. A utilização de PCE *stateful* pode permitir a visibilidade necessária para a criação e otimização de VNT em MLN, através da criação de novas FA na camada servidora sempre que os recursos nos TE-Links da camada cliente não forem suficientes ou puderem ser otimizados.

### 2.1.3.4 PCEP

O PCEP (VASSEUR; ROUX, 2009) é um protocolo definido para comunicação entre PCC e PCE bem como entre diferentes PCEs para solicitação de cálculo de rotas. O PCEP não inclui mecanismos para descoberta de PCE pelos PCC nem para a sincronização da TED pelo PCE.

A RFC 5521 (OKI *et al.*, 2009) define extensões ao PCEP para permitir que um PCC informe restrições relativas à exclusão de nós, links e/ou SRLGs em uma solicitação de rota. Estas exclusões podem ser úteis quando se quer evitar que a rota utilize determinados recursos por motivos administrativos ou mesmo para o caso em que rotas disjuntas de proteção são calculadas por diferentes PCEs. A RFC 5541 (ROUX *et al.*, 2009) define extensões ao PCEP para que seja possível que o PCE informe quais funções objetivo são suportadas no cálculo

de rotas, bem como para que o PCC informe quais funções objetivo ele deseja que sejam utilizadas na solicitação de cálculo. São definidas seis funções objetivo:

- Funções aplicáveis para o cálculo de um caminho:
  - *Minimum Cost Path* (MCP)
  - *Minimum Load Path* (MLP)
  - *Maximum residual Bandwidth Path* (MBP)
- Funções aplicáveis para o cálculo de um conjunto de caminhos:
  - *Minimize aggregate Bandwidth Consumption* (MBC)
  - *Minimize the Load of the most loaded Link* (MLL)
  - *Minimize the Cumulative Cost of a set of paths* (MCC)

Pode ser vantajosa a realização de cálculo de vários caminhos de forma simultânea, de forma que o PCE possa encontrar soluções mais ótimas do ponto de vista da rede. Tais otimizações são referidas como *Global Concurrent Optimization* (GCO) e são desejáveis quando o PCC solicitante é um *Network Management System* (NMS). A RFC 5557 (LEE *et al.*, 2009) descreve requisitos e define extensões ao PCEP para permitir que um conjunto de caminhos seja calculado de forma concorrente, permitindo o emprego de GCO no processo. Em redes multicamada o escopo da re-otimização pode ser a alteração de capacidade dos LSPs atuais das camadas servidoras, ou mesmo a criação de novos de *bypass* nestas camadas, criando novas FAs a serem utilizadas pelas camadas cliente no processo de re-otimização. Este processo é denominado “Reconfiguração da VNT”. As principais extensões ao PCEP para suporte a GCO incluem:

- Indicador que a requisição é relativa a uma GCO.
- Uma *Global Objective Function* (GOF), podendo ser MBC, MLL ou MCC.
- *Global Constraints* (GC), incluindo:
  - Máxima utilização de link;
  - Mínima utilização de link;
  - Fator de *overbooking*;
  - Máximo número de saltos dos LSPs;
  - Nós ou links de exclusão.
- *Global Concurrent Vector* (GCV) contendo todas as solicitações de caminho individuais.

### 2.1.3.5 Características e requisitos específicos para redes GMPLS

Conforme exposto na RFC 7025 (OTANI *et al.*, 2013), LSPs em redes GMPLS diferenciam-se dos de redes puramente MPLS em aspectos como bidirecionalidade com alocação de banda simétrica ou assimétrica, tipos de comutação, de codificação, tipo de sinal cliente, atributos de proteção, utilização de interfaces *unnumbered*, controle explícito do rótulo durante o cálculo do caminho, entre outros.

Tipos de comutação incluem: *Packet-Switch Capable* (PSC), TDM, *Lambda Switch Capable* (LSC), *Fiber-Switch Capable* (FSC); tipos de codificação incluem pacotes, SDH, ODU, Lambda; e tipos de sinal cliente incluem *Fibre Channel* (FC), Ethernet e SDH.

Embora os documentos que especificam a arquitetura PCE incluam o suporte a redes MPLS e GMPLS, o foco dado inicialmente é em redes MPLS. Desta forma, (MARGARIA *et al.*, 2014) complementam os padrões adicionando extensões necessárias para redes GMPLS, incluindo OTN e *Wavelength Switched Optical Networks* (WSON). Como exemplo, para o caso de redes WSON, o processo de cálculo de caminhos envolve a realização de RWA seguindo uma das seguintes estratégias:

- R&WA combinados, em que o PCE realiza os processos de roteamento e alocação de lambda de forma integrada.
- R+WA em duas etapas, em que um PCE realiza o roteamento e outro PCE (ou outro processo no mesmo PCE) realiza a alocação de lambdas.
- *Routing + Distributed Wavelength Assignment* (R+DWA) em que o PCE realiza somente o roteamento, deixando para o processo de sinalização (RSVP-TE) realizar a alocação e distribuição de lambdas.

O próximo capítulo apresenta novos paradigmas para o controle e gerência de redes, com foco em redes ópticas de transporte.

## 3 Novos Paradigmas em Controle de Redes

Este capítulo apresenta fundamentos teóricos e estado da arte de redes autonômicas, gestão de redes com base em políticas (PBNM), redes definidas por software (SDN) e virtualização de redes. Adicionalmente, são descritas potenciais aplicações destes conceitos em redes ópticas de transporte NG-OTN.

### 3.1 Redes autonômicas

#### 3.1.1 Computação autonômica

Uma das primeiras menções a sistemas computacionais autonômicos com uma abrangência mais ampla foi em um manifesto da *International Business Machines Corporation* (IBM) em 2001 (KEPHART; CHESS, 2003) apontando para a dificuldade em gerenciar a crescente complexidade dos sistemas computacionais, que ultrapassa os limites da capacidade humana. O desenvolvimento de sistemas computacionais com suporte à execução de funções de controle de forma autonômica foi citado como uma das únicas opções para permitir a manutenção do ritmo de crescimento de tais sistemas.

Pode ser feita uma analogia entre *Autonomic Computing* (AC) e o sistema nervoso que controla a velocidade de batimento cardíaco, temperatura corporal, funcionamento dos órgãos como os pulmões, coração e rins de forma inconsciente. Desta forma, a parte consciente do cérebro fica livre destas funções primárias, porém vitais, podendo focar nas funções conscientes.

AC é definido como sistemas de computação que podem gerenciar a si próprios, prometendo uma transformação na interação entre operadores e sistemas de computação. Particularmente, AC vai na direção de mudar a especificação do comportamento dos sistemas de procedimentos de configurações de elementos de baixo nível para políticas orientadas por negócios.

Os autores Kephart e Chess (2003) afirmam que o primeiro passo necessário na pesquisa destes tipos de sistemas é a determinação das similaridades entre os sistemas de computação autonômica e organismos biológicos, sociedades auto-gerenciadas, entre outros. Esta estratégia auxilia na determinação de como estes sistemas deverão funcionar e quais são os obstáculos a serem enfrentados pelos pesquisadores para projetar os sistemas autonômicos. Adicionalmente, são definidos requisitos para estes sistemas, descritos a seguir:

» **Auto-gerência:** os sistemas AC deverão ajustar parâmetros de operação de acordo com falhas em hardware ou software, causadas por iniciativas maliciosas ou acidentalmente. Para isso os sistemas devem monitorar continuamente seu próprio uso. Estes sistemas devem ser capazes de realizar atualizações, testes de regressão para verificação da correta atualização e, em caso de erros, devem ser capazes de voltar ao estado inicial. O objetivo final é que os padrões e mecanismos de migração de aplicações e dados sejam totalmente transparentes, podendo até serem desconhecidos pelos administradores de redes.

» **Auto-configuração:** a tarefa de instalação, configuração e integração em grandes sistemas é complexa, cara, leva tempo e é susceptível a erros, além de necessitarem de diversas equipes cada uma especializada em diferentes partes do sistema. Sistemas AC deverão configurar a si próprios automaticamente de acordo com políticas de alto nível que deverão representar objetivos no nível de negócios que deverão especificar o que é desejado e não como deve ser feito. Quando um novo componente for introduzido, este deve incorporar-se de forma transparente, sendo que o restante do sistema deverá adaptar-se a sua presença.

» **Auto-otimização:** redes e sistemas distribuídos possuem centenas de características ajustáveis, que devem ser configuradas de forma adequada para que o sistema como um todo opere de forma ótima. Porém, poucas pessoas sabem como otimizar estes parâmetros. Estes sistemas são frequentemente integrados com outros sistemas igualmente complexos. Consequentemente, o ajuste de desempenho de ambos os sistemas deve ser modificado de forma que o efeito no sistema como um todo seja novamente de melhor desempenho. Sistemas autônomicos devem continuamente procurar formas de melhorar sua própria operação, identificando e executando configurações que os tornem mais eficientes em desempenho e custo.

» **Auto-reparo:** sistemas AC deverão detectar, diagnosticar e reparar problemas localizados resultantes de bugs ou falhas em software e hardware. Para isso, diversas técnicas poderiam ser usadas, permitindo a realização do diagnóstico, comparação deste com correções de software conhecidas (patches), instalação destas correções e re-teste.

» **Auto-proteção:** sistemas AC deverão ser autoprotetidos em dois sentidos. Deverão defender o sistema como um todo contra problemas de larga escala, problemas correlacionados provenientes de ataques maliciosos ou falhas em cascata. Eles devem também antecipar problemas com base em informações prévias coletadas por sensores e tomar medidas para evitar ou mitigar estes problemas.

### 3.1.2 Redes autônomicas

Apesar de as redes IP terem sido desenvolvidas para desempenharem suas funções de forma autônoma, por exemplo através de protocolos de roteamento dinâmicos como OSPF e *Intermediate System to Intermediate System* (IS-IS), descoberta automática de endereços

entre outras funções, a evolução das redes para prover cada vez mais diferentes tipos de serviços, com diferentes características, levou a que grande parte das funções tenham seu comportamento configurado nos elementos de rede. Desta forma, com o aumento na complexidade das redes, o conceito de redes autonômicas veio à tona com o objetivo de trazer de volta a inteligência das redes, permitindo que as funções de controle sejam realizadas por algoritmos embarcados na rede e não configurados por operadores humanos (BEHRINGER *et al.*, 2013). O mesmo conceito pode ser estendido a redes ópticas de transporte, já que a complexidade na operação destas tem aumentado de forma expressiva com a adição de elementos reconfiguráveis, o surgimento do conceito de redes ópticas elásticas, bem como a pressão pela maior integração entre camadas.

Um *framework* para redes autonômicas é proposto por Behringer *et al.* (2013), no qual estas são definidas como redes compostas por elementos de rede autonômicos, os quais comunicam-se através de um plano de controle auto-organizável. O *framework* inclui: o conceito de agentes autonômicos de serviços que implementam comportamento autonômico para determinadas funções ou serviços; agentes com suporte a autoconhecimento e conhecimento da rede; políticas de alto nível da rede, a serem interpretadas pelos agentes autonômicos, de forma que o objetivo global seja instanciado em seu escopo local; *loops* de realimentação; um agente de usuário, responsável por ser um *front-end* com usuários ou aplicações; bem como um plano de controle autonômico, permitindo que os diferentes elementos de rede comuniquem-se, realizando rotinas como descoberta de topologia e realimentação dos *loops* de controle.

O ETSI estabeleceu um *Industry Specification Group* (ISG) denominado *ETSI ISG - Autonomic network engineering for the self-managing Future Internet* (AFI) que especificou cenários, casos de uso e definição de requisitos para redes autonômicas, considerando cenários da Internet futura (ETSI Group Specification, 2011). Adicionalmente, foi definida uma arquitetura de referência para redes autonômicas, cognição e auto-gerenciamento neste contexto denominada *Generic Autonomic Network Architecture* (GANA) (ETSI Group Specification, 2013), a qual define blocos funcionais e seus pontos de referência.

### 3.1.3 Redes ópticas cognitivas

Conforme proposto por Zervas e Simeonidou (2010), a utilização de cognição em redes ópticas de transporte é um dos principais caminhos no sentido da implementação de automação na operação destas redes, às quais são cada vez mais complexas. Para isso, as redes ópticas devem ser “cientes” (ter percepção das condições atuais), adaptativas (planejar, decidir e agir em determinadas condições), e aprender sobre estas adaptações usando tais informações para auxiliar em decisões futuras. Os autores introduzem uma arquitetura

para redes ópticas cognitivas, considerando múltiplas camadas, incluindo planos de dados, controle, gerência, serviços e aplicações. Os objetivos são definidos em uma camada definida como *Cognition Definition Language* (CDL), especificada através de ontologias de cognição. Adicionalmente, é definida uma arquitetura de nó óptico cognitivo, incluindo suporte à reconfiguração de parâmetros de operação nas diversas camadas, incluindo formato de modulação, mapeamento do serviço para o domínio e camada de transporte, além do suporte à reconfiguração da própria arquitetura do nó.

Os autores Miguel *et al.* (2013) descrevem resultados do projeto *Cognitive Heterogeneous Reconfigurable Optical Network* (CHRON), no qual é proposta uma arquitetura de rede óptica com uma entidade centralizada para a implementação de decisões utilizado cognição, implementando um “*loop cognitivo*” que inclui blocos de planejamento, decisão, ação, aprendizado e observação. Já Wei *et al.* (2012) propõem um *framework* para redes ópticas cognitivas que objetiva criar um substrato para otimização cruzada entre as camadas (*cross-layer*), permitindo realizar um balanço entre transparência e flexibilidade (comutação óptica ou elétrica), bem como permitir o controle da rede através de aplicações que interagem com o *framework*. Desta forma, pode-se interpretar que tais propostas caminham no sentido de uma arquitetura mais ampla de redes ópticas definidas por software, como discutiremos na seção 3.3.

## 3.2 Gerência de redes com base em políticas

No contexto de gerência de redes, políticas denotam a regulação de recursos da rede com base em critérios administrativos. Políticas são especificadas pelo gerente da rede com o objetivo de definir como os recursos ou serviços podem ou não ser utilizados. Um sistema de gerência por políticas transforma essas políticas em alterações na configuração da rede, permitindo que este sistema ofereça uma solução de controle automático de mecanismos específicos na rede.

Políticas oferecem abstrações para simplificar o gerenciamento de vários mecanismos na rede, tais como *Quality of Service* (QoS), mecanismos de segurança e outros. Na verdade, políticas são uma forma de traduzir contratos de uso da rede, tais como SLAs em parâmetros de configuração específicos nos elementos de rede. Uma política é formada por uma ou mais regras que governam como os recursos da rede serão utilizados. Por sua vez, regras são formadas por condições e ações. Ações devem ser tomadas para modificar o estado dos elementos de rede, caso as condições da política sejam satisfeitas.

Com a automatização dos mecanismos de configuração, o gerenciamento por políticas permite mudar o paradigma na gerência de redes, as quais podem ser gerenciadas como

um todo, não sendo necessária a configuração de elementos individualmente. No contexto de gerência de redes, políticas descrevem objetivos da empresa que eventualmente são representados por contratos tais como SLAs. Conforme o paradigma PBNM (STRASSNER, 2003), esses objetivos devem ser traduzidos em parâmetros de configuração nos elementos de rede. Um sistema de gerência de redes por políticas requer a adoção de uma linguagem para representação de políticas, um padrão para o armazenamento de políticas em um repositório, além de um esquema de transporte e interpretação das políticas. As seções seguintes discutem cada um destes componentes, além de opções de implementação.

Políticas podem ser utilizadas para governar uma vasta gama de mecanismos em redes. Os tipos mais comuns são políticas de controle de acesso; políticas de tráfego inter e intra-domínio, incluindo políticas de filtros, roteamento, engenharia de tráfego e QoS; políticas de proteção e restauração, incluindo cenários multicamadas; políticas de gerenciamento, entre outras.

Há na literatura uma vasta gama de trabalhos relacionados ao uso de políticas no gerenciamento de redes. Podemos classificar estes trabalhos de acordo com as seguintes áreas de concentração de contribuições, apresentando exemplos de referências para cada área:

- Arquitetura de sistemas de gerenciamento com base em políticas (PBNM) [J,L];
- Modelos de informação para sistemas PBNM (STRASSNER, 2002) e [K];
- Aplicação de PBNM para gerenciamento de sistemas e/ou tecnologias específicas, incluindo a criação de modelo informação para o gerenciamento destes [I];
- Criação de linguagens para a representação e compilação de políticas (LUPU *et al.*, 2000);
- Criação de técnicas e algoritmos para detecção e resolução de conflitos intra e inter políticas em sistemas PBNM (DUNLOP, 2002);
- Uso de técnicas como Inteligência Artificial (IA), Fuzzy, dentre outras, para o suporte a tomada de decisões em sistemas PBNM [N].

### 3.2.1 Arquitetura de referência para sistemas PBNM

A tese de doutorado de Magrath (2006) apresenta uma arquitetura de referência para sistemas PBNM, ilustrada na Figura 8, a qual pode ser dividida em quatro subsistemas principais: gerenciamento de eventos, inferência de políticas ou motor de políticas, executor e OAM. Adicionalmente são requeridas as políticas do tipo *Event-Condition-Action* (ECA) e um compilador que traduz as políticas em código executável pelo motor de políticas.

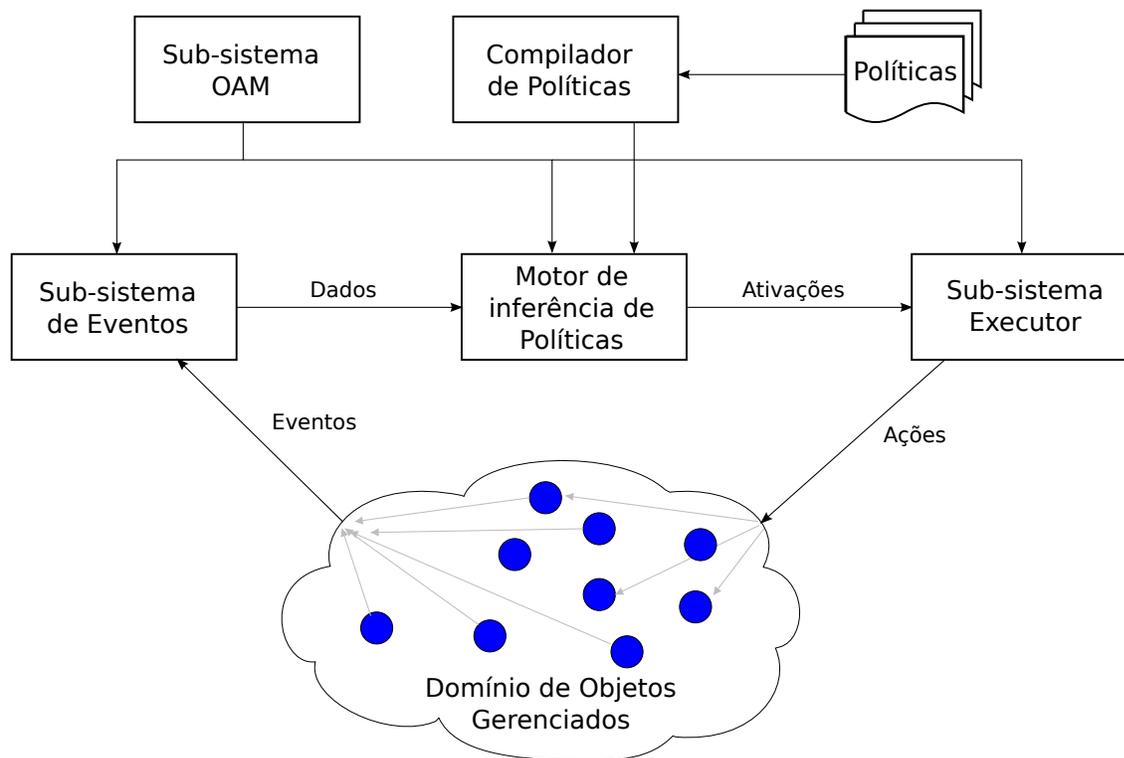


Figura 8 – Arquitetura de referência para sistema PBNM

Uma das contribuições da tese é a utilização dos algoritmos de análise de padrões, incluindo RETE (FORGY, 1982) e MATCHBOX (PERLIN; DEBAUD, 1989) para inferência de políticas, realizando análise de desempenho. Uma das características destes algoritmos é o mapeamento da política em grafos. Como exemplo, o algoritmo RETE utiliza o conceito de comparação de conjuntos de dados (fatos) com regras através de um motor de regras. O mecanismo é mapeado em um grafo direcional acíclico que permite análises parciais, junções, elementos de memória implementados como vértices do grafo e vértices de resolução de conflitos, permitindo *matches* do tipo *many-many*.

### 3.2.2 Linguagens para representação de políticas

Várias ferramentas foram propostas para a realização de gerenciamento baseado em políticas. No entanto, não há consenso em uma linguagem comum que ofereça uma solução unificada para suportar os diferentes conceitos de modelos de políticas. Segundo citado por Magrath (2006) foram identificados os seguintes requerimentos para uma linguagem de políticas:

- Suporte à políticas de segurança para controle de acesso e delegação para a transferência temporária de direitos de acesso;

- Técnicas de estruturação, como coleção ou agrupamento de objetos, para facilitar a especificação de políticas relacionadas a grandes sistemas com grande quantidade de objetos;
- Políticas de gerenciamento e de segurança compostas, que podem ser relacionadas a um papel específico ou uma posição dentro de uma organização, possibilitando o agrupamento de aplicações;
- Possibilidade de se detectar e avaliar conflitos e inconsistências de políticas;
- Possibilidade de extensão para acomodar os novos tipos de políticas. Esta extensão pode facilmente ser alcançada através do uso de heranças e de uma linguagem orientada a objetos;
- A linguagem deve ser compreensível e de fácil uso para seus usuários.

Na tese de Magrath (2006) é feita uma comparação e a análise de várias linguagens propostas para a representação de políticas em redes, incluindo a linguagem proposta pelo IETF, a *Policy Framework Definition Language* (PFDL). Os principais parâmetros utilizados na comparação entre políticas são: o suporte à detecção automática de conflitos nas políticas e facilidade de representação de políticas de alto nível. Algumas propostas de linguagens de políticas são descritas a seguir.

1. **PFDL** (STRASSNER; SCHLEIMER, 1998) é uma linguagem que permite a representação de políticas de redes em ambientes heterogêneos, viabilizando o mapeamento de especificações no nível da empresa para parâmetros nos dispositivos de rede.

Na PFDL, uma política é definida formalmente como uma agregação de regras. Cada regra é formada por um conjunto de condições e um conjunto de ações. A forma geral de representação de uma regra, suas condições e ações é mostrada a seguir:

```
se <condição1> e <condição2> ... e <condiçãoN>  
então <ação1> e <ação2> ... e <açãoM>
```

Eventualmente podem ocorrer conflitos entre ações a serem executadas na rede. Se um conflito ocorre entre ações da mesma política, esse é definido como conflito intra-política; caso contrário, esse é um conflito inter-políticas. O primeiro pode ser resolvido com ordenação na execução das ações; já o segundo pode ser resolvido pelo estabelecimento de prioridades entre políticas diferentes.

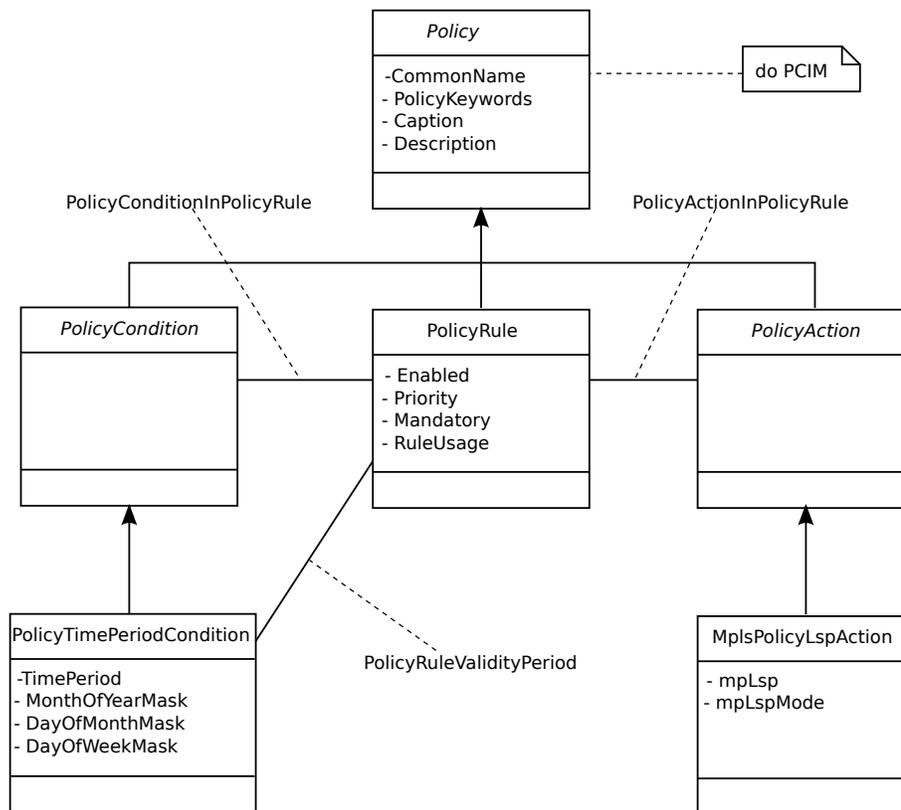


Figura 9 – Modelo PCIM

A PFDL é baseada no *Common Information Model* (CIM). Este modelo define uma hierarquia de classes que são usadas na representação de políticas. A PFDL foi adotada pela arquitetura de políticas do IETF, o *Policy Core Information Model* (PCIM). O modelo de classes para a representação de políticas básicas é mostrado na Figura 9.

Como é baseada no CIM, esta estrutura de políticas orientada a objetos tem mapeamento direto para um diretório da família X.500 acessível via *Lightweight Directory Access Protocol* (LDAP).

2. **Ponder** (DAMIANOU, 2002) é uma linguagem declarativa e orientada a objetos para especificar diferentes tipos de políticas, para agrupar políticas em papéis e relacionamentos, e então, definir as configurações destes papéis e relacionamentos como uma estrutura de gerenciamento. Ponder define quatro tipos básicos de políticas: autorização, restrição, obrigação e delegação.

O formato das **políticas de autorização** representadas pela linguagem Ponder é mostrado a seguir:

```
inst (auth+|auth-)[pathname]policyName{
```

```

subject [<type>]      domain-scope-expression;
target [<type>]       domain-scope-expression;
actionList            actionlist;
[when                 constraint-expression;]
}

```

Restrições são representadas através de um subconjunto da linguagem *Object Constraint Language* (OCL). Exemplo de uma política de autorização:

```

auth+ /usr/policies/allocateBandwidth {
  subject /usr/policies/groups/executiveGroup;
  target  /usr/realms/site1 OR /usr/realms/site2;
  action  enable(videoconference, bandwidth=1M, priority=5);
  when    time.between(1400, 1700); }

```

Este exemplo mostra a autorização de videoconferência para os usuários pertencentes ao grupo *executiveGroup*. A ação da política é a habilitação de videoconferência com uma banda alocada de 1Mbps e prioridade 5 entre as 14:00 e 17:00 h.

### 3.3 Redes definidas por software (SDN)

SDN é um paradigma emergente na área de redes de computadores que dá uma nova perspectiva de mudanças no sentido de superar as limitações atuais da infraestrutura de redes (KREUTZ *et al.*, 2014). Em primeiro lugar, quebra a integração verticalizada, separando a lógica de controle do plano de encaminhamento de roteadores e comutadores. Em segundo lugar, com a separação dos planos de controle e de dados, comutadores tornam-se simples dispositivos de encaminhamento, já que a complexa lógica de controle pode ser realizada por um controlador ou sistema operacional de rede logicamente centralizado, simplificando a avaliação e aplicação de políticas, reconfiguração e evolução das redes. Desta forma, a rede torna-se programável, através de aplicações SDN.

Uma visão simplificada da arquitetura é ilustrada na Figura 10. É importante enfatizar que um modelo de programação centralizado não incorre na centralização do controlador do ponto de vista físico. De forma a garantir níveis adequados de desempenho, escalabilidade e confiabilidade, sistemas SDN de produção deverão ser distribuídos fisicamente e eventualmente geograficamente.

O protocolo OpenFlow (MCKEOWN *et al.*, 2008) foi concebido na Universidade de Stanford com o objetivo de permitir que dispositivos de rede possam implementar somente o

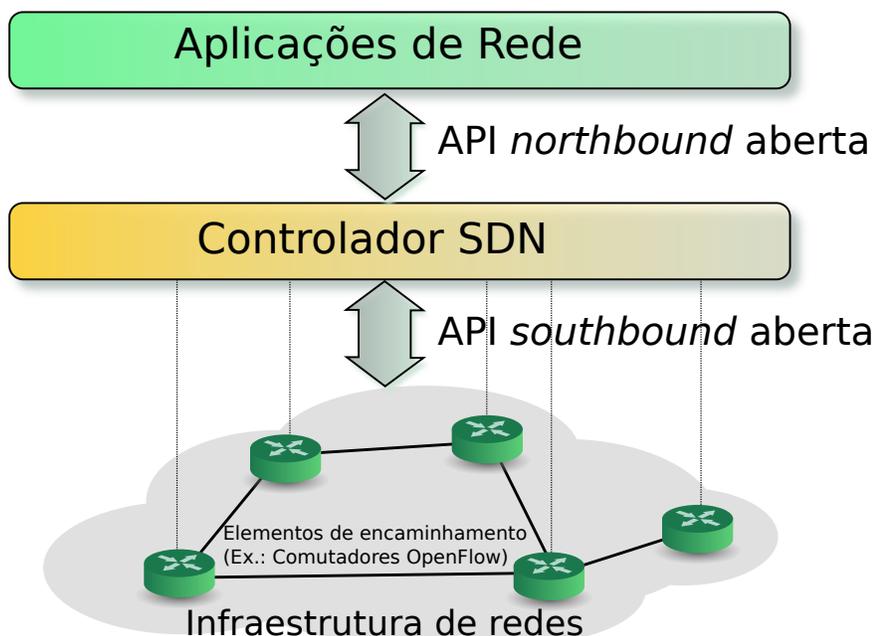


Figura 10 – Uma visão simplificada da arquitetura SDN

plano de dados, recebendo comandos de um controlador logicamente centralizado que implementa o plano de controle da rede como um todo. Os principais motivadores de tal proposta foram: (1) os altos custos de roteadores e comutadores comerciais, principalmente devido ao custo do software embarcado responsável pelo plano de controle; (2) o baixo custo de hardware de comutadores Ethernet; e (3) a possibilidade de utilização de alto poder computacional (Ex.: via virtualização e computação elástica) para a implementação do plano de controle através do controlador logicamente centralizado.

A arquitetura OpenFlow é dividida em duas partes: o protocolo OpenFlow, utilizado para a configuração de tabelas de fluxos e o *of-config* utilizado para gerência de configurações não relacionadas às tabelas de encaminhamento, como configurações básicas, de portas e mecanismos de *High Availability* (HA). Na prática, o *of-config* é a definição de um modelo YANG, utilizando NETCONF como protocolo base.

Em uma entrada OpenFlow quaisquer campos dos cabeçalhos de camadas 2 e 3 podem ser utilizados para classificação e podem ser modificados, conforme ilustrado na Figura 11. Desta forma, podem ser construídas regras complexas e flexíveis para o encaminhamento de pacotes, incluindo listas de controle de acesso. A partir da versão 1.1 do OpenFlow são suportadas funcionalidades como o suporte a múltiplas tabelas, passagem de metadados e execução de ações predefinidas, permitindo por exemplo a criação de *pipelines* lógicos de processamento de fluxos nos comutadores.

A arquitetura OpenFlow prevê os paradigmas proativo, reativo e misto para a configu-

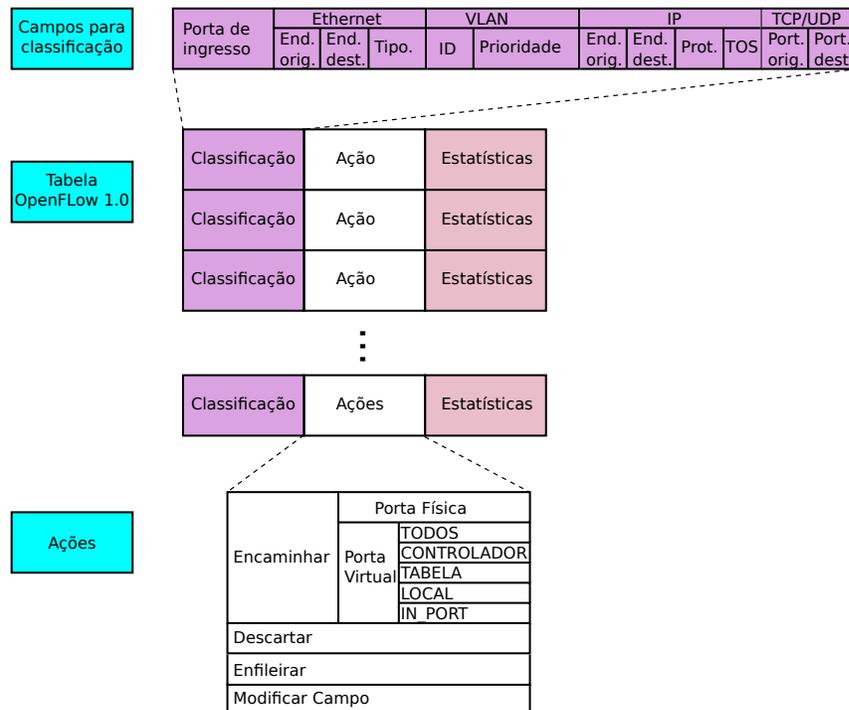


Figura 11 – Principais primitivas do OpenFlow 1.0

ração de fluxos. No modo proativo o controlador SDN pré-configura os fluxos nos comutadores. No entanto, se determinado pacote não é classificado por nenhum dos fluxos configurados, este pode ser descartado ou tratado de forma reativa através da opção *PACKET\_IN*, na qual o pacote é enviado ao controlador que pode criar o novo fluxo sob demanda.

Algumas das extensões da versão 1.5 do protocolo OpenFlow, geradas pelo grupo de trabalho de redes de transporte da *Open Networking Foundation* (ONF), tem foco na configuração de circuitos em redes ópticas de transporte. Os primeiros esforços de tal grupo caminham na direção da definição de modelos de abstração das redes ópticas, criando modelos abstratos padronizados de comutadores formados por um equipamento ou por uma rede, de forma que este comutador virtual possa ser configurado através do protocolo OpenFlow.

### 3.3.1 Controladores SDN

Pode-se considerar que o primeiro controlador SDN com suporte a OpenFlow é o NOX<sup>1</sup>. O mesmo foi desenvolvido pela empresa *Nicira Networks* e disponibilizado para a comunidade, com código aberto, em 2008. O NOX é implementado em C++ e inclui aplicações SDN de exemplo como descoberta de topologia, visão da rede como um único comutador, entre outras.

<sup>1</sup> <http://www.noxrepo.org/nox/about-nox/>

Atualmente estão disponíveis vários controladores com código aberto, como por exemplo o Floodlight<sup>2</sup>, ONOS<sup>3</sup> e OpenDayLight, bem como produtos comerciais que incluem funções de controladores SDN, como vCloud/vSphere da VMware, *Nicira Network Virtualization Platform* (NVP) e BNC<sup>4</sup> da Big Switch Networks.

A Figura 12 ilustra uma arquitetura idealizada de um controlador SDN, introduzida por Nadeau e Gray (2013). Observa-se que, na camada inferior, o controlador é formado por bibliotecas e adaptadores para a comunicação com elementos de rede através de protocolos padronizados como OpenFlow, PCEP, NETCONF, entre outros. Na parte central, o controlador é formado por uma camada de abstração, bem como um ambiente de execução das aplicações SDN, formado por APIs, ou *frameworks* para comunicação em sistemas distribuídos como REST e *Remote Procedure Call* (RPC). Já na parte superior, observam-se três grupos de funções: gerenciamento de serviços, orquestração da rede e serviços de rede.

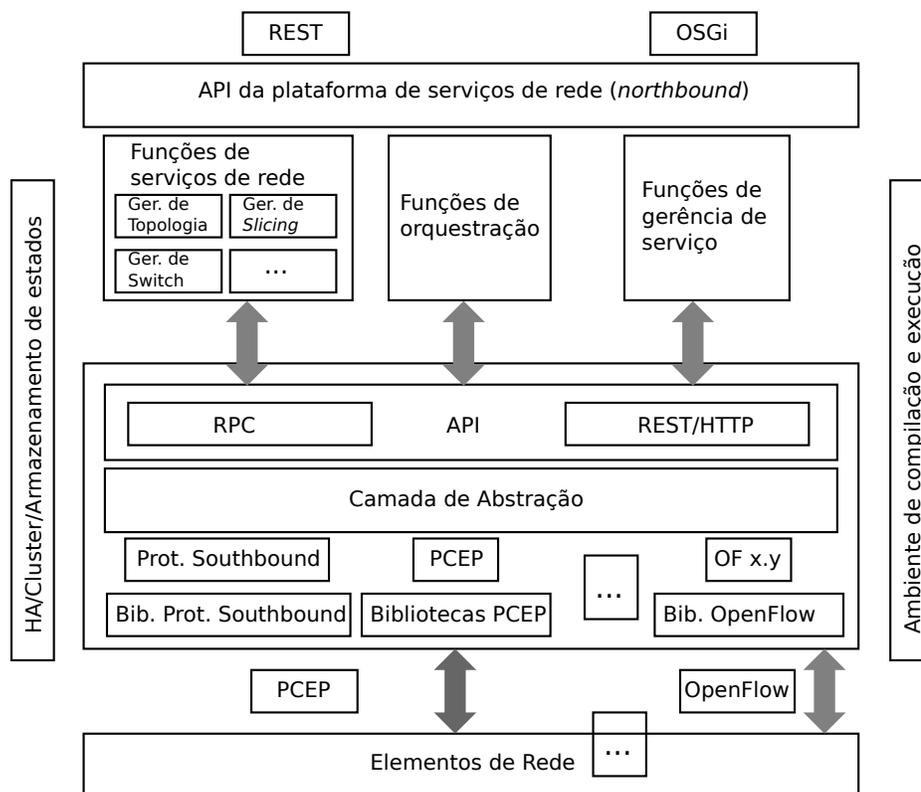


Figura 12 – Arquitetura de referência para controlador SDN, adaptado de Nadeau e Gray (2013)

<sup>2</sup> <http://www.projectfloodlight.org/floodlight/>

<sup>3</sup> <http://onosproject.org/>

<sup>4</sup> <http://www.bigswitch.com/products/SDN-Controller>

### 3.3.1.1 OpenDayLight

O OpenDayLight<sup>5</sup> é um projeto de uma plataforma SDN que possui como núcleo um controlador modular, plugável e aberto dos pontos de vista de arquitetura, código fonte, modelo de desenvolvimento e de governança. O controlador é implementado em software, utilizando linguagem Java, podendo ser contido em uma *Java Virtual Machine* (JVM), de forma que pode rodar em qualquer Sistema Operacional (SO) que suporte Java. O controlador expõe uma API *northbound* para a implementação de aplicações SDN, baseada no framework *Open Service Gateway initiative* (OSGi) para aplicações que rodam no mesmo espaço de memória que o controlador ou REST para aplicações localizadas por exemplo em outras máquinas. O OpenDaylight especifica que a lógica de controle e algoritmos residam nas aplicações SDN, que utilizam a API do controlador para obter informações e realizar configurações sobre a rede. Adicionalmente, o controlador implementa um conjunto de módulos plugáveis que podem ser utilizados pelas aplicações para a realização de tarefas comuns, como coleta de estatísticas, descoberta e abstração de topologia. O controlador possui uma camada de abstração denominada *Service Abstraction Layer* (SAL) que permite a configuração de serviços em um formato independente do protocolo de configuração utilizado na interface *southbound*. Desta forma, a interface *southbound* é capaz de suportar múltiplos protocolos através de plugins para protocolos como OpenFlow, BGP-LS (GREDLER *et al.*, 2014), PCEP, entre outros.

A Figura 13 ilustra a arquitetura geral do OpenDaylight.

### 3.3.1.2 Projeto TeraStream

Terastream é a primeira iniciativa SDN da operadora *Deutsche Telekom* (CLAUBERG, 2013), o qual é baseado nos seguintes princípios (ZALUSKI *et al.*, 2013):

- Programar serviços ao invés de re-arquitetar a rede e sistemas de gerência sempre que for necessário implementar um novo serviço.
- Redução na quantidade de tecnologias, utilizando somente IP e transmissão óptica. A rede não utiliza comutação OTN, L2 ou MPLS. Desta forma, as camadas IP e óptica devem ser integradas, evitando mecanismos redundantes de tratamento de falha, controle e operação, reduzindo custos.
- Minimizar interfaces internas através de ações como distribuir conexões de *peering*, realizar *offload* de tráfego externo e conexão de *data centers* diretamente aos roteadores de núcleo.

---

<sup>5</sup> <http://www.opendaylight.org/>

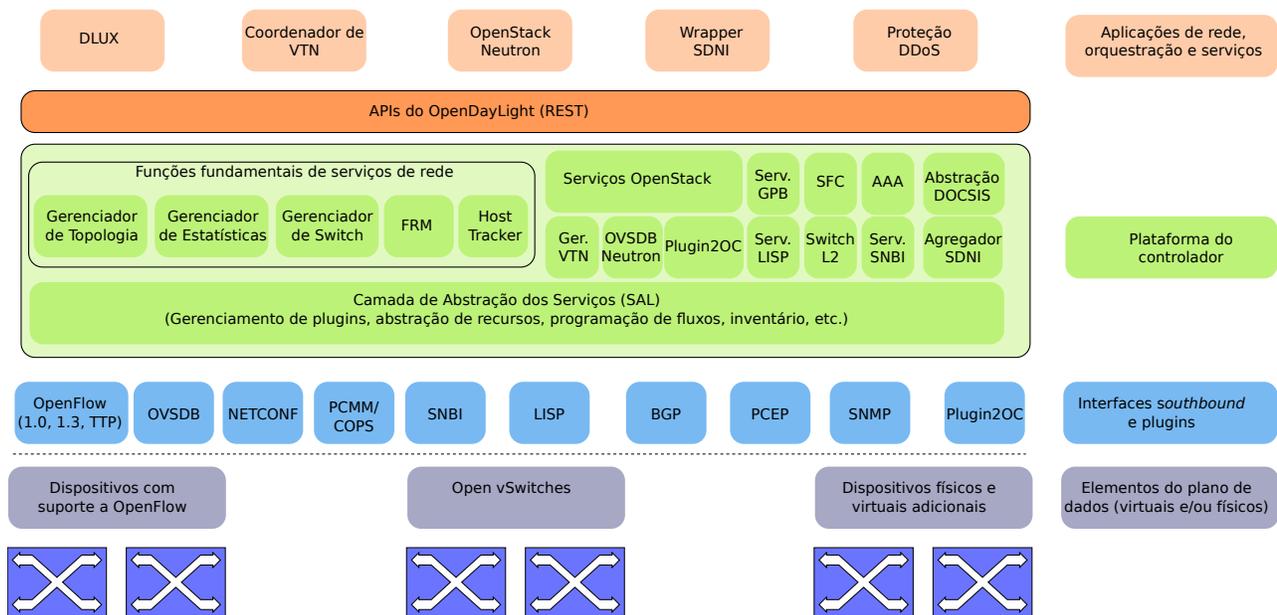


Figura 13 – Arquitetura do projeto OpenDaylight, versão Helium (OPENDAYLIGHT, 2015)

- Modelo de infraestrutura em nuvem, implementando o conceito de NFV.

Do ponto de vista da camada óptica, o projeto utiliza um esquema de transmissão DWDM estático do tipo *drop and waste*, no qual o sinal passa por um *splitter* óptico em cada site, de forma que todos os canais são derivados e encaminhados. A seleção de canal é feita pelo receptor coerente do roteador.

A Figura 14 apresenta uma visão geral da arquitetura SDN proposta para o projeto TeraStream, na qual as aplicações de gerenciamento interagem com uma entidade logicamente centralizada que realiza a abstração dos serviços de rede, podendo utilizar protocolos padronizados (Ex.: OpenFlow e NETCONF) para a interação com os elementos de rede.

Conforme detalhado por Lothberg (2013), a arquitetura de controle baseada em SDN utiliza o conceito de um *Operations Support System* (OSS) de tempo real, implementado utilizando-se como base a ferramenta *Network Control System* (NCS) desenvolvida pela empresa Tail<sup>6</sup>. O NCS permite que os serviços sejam definidos em modelos de dados (via linguagem YANG) ao invés de serem codificados como casos de uso *hard coded* no software. Os modelos são utilizados para funções como geração automática das interfaces de usuário, APIs *northbound*, modelos de base de dados, sequências de comandos e chamadas *southbound*. Os modelos de serviços em YANG são convertidos em sequências de comandos que podem ser enviados ao dispositivo de rede conforme a linguagem do modelo e fabricante (Ex.: co-

<sup>6</sup> <http://www.tail-f.com/>

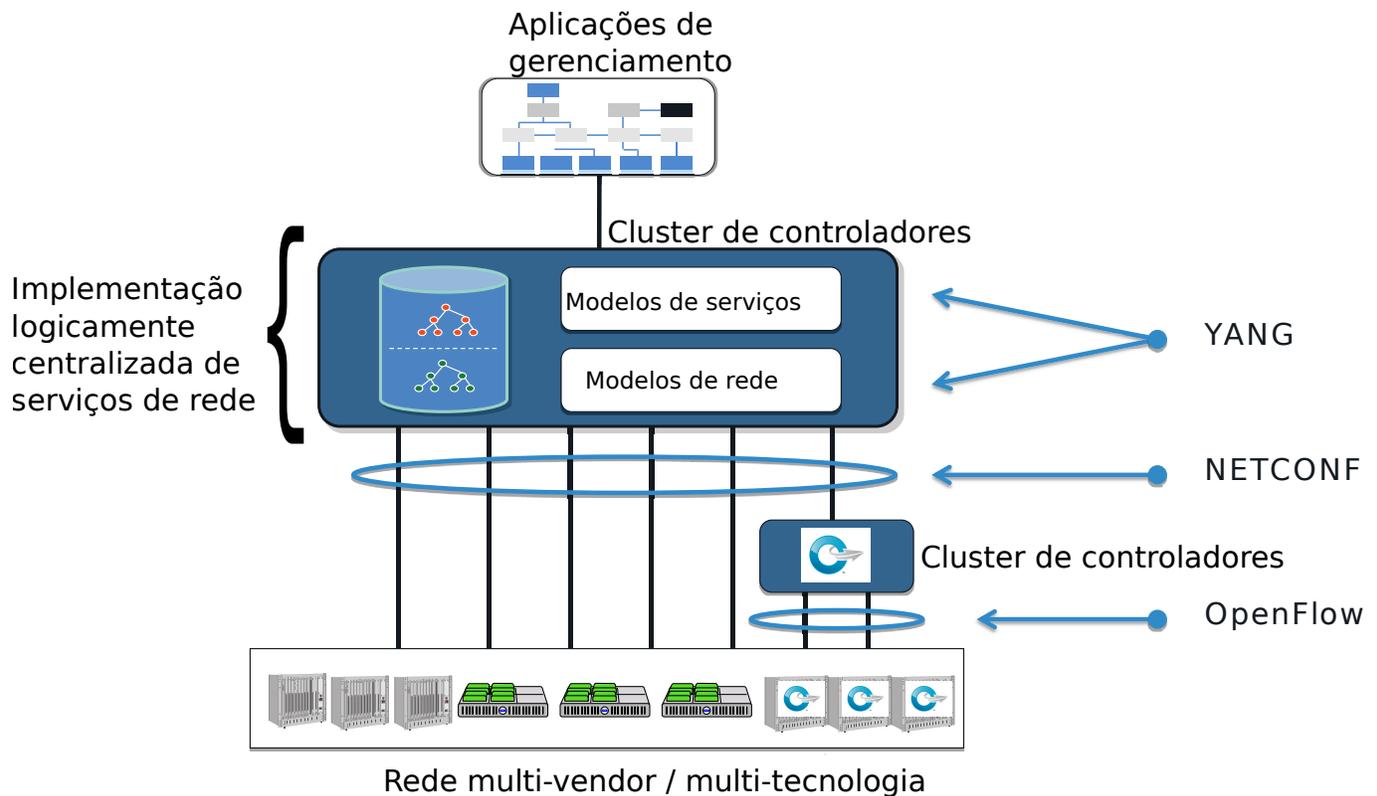


Figura 14 – Arquitetura SDN do projeto TeraStream

mandos CLI específicos de fabricante, comandos NETCONF, *Simple Network Management Protocol* (SNMP), REST, etc) (TAIL-F, 2014).

### 3.3.2 NETCONF como protocolo em redes SDN

Conforme documentado na RFC 3535 (SCHOENWAEELDER, 2003), entre os anos de 2001 e 2002, várias operadoras de telecomunicações manifestaram que as soluções e protocolos para o gerenciamento de redes, na época, não respondiam a seus requerimentos, especialmente no que diz respeito à gerência de configuração. Desta forma, em junho de 2002, a *Internet Architecture Board* (IAB) realizou um *workshop* sobre gerenciamento de redes, no qual concluiu-se que era necessário haver funcionalidades como suporte a transações, *rollback*, capacidade de salvar e restaurar configurações de equipamentos, entre outras (SHAFER, 2011).

Com base nestas conclusões, o IETF formou o grupo de trabalho NETCONF, o qual definiu e especificou o protocolo NETCONF (ENNS *et al.*, 2011). Este protocolo define um mecanismo simples, no qual aplicações de gerência de redes (que atuam como clientes) podem invocar ações de configuração nos elementos de rede (que agem como servidores). Já em

2008, o grupo de trabalho NETMOD produziu as especificações da linguagem YANG (BJORKLUND, 2010), que define modelos de dados para o protocolo NETCONF.

O Protocolo NETCONF define um mecanismo que permite a configuração de elementos de rede, permitindo que configurações sejam obtidas, manipuladas e enviadas aos equipamentos de forma parcial ou total, através de conexões persistentes e seguras, com suporte a transporte através do protocolo *Secure Shell* (SSH). NETCONF utiliza um paradigma baseado em RPC. O cliente codifica uma chamada RPC em *Extensible Markup Language* (XML) e a envia a um servidor utilizando uma sessão orientada à conexão. Por sua vez, o servidor responde com uma mensagem também codificada em XML. Os formatos tanto a requisição como da resposta são descritos em *XML Document Type Definition* (DTD) ou *XML schemas*, de forma que ambas as partes possam reconhecer o formato e as restrições impostas na troca de mensagens.

NETCONF pode ser usado em conjunto com ferramentas de transformação XML, como *XML Stylesheet Language for Transformations* (XSLT), com o objetivo de prover a geração automatizada de configurações. As informações sobre topologias de rede, enlaces, políticas e serviços, podem ser obtidas em formatos independente de fabricante e transformadas através de *scripts* XSLT para formatos específicos de fabricante, sistema operacional, versão de software, entre outros. O resultado pode ser enviado para o dispositivo através do protocolo NETCONF.

A especificação do NETCONF define a separação de informações de configuração e de estado, permitindo que tais informações sejam tratadas de forma diferenciada. Como exemplo, a realização de configurações deve estar associada a transações, enquanto as informações operacionais, ou de estado, não.

O modelo de comunicação do NETCONF utiliza mensagens dos tipos `<rpc>` e `<rpc-reply>`. O elemento `<rpc>` codifica uma requisição que é enviada do cliente para o servidor. Já a mensagem `<rpc-reply>` é enviada em resposta à uma mensagem `<rpc>`, a qual sempre deve possuir um atributo *“message-id”* idêntico ao mesmo atributo da mensagem `<rpc>`. Outro tipo de mensagem é o `<rpc-error>`, que é enviada em caso de erro de processamento de uma requisição.

NETCONF define a existência de uma ou mais bases de dados de configuração, permitindo a realização de operações nestas bases. São especificados três bases de configuração: a configuração corrente (*running configuration*), a configuração candidata e a de inicialização, a qual é utilizada quando o equipamento é inicializado.

O protocolo prevê um conjunto reduzido de operações para o gerenciamento de configurações e obtenção de informações de estado dos equipamentos. O protocolo base inclui

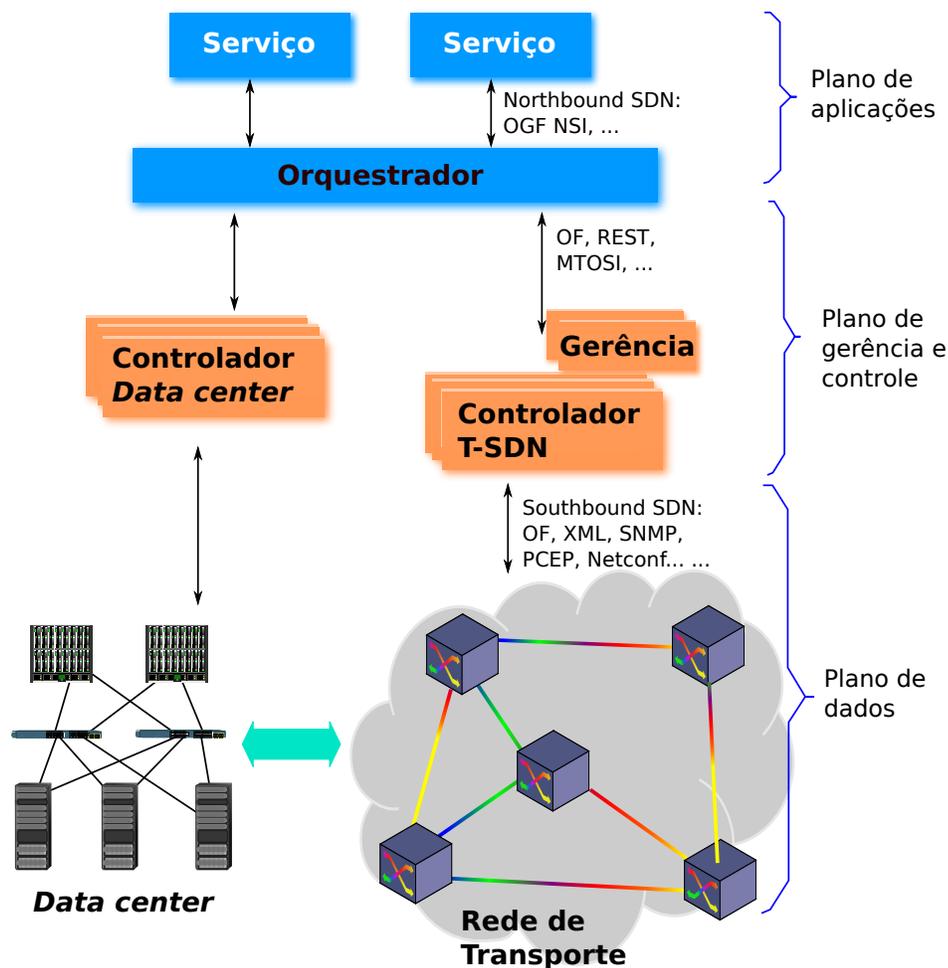


Figura 15 – Arquitetura SDN para redes de Transporte, OIF

as seguintes operações: *get*, *get-config*, *edit-config*, *copy-config*, *delete-config*, *lock*, *unlock*, *close-session* e *kill-session*.

### 3.3.3 Atividades relacionadas à T-SDN no OIF

O grupo de trabalho *carrier* do OIF definiu um conjunto de requisitos de alto nível (GERLACH, 2013) para a implementação de arquiteturas SDN para redes de transporte. A arquitetura de referência definida é ilustrada na Figura 15.

A seguir é apresentado um resumo dos requisitos apresentados pelo OIF:

- **Comunicação entre componentes SDN:** deve haver conectividade entre os componentes, com suporte a resiliência, escalabilidade, desempenho e segurança, incluindo mecanismos como controle de congestionamento dos pacotes de controle, mecanismos de segurança, mecanismos de OAM e de garantia de desempenho.

- **O orquestrador** deve realizar as seguintes funções: coordenar ações entre os *data centers* e as redes de transporte, incluindo tratamento de alarmes e requisições de serviço; suportar modelos de virtualização (da rede de transporte e *data centers*); suportar a aplicação de políticas; suportar mecanismos de balanceamento de carga nas redes de transporte e *data centers*; mecanismos de escalonamento e agendamento de serviços; suportar tratamento de requisitos de *Service Level Agreement* (SLA); além de mecanismos de proteção e restauração.
- **Plano de controle (controlador SDN)**: deve suportar gerenciamento de conexões, mecanismos de descoberta, confiabilidade (Ex.: restauração), funções de abstração, provendo interfaces *northbound* ao orquestrador e *southbound* para os elementos de rede. O plano de controle deve operar em um ambiente que utiliza NFV, devendo suportar redes virtuais em um contexto multicamadas.
- **Plano de gerência**: deve suportar as interfaces apresentadas na Figura 15, suportando FCAPS em um ambiente de redes virtuais em um contexto multi-camadas.
- **Plano de dados**: suporta funções como encapsulamento, cross-conexões físicas e lógicas, encaminhamento, transmissão e mecanismos de proteção, aplicando regras e políticas definidas nos planos de gerência e controle. Adicionalmente, deve prover interfaces *northbound* com os planos de controle e gerência, suportando provisionamento e coleta de informações. O plano de dados deve suportar o particionamento de recursos (*network slicing*), viabilizando um ambiente de redes virtuais em um contexto multi-camadas.

### 3.3.3.1 Arquitetura de políticas em SDN com mecanismo declarativo

A arquitetura definida pelo OpFlex (SMITH *et al.*, 2014) utiliza um modelo declarativo de gerência, no qual as políticas são definidas em um repositório centralizado, o *Policy Repository* (PR), e implementadas de forma distribuída através dos *Policy Elements* (PEs). O protocolo OpFlex é utilizado na comunicação entre PRs e PEs, permitindo a comunicação bidirecional de políticas, eventos, estatísticas e falhas.

O principal argumento para a utilização do modelo declarativo de gerência, em contrapartida ao modelo imperativo é que o primeiro oferece maior potencial de escalabilidade em ambientes heterogêneos e com alto nível de dinamicidade, já que os sistemas declarativos operam de forma que elementos distribuídos possam ter informações locais mais detalhadas, tomando decisões locais para levar ao sistema para o estado desejado (definido na política). Este comportamento pode ser comparado ao de um controlador de voo que informa ao piloto do avião as informações sobre a rota a ser seguida, confiando que o piloto realizará as

ações necessárias para seguir as ordens. No entanto, o controlador não tem informações locais ao avião como velocidade do vento, inclinação, peso, entre outras, tratadas pelo piloto e pelo sistema computacional do avião que levarão este para o estado definido pela política (rota/altitude a serem seguidas).

No modelo proposto, cada domínio administrativo possui um PR que é visto como uma entidade lógica que concentra todas as políticas do domínio. O PR trata as requisições de resolução de políticas dos PEs. As políticas, que representam uma declaração do estado desejado, são configuradas diretamente pelo administrador ou indiretamente pelas aplicações (através de traduções de políticas de mais alto nível).

Todos os componentes lógicos e físicos do domínio administrativo são modelados como uma *Management Information Tree* (MIT). Iniciando na raiz da estrutura hierárquica, todas as políticas do sistema podem ser alcançadas através das relações de contenção dos tipos pai-filho. Cada nó na árvore representa um *Managed Object* (MO), ou um grupo destes, contendo estados administrativo e operacional. Um MO pode representar um objeto concreto (e.g., um comutador ou uma interface) ou um objeto lógico, como uma política ou falha.

O protocolo OpFlex é utilizado pelos elementos (PEs e PR) para troca de informações sobre políticas e dados operacionais. O protocolo usa *JavaScript Object Notation* (JSON), XML, ou *OpFlex-Binary-RPC* como encapsulamento.

Os PEs residem nos nós físicos ou virtuais que são sujeitos a controle por políticas em um determinado domínio administrativo. Os PEs recebem gatilhos de políticas (*policy triggers*) através de gatilhos locais ou invocados por outros PEs. Gatilhos locais envolvem transições de estados de MOs. Já os gatilhos invocados por outros PEs são transmitidos usando o protocolo OpFlex. Ambos os tipos de gatilhos resultam em resolução de políticas, através de requisições ao PR, também via OpFlex.

As políticas são representadas como sub-árvores de MOs. Na resolução da política, o PE “renderiza” esta para o subsistema envolvido, realizando um monitoramento contínuo do estado desejado, podendo realizar ações corretivas locais, se necessário, para garantir a manutenção da aplicação das políticas de seu escopo.

## 3.4 Virtualização de redes

O conceito de virtualização de redes surgiu como uma proposta de extensão da noção de virtualização de recursos computacionais para o de virtualização de redes, podendo este ser definido como “uma técnica para isolamento de recursos computacionais e de rede através de virtualização para a alocação destes à redes lógicas (virtuais), permitindo, desta

forma, a instanciação e acomodação de múltiplas redes virtuais que suportem programação e configuração de forma similar às redes não virtualizadas” (NAKAO, 2010). Diferentemente de *Virtual Private Networks* (VPNs) que proveem conectividade entre membros e isolamento de outras VPNs através de uma rede compartilhada, a virtualização de redes objetiva disponibilizar características adicionais, como programabilidade (como as que permitem planos de controle independentes para diferentes redes virtuais), conhecimento da topologia, suporte a diferentes níveis de abstração da rede e recursos físicos, reconfiguração rápida, além de isolamento de recursos.

Um dos primeiros trabalhos relativamente completos sobre virtualização de redes foi publicado em 2010 por Chowdhury e Boutaba (2010), no qual é proposta uma arquitetura de referência para o ambiente de virtualização de redes, contendo modelos de negócios que incluem um novo ator, denominado provedor de infraestrutura. Desta forma, os provedores de serviço administram suas redes virtuais, porém o provedor de infraestrutura administra a rede física. Adicionalmente, são definidos objetivos de projeto e requisitos para as redes ópticas virtuais, como flexibilidade, gerenciabilidade, escalabilidade, isolamento, programabilidade e suporte ao legado. Finalmente, os autores citam vários projetos sobre virtualização de redes na época.

A virtualização de servidores em *data centers* tem imposto desafios para a infraestrutura de redes. A separação das redes em *Virtual Local Area Networks* (VLANs), utilizando abordagens tradicionais como *Spanning Tree Protocol* (STP) para gerenciamento de topologia (remoção de *loops*) não é mais adequada para os requisitos impostos pelos ambientes de *data centers*, devido a fatores como quantidade de máquinas físicas e virtuais, requisitos de conectividade, dinamicidade e distribuição dos *data centers* em diferentes localidades (virtualização de *data centers*). Neste contexto, a empresa VMware<sup>7</sup> propôs e tem implementado a tecnologia *Virtual eXtensible Local Area Network* (VXLAN) (MAHALINGAM *et al.*, 2013), que tem sido utilizada como uma forma de criar uma rede de camada 2 sobreposta sobre uma rede de camada 3, a qual suporta mecanismos mais avançados de roteamento (sem *loop*), escalabilidade na quantidade de *endpoints*, bem como mecanismos avançados de balanceamento de tráfego como *Equal Cost Multipath* (ECMP). O isolamento dos grupos de máquinas virtuais de interesse é feito através da criação de segmentos VXLAN, os quais são identificados através de identificadores de 24 bits denominados *VXLAN Network Identifier* (VNI). O encapsulamento é comumente feito pelo *Hypervisor*, através de uma entidade denominada *VXLAN Tunnel Endpoint* (VTEP). Desta forma, esta foi umas das primeiras soluções que permitem uma integração da camada de virtualização de servidores (*Hypervisor* de plataforma computacional) com a camada de virtualização de redes, através de *Hypervisor*

---

<sup>7</sup> <http://www.vmware.com/>

de rede (como por exemplo a solução *SDN for Virtual Environments* (SDN-VE) (IBM, 2013)).

Outra solução de virtualização de redes é o *FlowVisor* (SHERWOOD *et al.*, 2010) que implementa o conceito de *Hypervisor* para redes *OpenFlow* (MCKEOWN *et al.*, 2008), permitindo que os comutadores *OpenFlow* tenham a visão do *Hypervisor* como um controlador único e que cada controlador tenha a visão de um segmento de rede (*slice*). *FlowVisor* implementa o conceito de delegação recursiva, permitindo instâncias de *FlowVisors* sejam cascadeadas, de forma que as instâncias superiores funcionam como controladores *OpenFlow* para as inferiores. Outra característica importante é a separação da camada de virtualização da camada de controle, ou seja, não é necessário incorporar primitivas de virtualização no protocolo *OpenFlow* nem nos controladores.

A família de linguagens denominada *Frenetic* foi proposta no contexto de virtualização de redes (GUTZ *et al.*, 2012), em que, o isolamento é garantido a nível da linguagem, ao invés de mecanismos tradicionais como VLANs, rótulos MPLS, *Firewalls*, ou *Hypervisor* de rede. Os autores argumentam que o isolamento através de construções da linguagem pode ocorrer nos níveis físico, de tráfego e de controle, possibilitando a aplicação de verificação formal. Cada *slice* é definido em termos de um grafo que representa uma versão restrita da topologia física da rede, de um mapeamento dos nós do grafo à rede física e de um conjunto de predicados que especificam quais pacotes podem entrar no perímetro do *slice*. Programadores especificam um programa separado para cada *slice*, sendo função do compilador compor as regras globais a serem instaladas em cada equipamento, com base em políticas, as quais deverão garantir o isolamento entre o *slices*.

Observa-se que SDN não é um pré-requisito para a implementação de virtualização de redes (a abstração de uma rede física na forma de uma rede lógica). Da mesma forma, a separação de um plano de controle logicamente centralizado do plano de dados, definida em SDN, não implica na virtualização da rede. No entanto, pode-se considerar que SDN e virtualização de redes possuem uma relação simbiótica em três formas principais (FEAMSTER *et al.*, 2014):

1. SDN é uma tecnologia que habilita a virtualização de redes. Em ambientes de computação em nuvem, a utilização de um controle logicamente centralizado permite a criação de redes *overlay* sem necessariamente haver a necessidade da criação de um *switch* virtual mapeado a cada entidade física. Desta forma, os *switches* virtuais podem ser reconfigurados de forma dinâmica quando máquinas virtuais movem para outras localidades. Um exemplo de controlador SDN que permite a virtualização de redes em ambientes de computação em nuvem é o VMware NSX (VMWARE, 2015).
2. Virtualização de redes tem sido utilizada para a avaliação e testes de arquiteturas,

controladores e aplicações SDN, antes que estas sejam implementadas nas redes operacionais.

3. A virtualização (*slicing*) de recursos em equipamentos em uma SDN é mais simples de ser realizada que em equipamentos que possuem plano de controle. Em equipamentos com plano de controle simplificado, a separação dos recursos é feita no nível do controlador SDN, utilizando tecnologias de separação de tráfego no nível do plano de dados para isolamento de recursos entre as redes virtuais. Já nos equipamentos tradicionais, é necessário que cada equipamento tenha múltiplas instâncias de plano de controle (uma para cada rede virtual), ou é necessário estender os protocolos para que suportem a segregação dos recursos, como é feito em VPNs MPLS.

Vale ressaltar que alguns benefícios comumente atribuídos a SDN como a otimização dos recursos físicos pela utilização de virtualização são, na verdade, provenientes da virtualização. No entanto, SDN facilita a virtualização de redes, através da criação de abstrações e separação dos planos de controle e de dados.

### 3.4.1 Virtualização de redes ópticas de transporte

A virtualização da rede óptica pode ser definida como uma composição de múltiplas *Virtual Optical Networks* (VON), coexistindo de forma simultânea sobre a mesma infraestrutura física. Uma VON é um conjunto de nós ópticos interconectados por enlaces virtuais. A virtualização de nós e enlaces pode ser obtida através do particionamento e/ou agregação de recursos físicos da rede óptica (HAMMAD *et al.*, 2013), através do mapeamento de quais recursos (Ex.: nós, enlaces, lambdas) devem ser designados para cada VON. Dentre as motivações para a virtualização de redes ópticas destacam-se o provimento de serviços do tipo NaaS (CIENA, 2013), aplicações para interligação de *data centers*, integração IP-óptica, bem como compartilhamento de infraestrutura de transporte entre diferentes provedores (SCHOENTGEN, 2012).

O autor Schmitt (2012) argumenta que OTN provê uma solução elegante para o transporte e gerenciamento de tráfego em uma rede óptica virtualizada, já que provê agregação (*grooming*) com suporte nativo a isolamento entre diferentes conexões. Com a adição de funcionalidades de comutação no nível de ODU, a implementação e operação destas redes ganha alto grau de dinamicidade, permitindo a implementação de novos serviços e cenários de rede. Outra proposta recente que permite o particionamento dos recursos de rede é a tecnologia *Optical Orthogonal Frequency Division Multiplexing* (O-OFDM) (HAMMAD *et al.*, 2013), já que permite que cada enlace virtual seja mapeado para uma sub-portadora em um enlace fim a fim.

Contudo, a implementação de novas tecnologias de redes ópticas e a necessidade de prover serviços dinâmicos, incluindo virtualização, impõe novos desafios para a operação de redes ópticas de transporte. Rotinas consideradas simples em redes tradicionais, como equalização, controle de ganho óptico, isolamento de canais e a camada de controle, incluindo GMPLS, devem ser revistas com o objetivo de adaptá-las a este novo cenário.

### 3.4.2 Virtualização de funções de redes

O conceito de NFV [R] foi introduzido com o objetivo de trazer para a área de redes o conceito de virtualização de infraestrutura já utilizado amplamente na área de computação. Desta forma, foi criado um grupo de especificações para a indústria ISG<sup>8</sup> dentro do ETSI, formado principalmente por operadoras, objetivando definir requisitos, arquitetura e *framework* para NFV. Potenciais exemplos para a aplicação de NFV identificados pelo ISG do ETSI (CHIOSI, 2012) incluem a virtualização de elementos de comutação, controle e sinalização, *gateways* de túneis, DPI, *Authentication, Authorization, and Accounting* (AAA), *Content Distribution Networks* (CDNs), *Web Proxy, Customer Premises Equipments* (CPEs), aceleradores de aplicações, funções de segurança como *firewalls* e *Intrusion Prevention System* (IPS).

A implementação de redes com funções virtualizadas tem os seguintes objetivos principais:

- Melhorar eficiência no uso de CAPEX comparando-se a implementações com hardware dedicado. Isto é alcançado utilizando hardware comercial de uso geral (Ex.: servidores e armazenamento) para prover funções de dispositivos de rede através de técnicas de virtualização. O compartilhamento de hardware para realização de diferentes funções, bem como a redução dos tipos de dispositivos utilizados na rede, podem auxiliar na redução tanto de CAPEX como de OPEX.
- Flexibilidade na designação das funções de rede, denominadas *Virtualised Network Functions* (VNFs), ao hardware. Isto permite obter escalabilidade, já que desacopla as funcionalidades de hardware e locais específicos, permitido que as funções (software) virtualizadas sejam posicionados de forma flexível nos locais mais apropriados, referidos como *NFV Infrastructure Point of Presences* (NFVI-PoPs) (ETSI Group Specification, 2013), dependendo das condições externas como demanda, falhas, horário do dia, política de compartilhamento de recursos, ciclo de liberação (Ex.: versões alfa, beta, produção). NFVI-PoPs podem incluir sites de clientes, Ponto de Troca de Tráfego (PTT), centrais de operadoras ou *data centers*.

<sup>8</sup> <http://portal.etsi.org/portal/server.pt/community/NFV/367>

- Rápida implementação de novos serviços, sem necessidade de alterações em plataformas de hardware.
- Melhora de eficiência nos processos operacionais através da automatização destes, resultando em uma drástica redução em OPEX.
- Redução na utilização de energia elétrica, contribuindo com as iniciativas “*green networks*”, já que permite a migração de cargas e desligamento de hardware não utilizado.
- Padronização e abertura de interfaces entre funções virtualizadas e a infraestrutura com as entidades de gerenciamento, de forma que os diferentes elementos possam ser oferecidos por diferentes fornecedores de forma desacoplada.

#### 3.4.2.1 Arquitetura de referência para NFV

NFV visa a implementação de *Network Functions* (NF) como entidades de software que rodam sobre a *NFV Infrastructure* (NFVI). Foi definida uma arquitetura de referência para a implementação de NFV pelo ETSI (ETSI Group Specification, 2013) que objetiva permitir a instanciação dinâmica de funções de rede virtuais (VNFs), bem como a relação entre estas.

Observa-se que pode existir relação de conectividade de rede entre diferentes VNFs, sendo esta definida como *VNF Forwarding Graph* (VNF-FG), os quais podem definir por exemplo a conectividade lógica entre VNFs relativas a *firewalls*, balanceadores de carga, *Network Address Translation* (NAT), servidores web, entre outros.

A Figura 16 ilustra a arquitetura de alto nível para virtualização de funções de redes. A Arquitetura é dividida em três blocos principais:

- Funções de rede virtualizadas (VNFs);
- Infraestrutura NFV (NFVI)
- Gerenciamento e orquestração NFV.

Os blocos funcionais da arquitetura são descritos a seguir:

- VNF é a virtualização e funções de redes. Exemplos de funções de rede incluem as desempenhadas por roteadores, firewalls, gateways residenciais, elementos da arquitetura *3rd Generation Partnership Project* (3GPP) como o *Mobility Management Entity* (MME) e o *Packet Data Network Gateway* (PGW), servidores de autenticação ou servidores *Dynamic Host Configuration Protocol* (DHCP). Uma função de rede pode

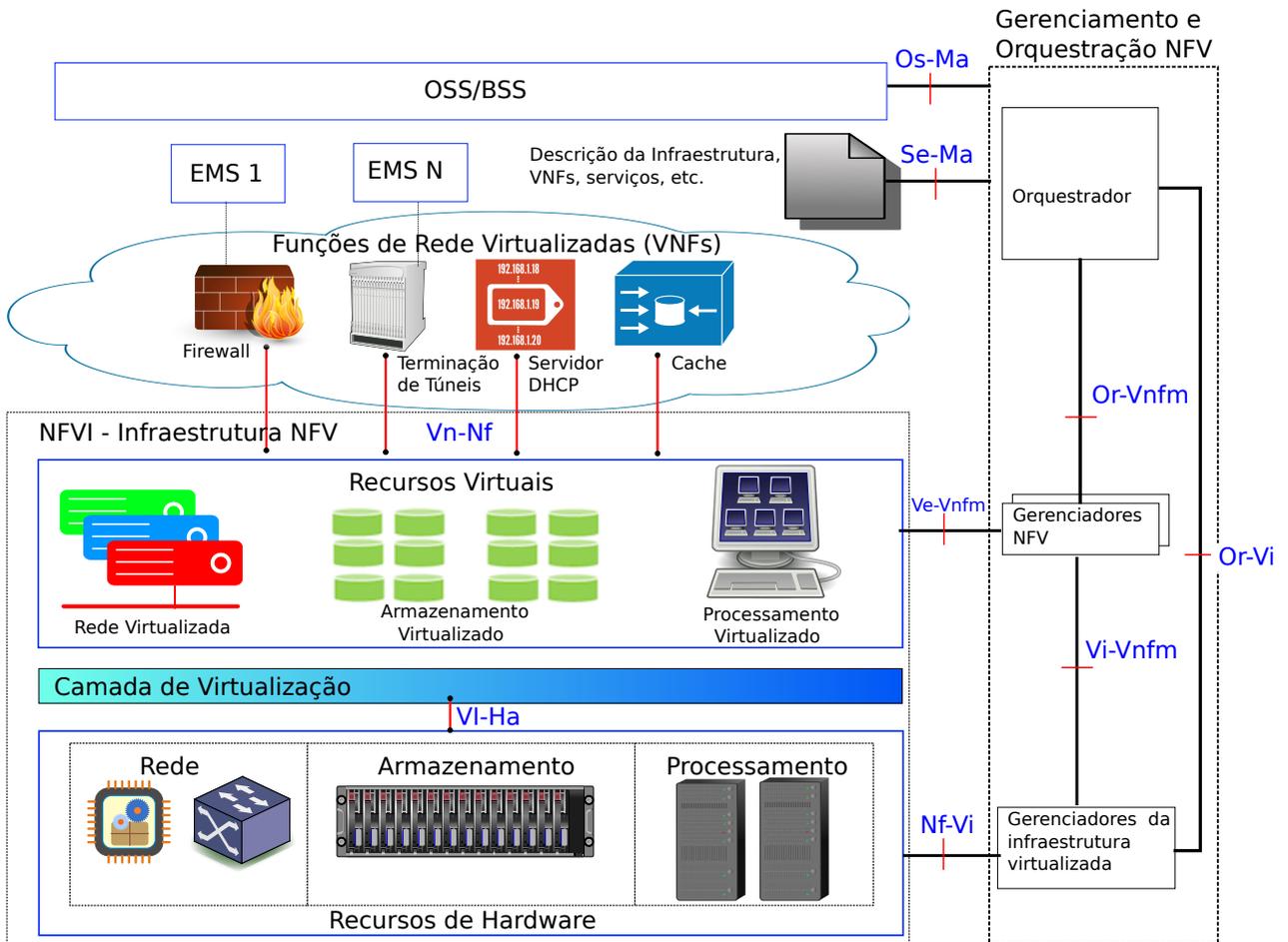


Figura 16 – Arquitetura de Referência para NFV. Fonte: [R].

ser decomposta em diferentes componentes externos, os quais podem ser implementados em diferentes máquinas virtuais. No entanto, o comportamento funcional de uma função de rede deve ser independente se tal função é virtualizada em múltiplas *Virtual Machines* (VMs), em uma única VM ou não é virtualizada.

- NFVI é a composição dos recursos de hardware e software necessários para a implementação, execução, e gerenciamento das VNF. A infraestrutura para o provimento de NFVI pode ser distribuída em diferentes localidades (ex.: diferentes NFVI-PoPs, de forma que a rede que provê conectividade entre os NFVI-PoPs faz parte da NFVI. No entanto, a camada de virtualização para os recursos de hardware visto pelas VNFs permite que a NFVI possa ser vista como uma entidade única.
- Gerenciamento e orquestração: os gerenciadores da infraestrutura virtualizada controlam a interação da VNF com os recursos físicos sob sua autoridade, realizando gerenci-

amento de recursos (Ex.: alocação, desalocação e inventário), além de operações como visibilidade da infraestrutura, coleta de informações para gerência de falhas e desempenho. Já os gerenciadores das VNFs são responsáveis pelo gerenciamento do ciclo de vida das VNFs, incluindo operações como instanciação, atualização e finalização. Outras entidades incluem o *Element Management System* (EMS), que provê funções típicas de gerenciamento para uma ou mais VNFs, e o orquestrador, responsável pelo gerenciamento dos serviços, orquestrando recursos de infraestrutura e de software para as VNFs. Finalmente, a base de configuração inclui informações de configuração dos serviços, das VNFs e da infraestrutura. Inclui templates para a implementação de VNFs, grafos de encaminhamento das VNFs, informações relativas aos serviços e à infraestrutura.

#### 3.4.2.2 Composição de serviços de redes

Um serviço de rede fim a fim (ex.: Acesso Internet banda larga, VPN, serviço web) pode ser descrito por um VNF-FG. Desta forma, o comportamento de um serviço de rede é a combinação do comportamento de seus blocos funcionais, que podem incluir funções de rede (NFs) individuais, conjuntos de NFs, grafos de encaminhamento de NFs, e/ou infraestrutura de rede.

A Figura 17 mostra um exemplo de um serviço de rede fim a fim e as diferentes camadas envolvidas no processo de virtualização deste. Os NFVI-PoPs possuem recursos de armazenamento, processamento e rede, bem como conectividade entre estes. O VNF-FG é composto por duas VNFs (VNF-1 e VNF-3) interconectadas por um VNF-FG aninhado (VNF-FG-2) que é composto por três VNFs (VNF-2A, 2B e 2C).

As funções de virtualização são instanciadas sobre a camada de virtualização, permitindo que a localização física e quais recursos físicos estão utilizados para o provimento das VNFs e do serviço sejam abstraídos.

### 3.4.3 Virtualização de funções de redes de transporte

O conceito de NFV tem sido explorado no contexto de redes de dados, abrangendo equipamentos que atuam entre as camadas dois e sete do modelo *Open Systems Interconnection* (OSI). No entanto, verifica-se que alguns trabalhos com aspectos similares a NFV vêm sendo propostos no contexto de redes ópticas de transporte. A proposta de arquitetura de elemento de rede óptica sob demanda (AMAYA *et al.*, 2011) contempla a adaptação dinâmica da arquitetura interna de um elemento de rede, com base nos requisitos da rede como tráfego, processamento e/ou tratamento do sinal óptico. Em tal proposta, é utilizado um OXC que funciona como um *backplane* óptico, no qual dispositivos como amplificadores,

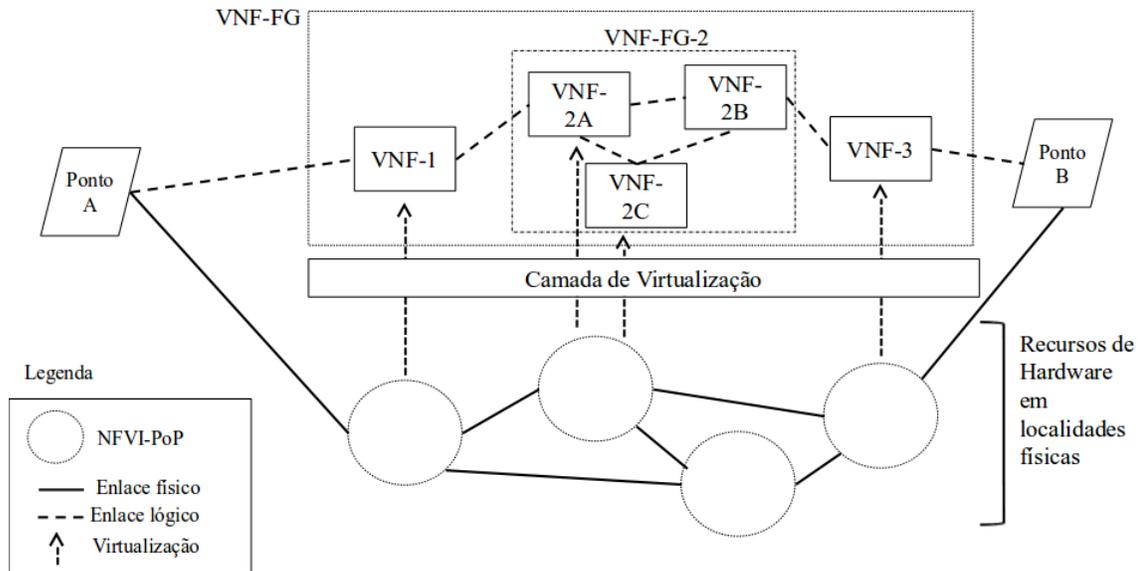


Figura 17 – Exemplo de um serviço de rede fim-a-fim com VNFs e VNF-FGs aninhados. Fonte: [R].

módulos WSS, multiplexadores/demultiplexadores, entre outros, são conectados, permitindo que a conectividade entre tais dispositivos seja organizada sob demanda. Esta proposta pode ser considerada como um primeiro passo para a aplicação do conceito de NFV para elementos de rede puramente óptica. Algoritmos para a sintetização da arquitetura do elemento de rede óptica são explorados por Garrich *et al.* (2012) e por (GARRICH; SIQUEIRA, 2014 -[P]) que exploram o conceito de arquitetura sob demanda para a construção flexível de blocos de ADD/DROP em ROADMs CDC.

O conceito de redes ópticas elásticas, com sistemas coerentes, com suporte a adaptação de formato de modulação dos transmissores conforme demanda de tráfego e às condições da rede, traz requisitos de NFV aos transmissores e receptores ópticos. Isto se dá porque os transponders que antes suportavam somente uma taxa e um formato de modulação passam a ter que suportar diferentes condições. Eventualmente, tais equipamentos poderão, por exemplo, alterar a imagem de *Field-Programmable Gate Arrays* (FPGAs), ou a disposição de circuitos implementados em *silicon photonics* para permitir a alteração dos parâmetros conforme esperado. O conceito de transponders multifluxo (JINNO; HIRANO, 2013) com suporte a processamento em camada 2, adaptação da taxa do lado rede e cross-conexão de ODU, possui características relacionadas a NFV.

A virtualização de funções de redes de transporte, juntamente com a revolução no modo de controle da rede e da configuração de regras de encaminhamento nas múltiplas camadas, viabilizado pelas redes definidas por software (SDN), traz uma nova perspectiva

para a forma de arquitetar novas redes e até mesmo na forma de interconexão de provimento de serviços com as redes atuais. Juntamente com o conceito de NFV, está ocorrendo uma grande evolução nos *chips* oferecidos no mercado, bem como nas FPGAs, e *Digital Signal Processors* (DSPs) que suportam cada vez mais desempenho e velocidade, oferecendo flexibilidade para a modificação nas funções realizadas pelo equipamento. Como exemplo, o fabricante Broadcom (BROADCOM, 2013) oferece a possibilidade do desenvolvimento de placas de linha para tratamento tanto de pacotes como de OTN, através de uma mesma matriz de comutação baseada em células. Desta forma, já se observa uma tendência em que os equipamentos de transmissão estão evoluindo para grandes plataformas com suporte a comutação tanto óptica como elétrica através de matriz de comutação baseada em células, e os grandes roteadores passam a suportar matriz de comutação óptica como alternativa para diminuir tráfego na matriz elétrica. Tal cenário apresenta novas possibilidades, porém inúmeros desafios.

## 4 Arquitetura SDN para NG-OTN com Base em Políticas

Este capítulo apresenta uma proposta de arquitetura SDN para controle de redes ópticas de transporte, com suporte a políticas. A arquitetura é constituída pelos seguintes pilares: (i) modelagem dos elementos de rede, incluindo suas interconexões, restrições e capacidades, utilizando a linguagem YANG; (ii) composição dos modelos dos elementos de rede e suas relações em um modelo que representa a rede, suportando transformações para representação da rede como grafos de propriedades; e (iii) um modelo de políticas que viabiliza a operação autônoma do controlador SDN, o qual permite a definição de políticas através de objetos associados ao grafo de propriedades da rede.

### 4.1 Objetivos

Conforme contextualizado no Capítulo 3, vários fatores têm gerado a necessidade de novos paradigmas para a arquitetura e operação das redes de telecomunicações, incluindo novos tipos de serviços com alto requisito de banda, novos tipos de dispositivos conectados, novas formas de implementar *data centers* utilizando tecnologias de virtualização, assim como novas formas de comercialização tais como *Software as a Service* (SaaS).

O paradigma SDN permite a simplificação na operação das redes, já que viabiliza a programabilidade destas, permitindo que tarefas realizadas atualmente de forma “manual” sejam feitas de forma automática e padronizada. O conceito de políticas é fundamental nas arquiteturas SDN, já que permite a definição das diretrizes e restrições de operação do controlador e das aplicações, viabilizando a operação autônoma. Desta forma, considera-se que a aplicação do paradigma SDN, em conjunto com PBNM, são viabilizadores do crescimento em escala e em complexidade das redes de telecomunicações, incluindo as redes ópticas de transporte.

No entanto, os conceitos sendo explorados e desenvolvidos no contexto de redes de pacotes e de *data centers* não podem ser aplicados diretamente às redes de transporte, pois estas possuem diversas particularidades como restrições de camada física, bloqueio e contenção de ROADMs, estruturas de mapeamento, multiplexação e comutação OTN.

Desta forma, observa-se a necessidade de estender a arquitetura SDN para que seus conceitos possam ser aplicados às redes ópticas de transporte, de forma a viabilizar o cres-

cimento em escala e complexidade destas redes, viabilizando também a integração destas às camadas superiores de forma transparente, com cooperação entre as camadas.

## 4.2 Requisitos e decisões de projeto

### 4.2.1 Requisitos da solução

Muitas propostas de soluções SDN têm sido realizadas pela academia e pela indústria. Com isso, surge a necessidade de definir quais são os princípios fundamentais de um sistema SDN. São considerados os seguintes princípios fundamentais (KREUTZ *et al.*, 2014) como referência para a concepção da arquitetura proposta nesta tese:

1. Desacoplamento dos planos de controle e de dados.
2. Decisões de encaminhamento são tomadas com base em fluxos ou circuitos.
3. A lógica de controle é implementada em uma entidade externa, o controlador SDN ou sistema operacional de rede – NOS, o qual provê recursos essenciais e abstrações para viabilizar a programação dos dispositivos de rede através de uma visão logicamente centralizada.
4. A rede é programável através de aplicações SDN.
5. Através do controlador, a rede deve operar de forma autônoma, adaptando parâmetros de operação, com base em políticas pré-definidas.

Considerando o trabalho realizado no contexto de órgãos de padronização com foco específico em SDN para redes ópticas de transporte e os princípios SDN listados acima, os seguintes requisitos foram identificados para a definição da arquitetura proposta nesta tese:

» **Deve permitir a implementação das funcionalidades disponíveis atualmente nos planos de controle baseados em ASON/GMPLS**, incluindo aspectos relacionados à proteção e restauração, robustez de soluções distribuídas, bem como possuir interfaces padronizadas como *External Network-to-Network Interface* (E-NNI), PCEP, *Multi-Technology Operations System Interface* (MTOSI) (TMFORUM, 2008), entre outras.

» **Deve viabilizar uma profunda simplificação** na forma de operar e integrar redes e sistemas.

» **Deve possuir interfaces abertas, modelos de dados e flexibilidade** para permitir a programabilidade de redes e serviços, incluindo suporte nativo à virtualização (*network slicing*).

Adicionalmente, o autor desta tese estabeleceu uma forte relação entre SDN e PBNM, utilizando como requisito da arquitetura SDN que o funcionamento do controlador SDN possa ser governado por políticas. Este requisito foi reforçado após as iniciativas recentes da Cisco (SMITH *et al.*, 2014) e o grupo de trabalho Congress (BALLAND; HINRICHS, 2014) liderado por Martin Casado, um dos principais integrantes da equipe da Universidade de Stanford, que trouxe OpenFlow à tona em meados de 2008.

Os principais objetivos do desenvolvimento do controlador T-SDN são:

- Validar que os princípios definidos para SDN sejam aplicáveis ao cenário de redes ópticas de transporte;
- Avaliação prática da utilização de aplicações logicamente centralizadas para o controle de funções como estabelecimento de circuitos;
- Controle de parâmetros ópticos como equalização e formato de modulação;
- Facilitar o provisionamento dinâmico de circuitos, através de interface *northbound* padronizada, facilitando a integração da rede de transporte a sistemas de controle de *data centers*.

No entanto, observa-se que não foram considerados requisitos para a construção de uma plataforma com arquitetura escalável ou com as melhores práticas de engenharia de software, já que se trata de uma prova de conceito. Desta forma, a concepção da arquitetura proposta foi realizada utilizando-se as seguintes premissas e princípios como base:

- » O controlador SDN deve ser validado em laboratório, através de rede óptica composta por ROADMs, amplificadores ópticos e transponders, disponíveis no laboratório de redes ópticas da Fundação Centro de Pesquisa e Desenvolvimento em Telecomunicações (CPqD). Os ROADMs foram desenvolvidos no centro, e utilizam como base de seu controle um agente de configuração denominado *NETCONF Configuration Agent* (NCA) que permite a modelagem do equipamento em YANG, de forma que os modelos das funções realizadas no equipamento, bem como a interface NETCONF, são gerados e disponibilizados de forma automática pelo NCA.
- » Não há restrições quanto à escolha do protocolo de configuração. A definição do protocolo deve ser realizada com base em características como padronização, extensibilidade e suporte a transações.

- » O controlador não deve se limitar ao controle de ROADMs, devendo suportar equipamentos com comutação multicamadas (lambda, sub-lambda, frames ou pacotes), suportando implementação de novos critérios para encaminhamento.
- » O controlador deve permitir o controle de isolamento de recursos de forma a permitir a implementação de virtualização de redes. Múltiplos fluxos de diferentes redes virtuais devem correr através do mesmo hardware e dos mesmos enlaces sem interferência, visibilidade ou troca de informações entre estes.
- » O controlador deve permitir a integração entre camadas, viabilizando a eliminação de ineficiências causadas pela utilização de diferentes mecanismos de roteamento das diferentes camadas.
- » Deve ser suportada a abstração recursiva dos elementos de rede e da rede, de forma que os mesmos mecanismos de cálculo e manutenção de caminhos, recuperação de falhas, OAM e virtualização possam ser utilizados de forma integrada no contexto dos elementos de rede como da rede, de forma a minimizar a necessidade de reimplementação de mecanismos já desenvolvidos no nível dos elementos de rede.
- » Deve viabilizar o gerenciamento das redes virtualizadas. Tal requisito apresenta vários desafios (cf. (ESTEVES *et al.*, 2013)), incluindo a criação de abstrações para gerência do substrato, redes virtualizadas e serviços, alocação dinâmica de recursos, separação entre planos de controle e gerência, definição de APIs e interfaces padronizadas.

#### 4.2.2 Decisões de projeto

Para a implementação de um controlador T-SDN logicamente centralizado, que permita que aplicações de rede e serviços rodem sobre este, são necessários dois blocos de construção fundamentais, que formam a base do sistema:

**Modelo de informações** para a representação e abstração dos elementos de rede, da rede e dos serviços instanciados sobre esta. No contexto desta tese, este modelo será denominado *Information Model for NE, Network and Services* (IM-NENS).

**Sistema operacional de rede** que tenha a capacidade de instanciar o modelo IM-NENS, realizar validação, sincronização destas informações com os demais componentes da solução, disponibilizando protocolos e APIs para realizar interface com outros componentes da solução, suportar transações e segregação de redes para implementação de virtualização (*network slicing*).

Atualmente, há uma tendência para a utilização YANG como *lingua franca* para a modelagem de informações de protocolos de redes definidos no IETF. Adicionalmente, YANG/NETCONF são a base para o padrão de configuração *ofconfig* utilizado pelo ONF. Instâncias de modelos YANG podem ser validadas quanto à sua consistência e podem sofrer transformações XML, incluindo para grafos de propriedades, os quais podem refletir a arquitetura interna dos elementos de rede, topologias da rede, dos serviços e suas relações. Desta forma, a vasta gama de algoritmos e ferramentas para a manipulação de grafos pode ser utilizada no contexto do controlador T-SDN. Portanto, uma das decisões de projeto é a utilização de YANG como padrão para modelos de informação no contexto do controlador T-SDN.

Para a implementação do núcleo do sistema operacional de redes, a principal implementação aberta, disponível para a academia é o *OpenDayLight*. No entanto, na época do desenvolvimento do trabalho desta tese, ele não estava disponível, de forma que decidiu-se pela utilização de uma ferramenta comercial, denominada ConfD (TAIL-F, 2013), a qual é referenciada no contexto desta tese como *NETCONF Configuration Agent* (NCA). A Figura 18 ilustra a arquitetura do NCA, que inclui funcionalidades como transações, compilação do modelo YANG gerando API com o plano de dados, validação do modelo, separação entre base de configuração e base operacional, interfaces *northbound* NETCONF e REST.

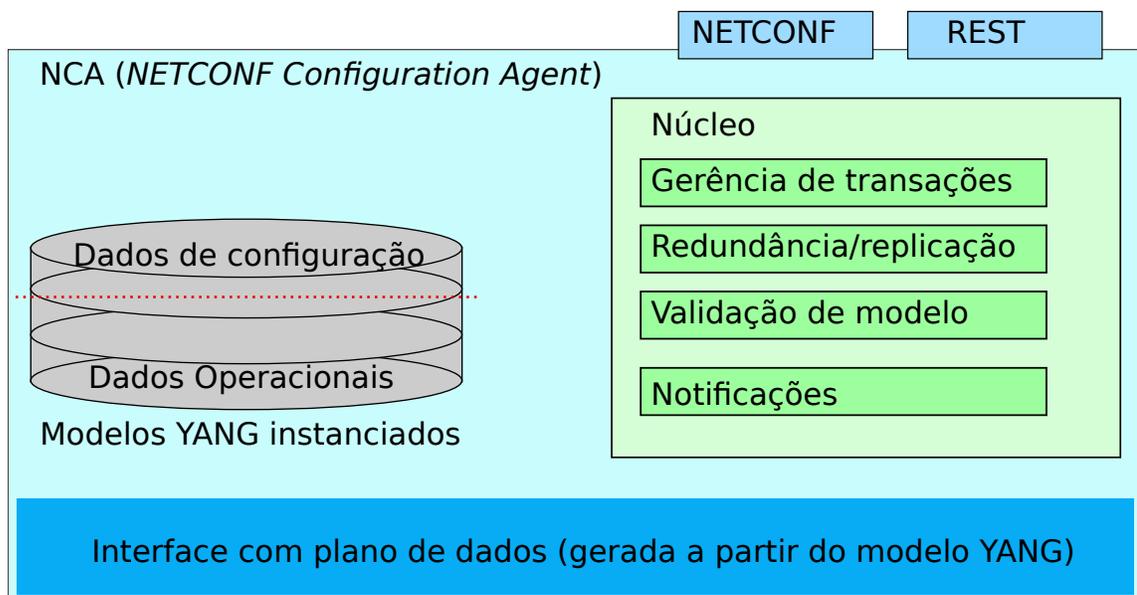


Figura 18 – Arquitetura do NCA

### 4.3 Arquitetura proposta

A arquitetura introduzida por M. Siqueira et al. [B] é ilustrada na Figura 19. A figura inclui os *Optical Network Elements* (O-NEs) e o controlador SDN denominado *Transport SDN Controller* (T-SDNC), que implementa funcionalidades como particionamento de redes (*slicing*), visão abstrata e unificada da rede, bem como uma API padronizada para que aplicações SDN possam implementar funções de controle de rede e de serviços.

O T-SDNC utiliza a mesma base do sistema de controle de um ROADM. Esta estratégia foi possível pela organização dos elementos de rede no NCA do controlador como se este fosse um grande NE composto por vários chassis. Desta forma, o T-NOS permite a abstração de toda a rede como se fosse um único equipamento. A matriz de conectividade entre os diferentes chassis virtuais é representada através de grafos de propriedades que permitem a representação da rede e das restrições das matrizes de cada ROADM, conforme ilustrado na Figura 19.

Desta forma, o T-SDNC permite que aplicações de otimização e controle da rede que rodam nos NEs possam ser migradas para o controlador, desempenhando suas funções com acesso às informações topológicas, de monitoração e configuração através de uma API provida por um sistema operacional de redes similar à API disponibilizada no próprio plano de dados do NE. Esta característica foi facilitada pela utilização do NCA como base para a construção do T-NOS. Observa-se que a API disponibilizada pelo NCA é proprietária, porém, como também externaliza interfaces do protocolo NETCONF e do protocolo equivalente ao NETCONF, baseado em REST (RESTCONF), esta pode ser considerada uma interface aberta para aplicações SDN.

Ao modelar equipamentos de transmissão utilizando YANG e disponibilizando interfaces NETCONF/RESTCONF, é possível definir estrutura, relações e restrições aos componentes dos elementos de rede, como WSS, *Optical Channel Monitor* (OCM), amplificadores ópticos e MCS. Tal modelo pode ser exportado e/ou transformado em estruturas que representam tais conexões e restrições como por exemplo grafos de propriedades. Desta forma, para a validação dos conceitos propostos nesta tese, são utilizados ROADMs modelados em YANG de forma nativa (HOOFT; MADEIRA, 2013), os quais expõem interfaces NETCONF/RESTCONF ao T-SDNC através de um aplicativo que faz o papel de sistema operacional do elemento de rede. Desta forma, o T-NOS elabora um ponto de vista global da rede, concatenando um subconjunto das informações dos O-NEs em um modelo unificado. No T-NOS, os diferentes O-NEs são organizados da mesma forma que placas de linha são organizadas em um sistema multichassi, onde cada O-NE é equivalente a uma placa de linha e a rede completa é equivalente a um elemento de rede multichassi. A separação pode ser

realizada de forma que o conceito de *network slicing* pode ser realizado de forma natural, permitindo a criação de diferentes instâncias virtuais dos recursos físicos, como elementos de rede, enlaces e comprimentos de onda.

O T-SDNC possui interfaces NETCONF/RESTCONF que permitem sincronização de informações dos modelos dos O-NEs com a base de dados do T-NOS. Adicionalmente, as mesmas interfaces poderiam ser disponibilizada como *northbound* do T-SDNC, permitindo aplicações externas como sistemas de gerência de redes gerenciarem a rede através do controlador SDN, de forma que possam ter a visão de toda a rede como um único *switch*.

### 4.3.1 Arquitetura de implementação do T-SDNC

A Figura 20 ilustra uma visão geral da arquitetura de referência para a implementação da camada de controle dos ROADMs a serem utilizados na prova de conceito do T-SDNC.

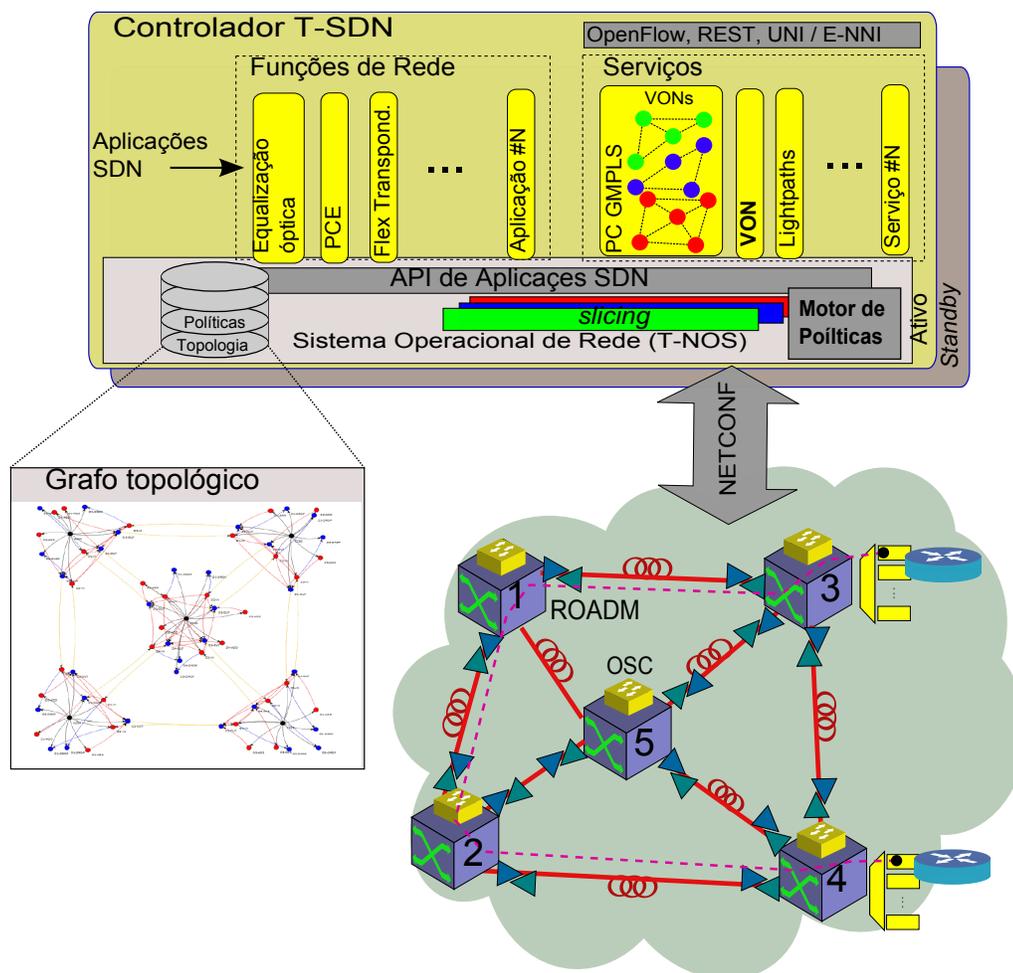


Figura 19 – Arquitetura do Sistema. Adaptado de [A]

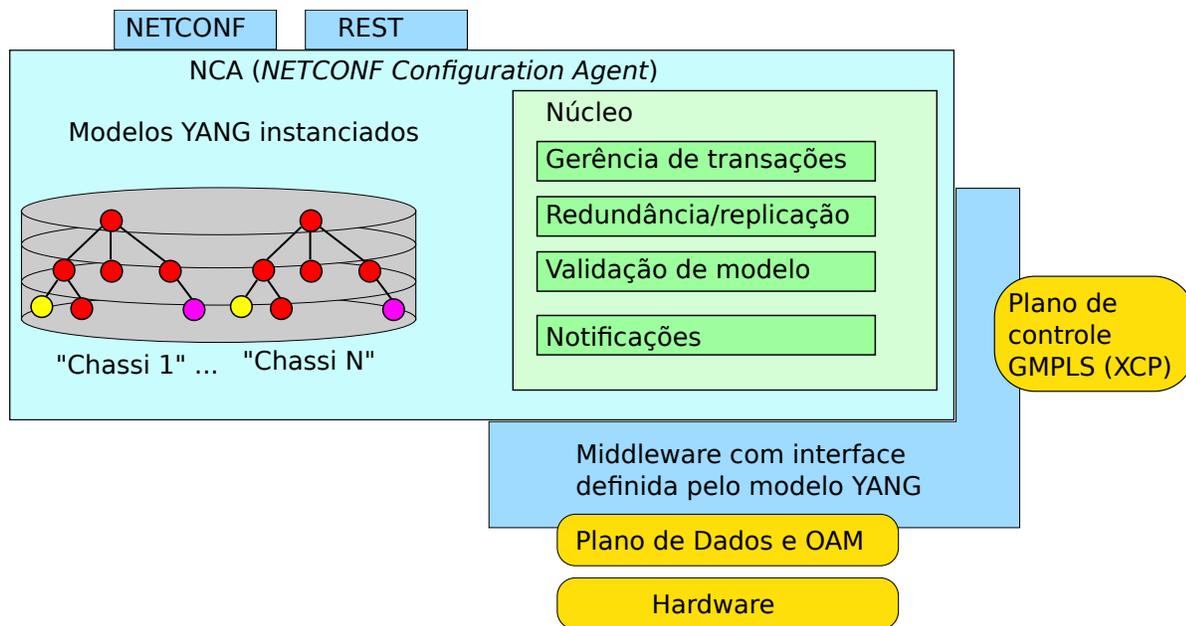


Figura 20 – Arquitetura de controle do ROADM baseada no NCA

Observa-se que é utilizada uma aplicação proprietária (o NCA) para a disponibilização de interfaces NETCONF e RESTCONF para controle do equipamento por um sistema de gestão. Adicionalmente, o NCA permite a geração automática, a partir dos modelos YANG, de interfaces para aplicações de controle do NE, bem como para o plano de dados, através de um *middleware* implementado no NCA. Observa-se que o NCA possui mecanismos para controle de transações, concorrência, validação do modelo, replicação da base de dados relativa à instanciação dos modelos YANG, que incluem dados operacionais e de configuração do NE. Observa-se que o modelo YANG do ROADM permite que um NE possa ter múltiplos chassis.

Observa-se que um detalhamento da arquitetura, utilizando modelos de referência como o *Reference Model of Open Distributed Processing* (RM-ODP), está fora do escopo desta tese, já que deveria ocorrer em uma etapa de implementação dos módulos de forma integrada, já considerando requisitos funcionais e não funcionais como desempenho e escalabilidade.

A Figura 21 ilustra como a solução de controle do ROADM foi estendida para ser utilizada como um controlador SDN que pode ser utilizado para controlar múltiplos NEs, suportando aplicações SDN. Observa-se que o modelo YANG dos ROADMs foi readequado para suportar mais um nível de hierarquia, de forma que cada NE possa ser representado como se fosse um chassis em um modelo de NE multichassi. Adicionalmente, observa-se que a interface do *middleware* que comunica-se com o plano de dados no ROADM foi conectada com um representante “virtual” de cada ROADM, que permite que as chamadas de funções

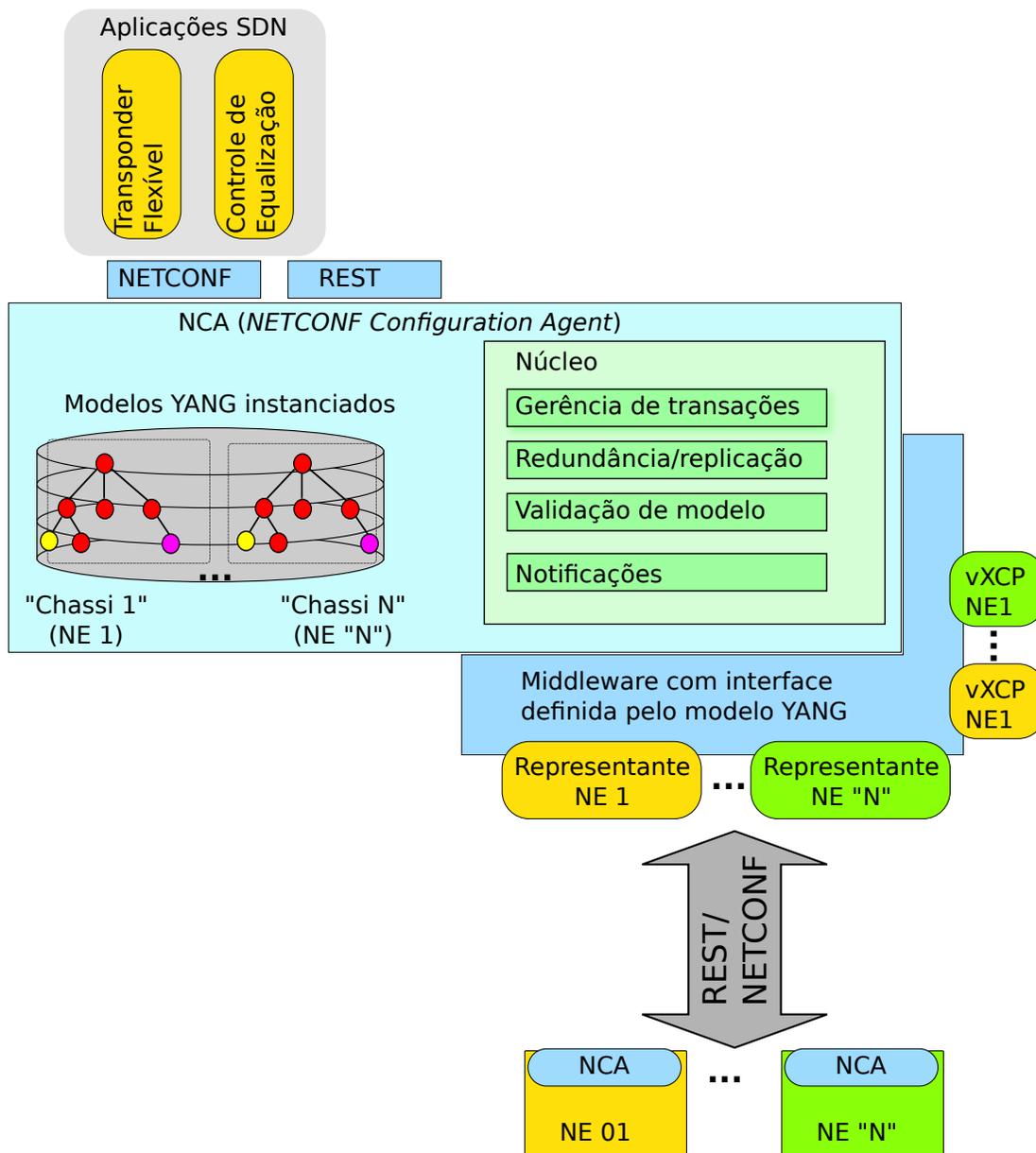


Figura 21 – Arquitetura de implementação do controlador T-SDN baseada no NCA

e notificações sejam repassadas ao ROADM real, através de NETCONF ou RESTCONF, o qual implementa seu próprio NCA com um espelho de um subconjunto do modelo do controlador, a partir de seu NE.

Observa-se que o NCA forma a base para o sistema operacional de rede previsto na arquitetura do controlador T-SDN.

### 4.3.2 Posicionamento da arquitetura proposta frente a esforços de padronização

Durante o ano de 2013, foi criado um grupo de trabalho em redes de transporte no ONF objetivando consolidar as propostas de extensões no protocolo *OpenFlow*, incluindo atividades como o estabelecimento de casos de uso, modelo de informação para redes de transporte e modelo de dados equivalentes a serem utilizados para a extensão dos protocolos padronizados pelo ONF. Desta forma, entende-se que o ONF caminha na direção de uma arquitetura SDN para redes de transporte baseada nos protocolos *OpenFlow* e *ofconfig* (ONF, 2014).

Já o OIF tem evoluído com uma proposta de arquitetura mais ampla, que inclui a utilização do protocolo OpenFlow para acesso aos equipamentos, mas que inclui a definição de interface REST entre o controlador de domínio e controladores da camada superior, como orquestradores. Adicionalmente, os esforços sendo realizados no OIF incluem propostas de criação de controladores SDN logicamente centralizados com a manutenção de interfaces já definidas no contexto de redes *Automatic Switched Optical Network* (ASON) como E-NNI, *User to Network Interface* (UNI), PCEP, além de interfaces de serviços como MTOSI.

Por outro lado, no IETF, propostas recentes de novos grupos de trabalho como SUPA ou *Yang Models for Configuration and Topology Abstraction* (YMCA) propõem a criação de modelos YANG para a representação de topologias da rede, permitindo a criação de abstrações para que aplicações SDN possam ter interfaces padronizadas para a realização da tradução de seus objetivos em configurações reconhecidas pelos elementos de rede. Adicionalmente, tem sido discutido no grupo *Software Defined Networking Research Group* (SDNRG) que há pouco valor em reinventar as funções que já são desempenhadas pelo plano de controle GMPLS em redes ópticas utilizando um paradigma SDN, no entanto, novas funções poderiam ser incorporadas ao cenário existente, como interfaces para controladores cliente e camada de virtualização de rede (LEE *et al.*, 2013).

## 4.4 Aplicações SDN

### 4.4.1 Plano de controle GMPLS virtualizado como aplicação SDN

Com o objetivo de manter interoperabilidade com redes legadas, bem como possibilitar funcionalidades providas pelo plano de controle GMPLS, denominado *eXtended Control Plane* (XCP) e ainda não implementadas como aplicações SDN, a arquitetura do T-SDNC prevê um ambiente de virtualização que permite a reconstrução da topologia dos elementos de rede em tal ambiente, bem como a interação do XCP com o T-NOS através da mesma API que é disponibilizada pelo elemento de rede para comunicação com o XCP distribuído, de

forma que não são necessárias alterações no XCP para que este opere no ambiente virtualizado. Este trabalho foi publicado no IMOC2013 [G] e apresentado informalmente na reunião trimestral do IETF que ocorreu em Julho de 2013. Uma abordagem similar foi introduzida em Outubro de 2013 no grupo de trabalho SDNRG do IETF (LEE *et al.*, 2013).

Outras possibilidades que surgem com esta abordagem são: implementação de VONs com diferentes instâncias de XCP, diminuição da complexidade e de máquinas de estados nos elementos de rede e atualização de software do XCP sem alterações nos elementos de rede.

Na seção 5.2 são apresentados resultados experimentais obtidos com a virtualização do XCP através do T-SDNC.

#### 4.4.2 Plano de controle baseado em SDN

No escopo desta tese, são considerados O-NEs (Ex.: ROADMs ou comutadores OTN) modelados em YANG de forma nativa [B], os quais expõem interfaces RESTCONF/NETCONF ao T-NOS através de um aplicativo que faz o papel de sistema operacional do elemento de rede. Desta forma, o T-NOS elabora um ponto de vista global da rede, concatenando um subconjunto das informações dos NEs em um modelo unificado. No T-NOS, os diferentes NEs são organizados da mesma forma que placas de linha são organizadas em um sistema multichassi.

A separação lógica permite a criação de diferentes instâncias virtuais dos recursos físicos, como elementos de rede, enlaces e comprimentos de onda. Este princípio de recursividade é similar ao modelo recursivo hierárquico de uma xBar lógica (MCCAULEY *et al.*, 2013) proposto para estender SDN/OpenFlow a redes de grande escala. A agregação de modo recursivo de blocos lógicos naturalmente introduz hierarquia à rede, evitando necessidade de mapeamento de modelos enquanto provendo uma clara separação de funções e papéis (ex.: roteamento, controle de acesso, engenharia de tráfego, detecção e tratamento de falhas, etc), porém permitindo mecanismos de controle de escopo local, como por exemplo, mecanismos de reação rápida a falhas, de forma que em larga escala a gerenciabilidade da rede como um todo seja facilitada.

A proposta de arquitetura se fundamenta no uso do modelo YANG do O-NE para representar suas restrições de encaminhamento internas, de forma que todas as possibilidades de conexões e características de cada caminho interno ao O-NE podem ser descobertas de forma automática pelo controlador. Desta forma, tanto os elementos de rede como a rede poderão ser representados como um grafo de propriedades  $G = (V; E)$ , onde  $V$  é o conjunto de vértices representando cada bloco de construção do O-NE (Ex.: WSS, amplificadores, OXC, matrizes agnósticas, etc) e  $E$  é o conjunto de arestas que representam as conexões

entre os componentes. Tanto os vértices como as arestas possuem propriedades específicas que caracterizam suas capacidades, características específicas e tipos de serviços suportados.

Desta forma, é possível a criação de uma base de dados de grafos (*graph database*) com uma abstração de toda a rede com suporte a determinado grau de visibilidade da parte interna de cada elemento de rede. O estabelecimento de caminhos pode ser realizado através de operações e algoritmos como travessia de grafos (*graph traversal*), com suporte a restrições através da verificação de atributos de vértices e arestas. A distribuição das funções de controle pode ser realizada utilizando o conceito de composição de grafos (BROCHELER *et al.*, 2010) através de bases de dados de grafos distribuídas ou pela replicação de subgrafos com visão mais abstrata no nível do sistema operacional de redes do controlador SDN.

Na seção 5.5 são apresentados resultados experimentais relativos à utilização do plano de controle baseado em SDN para o controle de uma rede constituída por elementos de rede de transporte flexíveis.

## 4.5 Modelo de políticas para T-SDN

No contexto do controlador T-SDN, há dois tipos principais de aplicações para sistemas de políticas: (i) controle de acesso a recursos, incluindo gerenciamento de concorrência entre aplicações SDN e autorização de comandos de operadores; e (ii) coordenar a operação autônoma do sistema, através de políticas que permitem a análise de condições da rede para a realização de ações, permitindo que a rede reaja de forma automática a situações que requerem ajustes em configurações, modo de operação, alterações no mapeamento de redes virtuais ou serviços na rede física.

Nesta tese, são explorados mecanismos de políticas no nível do controlador T-SDN objetivando coordenação e operação autônoma do sistema. Para tal, foram identificados os seguintes requisitos:

- Deve ser definido um modelo a partir do qual instâncias de políticas possam ser interpretadas e processadas por um motor de políticas, denominado *Policy Engine* (PEG).
- O PEG deve analisar as instâncias de políticas armazenadas em um repositório, verificando suas condições e caso necessário, devem ser acionadas aplicações SDN específicas, as quais deverão atuar sobre os elementos de rede através das interfaces do T-SDNC. Desta forma, evita-se que diferentes aplicações SDN atuem de forma concorrente e/ou conflitante.

- Deve haver um mecanismo de detecção e resolução de conflitos entre políticas, incluindo de forma preventiva, ou seja, antes que o conflito ocorra durante a operação do sistema.
- O motor de políticas deve ser capaz de analisar as condições e executar as ações com base nas informações da política armazenada no repositório. No entanto, o PEG deve ser agnóstico às informações específicas do modelo das políticas, de forma que suporte a análise de novos tipos de políticas sem que haja necessidade de alteração no PEG. Desta forma, o modelo de políticas deve conter informações e restrições suficientes para que o PEG possa atuar de forma agnóstica.

## 4.5.1 Arquitetura de políticas proposta

### 4.5.1.1 Modelo de políticas

Durante a última década, foram propostas diversas formas de modelar políticas, conforme discutido na seção 3.2. No entanto, a maioria destas propostas requer a especialização dos modelos de referência e do motor de políticas para tecnologias específicas. Desta forma, torna-se mais difícil aplicação de políticas fim-a-fim em cenários multi-tecnologias e/ou multi-fabricantes.

No entanto, o surgimento do paradigma SDN abriu uma nova perspectiva para a arquitetura dos sistemas PBNM, já que tais sistemas utilizam como princípio a criação de abstrações para as redes, incluindo o suporte a uma interface padronizada para a configuração do plano de dados dos equipamentos. Desta forma, os sistemas de políticas podem passar a atuar no nível do modelo de informações de abstração da rede, instanciado no controlador SDN, atuando de forma independente de tecnologias específicas empregadas no plano de dados dos elementos de rede.

Na arquitetura T-SDN baseada em NETCONF, as instâncias dos modelos de informação baseados em YANG dos elementos de rede podem sofrer transformações em grafos de propriedades, de forma que a composição destes grafos forma um supergrafo que representa a abstração da rede. Desta forma, através da utilização de linguagens de busca em bases de dados de grafos (WOOD, 2012; HOLZSCHUHER; PEINL, 2013), é possível a geração de subgrafos de interesse de análise da política, como por exemplo visões da topologia em diferentes camadas ou subgrafos dos serviços. Como exemplo, o subgrafo das conexões OTN no nível de *Optical channel Transport Unit* (OTU), permitem que as variáveis de desempenho desta camada, como taxa de erro pré FEC, possam ser acessadas na base operacional do modelo YANG a partir da transformação do subgrafo.

Desta forma, na proposta realizada nesta tese, ao invés de definir um modelo de

políticas genérico, mas que depende de uma especialização deste para a tecnologia alvo, é utilizada uma modelagem de políticas como grafo de propriedades, no qual o método de análise da política é simplificado, podendo ser inferido do próprio modelo IM-NENS, através da análise de atributos das arestas e vértices do grafo.

O método de análise de políticas é baseado na proposta denominada *policy flow*, introduzida por M. Siqueira et al. [C], no contexto de uma arquitetura que possui um motor de políticas distribuído. No entanto, no contexto desta tese a análise é distribuída através do grafo de políticas, de forma que a análise das condições da política ocorre em diferentes vértices e arestas do subgrafo de políticas, através da comparação de atributos operacionais dos vértices e arestas com faixas de valores pré-definidos no próprio modelo. Já a execução de ações das políticas no controlador T-SDN objetivam acionar a aplicação SDN mais adequada para mudar o estado da rede para que, da próxima vez que as condições foram verificadas, não sejam necessárias novas ações. No entanto, para que as aplicações SDN disponíveis não sejam parte do modelo das políticas, a ação limita-se a chamar uma função de um “despachador” do controlador SDN que recebe e mantém informações como objetivo e objeto da ação, histórico de chamada desta função pela política em questão e histórico de aplicações SDN que atuaram em chamadas anteriores. Desta forma, o “despachador” tem a capacidade de solicitar a atuação de aplicações SDN específicas que tenham-se registrado no controlador SDN, passando suas características relativas a quais suas áreas de atuação. Como exemplo, uma aplicação SDN que realiza equalização óptica global deverá informar o controlador SDN que suas funcionalidades incluem melhora de desempenho, com escopo na camada óptica.

Na proposta inicial do *policy flow* [C], a análise da política é realizada com base em um percurso pré-definido, no formato de um anel lógico que interconecta os elementos de rede que deverão fazer parte da análise da política. Já no contexto da proposta desta tese, o percurso da análise da política é definido como um algoritmo de travessia de grafo, o qual pode ser definido como atributo da própria instância da política. O algoritmo é definido através de alguma linguagem de busca em bases de dados de grafos, como *Cypher*<sup>1</sup> ou *Gremlin*<sup>2</sup>. A análise de políticas através da travessia de grafos foi estudada por Magrath (2006), em que o algoritmo *RETE* permite a realização de inferência de políticas modeladas em um grafo direcional acíclico, conforme detalhado na seção 3.2.1.

Para exemplificar a proposta, as condições da política podem definir que determinados contadores de desempenho devem ser verificados comparando-se às faixas de operação predefinidas no próprio modelo de abstração da rede – IM-NENS, sendo que os objetos de tal análise são definidos na política como um algoritmo definido em uma linguagem de busca em

<sup>1</sup> <http://neo4j.com/docs/stable/cypher-query-lang.html>

<sup>2</sup> <https://github.com/thinkarelius/titan/wiki/Gremlin-Query-Language>

grafos que retorna todos os vértices que representam pontos de monitoração não intrusiva de ODU.

Desta forma, observa-se que o grafo de cada política pode ser integrado ao “super grafo” de abstração da rede e de políticas, compartilhando vértices e arestas que definem tanto ligações entre entidades físicas ou lógicas da rede, como associações lógicas das políticas com determinados elementos lógicos, permitindo que as políticas possam ser associadas diretamente aos objetos do grafo que são de sua área de atuação.

#### 4.5.1.2 Policy flow

No contexto da proposta *policy flow* [C], foi projetado um mecanismo para a interpretação e implementação das políticas. O motor de políticas permite a avaliação de políticas complexas, modeladas não somente com o esquema tradicional de ECA, mas também considerando o conceito de situação, introduzido por Adi e Etzion (2002).

Uma visão geral sobre o fluxo de avaliação de políticas, incluindo avaliação de situação, é ilustrado na Figura 22. O *Policy Decision Point* (PDP) é formado por três estágios principais. O primeiro é responsável pelo recebimento de solicitações, eventos e pela identificação de situações que exigem análise de políticas e realização de ações. O segundo estágio é responsável pela análise de condições, autenticação, controle de mobilidade e gerenciamento de recursos (Ex.: banda disponível). O terceiro estágio é responsável pela execução de ações internas e externas. Ações internas ao PDP são executadas quando há necessidade de avaliações mais complexas, podendo utilizar o auxílio de simulações, ou mesmo quando há necessidade de nova análise de política (análise recursiva). Já as ações externas consistem na realização de configurações específicas nos elementos de rede para a implementação da política.

A Figura 22 ilustra o fluxo de avaliação de políticas pelo PDP realizado de uma forma centralizada. No entanto, as ANs requerem que o gerenciamento seja realizado de forma distribuída devido ao cenário dinâmico com mobilidade, não só de NEs, mas da própria rede. Desta forma, os componentes do motor de políticas foram reorganizados para que possam ser implementados através de *Autonomic Policy Servers* (APSs) e *Autonomic Policy Agents* (APAs), os quais realizam autodescoberta e comunicam-se através de uma infraestrutura de serviços web.

A utilização de serviços web como tecnologia de integração entre diferentes aplicações facilita o desenvolvimento de sistemas futuros interoperáveis, apresentando vantagens sobre tecnologias como *Common Object Request Broker Architecture* (CORBA), *Distributed Component Object Model* (DCOM), *Remote Method Invocation* (RMI) e outros, que adotam

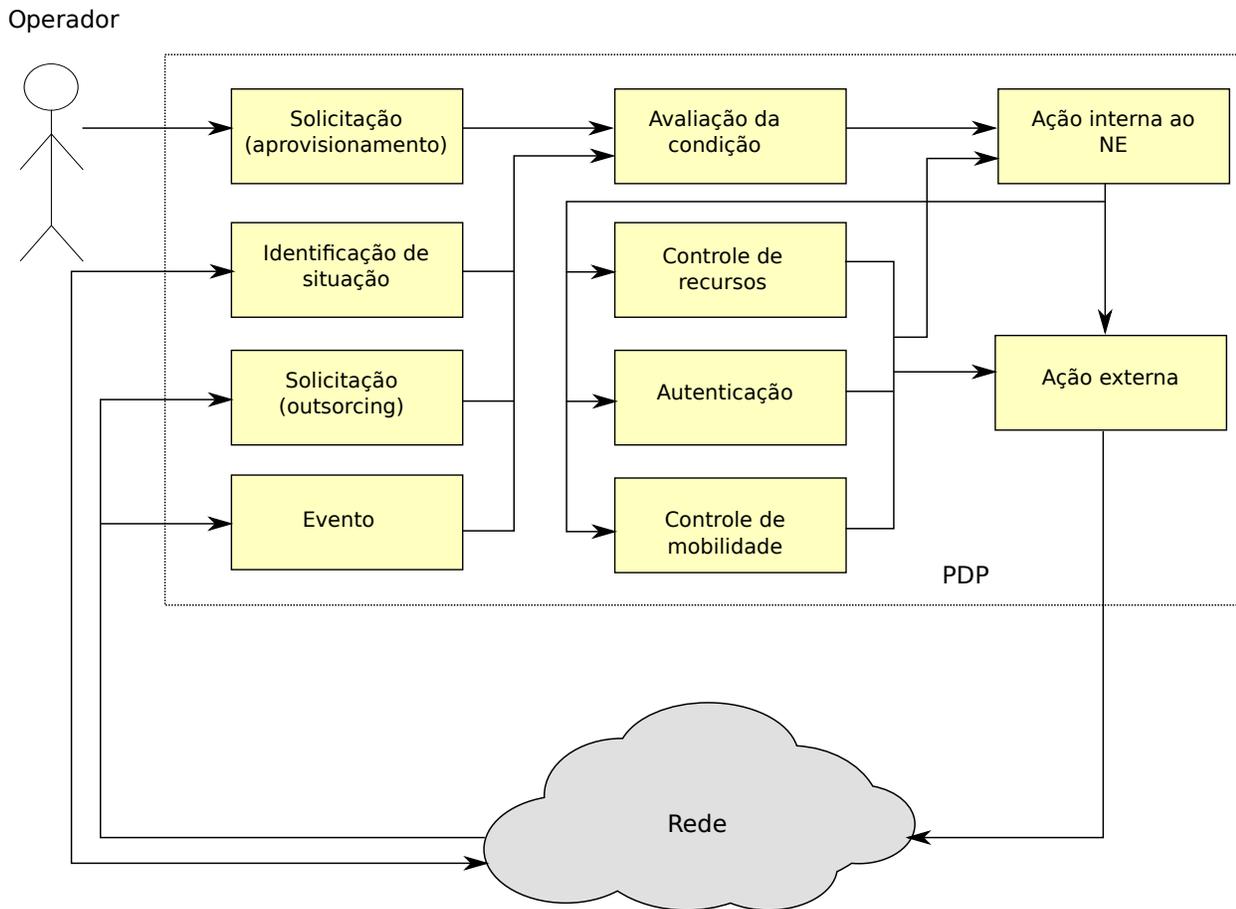


Figura 22 – Fluxo de avaliação de política. Fonte: [C].

mecanismos de distribuição fortemente acoplados, comunicação síncrona e fraca interoperabilidade com outras tecnologias e/ou implementações. O uso de XML e *Hypertext Transfer Protocol* (HTTP) tende a permitir um acoplamento mais fraco e um maior grau de interoperabilidade, já que permite que os serviços disponíveis sejam publicados e descritos.

A distribuição do motor de políticas foi realizado utilizando-se o conceito de *policy flow*. O APS é responsável pela ativação da política. Conforme ilustrado na Figura 23, o primeiro passo realizado pelo APS é a determinação do conjunto de APAs que formarão um anel lógico, criando caminhos lógicos entre estes. No passo 02 a política é distribuída pelos APAs e ativada. A partir deste ponto de operação, qualquer APA poderá disparar a análise e execução das ações da política na ocorrência dos eventos, solicitação ou identificação de situação pelo próprio APA.

O *policy flow* consiste na execução de um conjunto de funções relativas ao motor de políticas por um APA, o qual passa a responsabilidade pela análise e execução da política

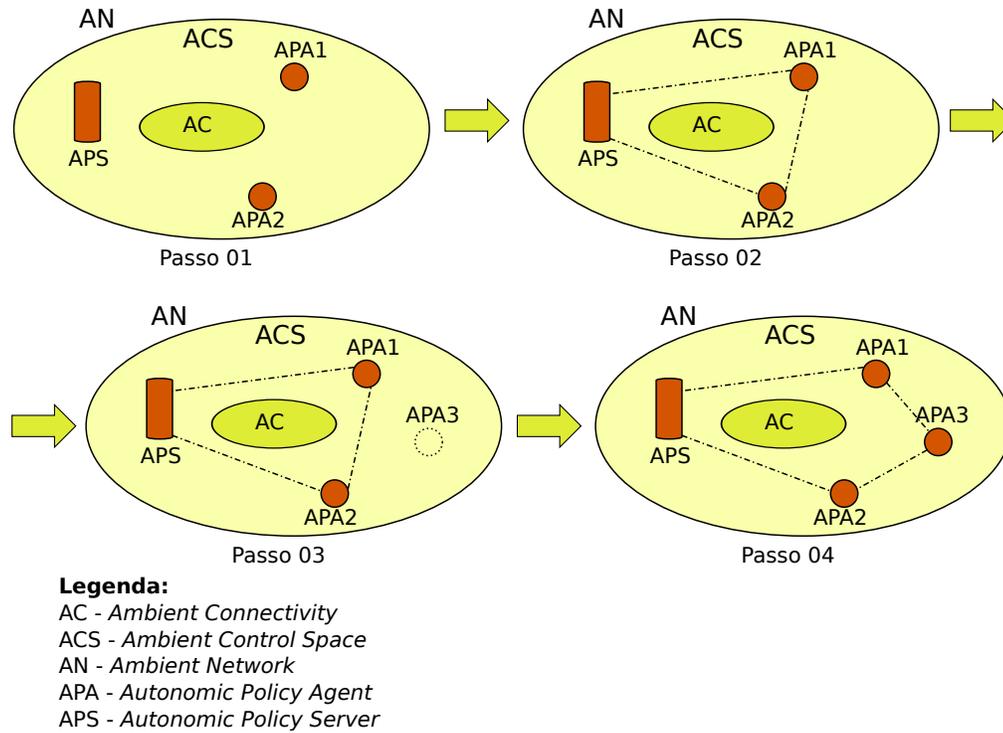


Figura 23 – Exemplo de operação do *policy flow*. Fonte: [C].

para o outro APA no anel, o qual continua a análise e execução se aplicável, passando ao outro APA no anel, até que todos os APAs no anel marquem a política como analisada e executada. Cada APA recebe a política em um estado de avaliação e execução, de forma que ele possa continuar a análise de onde o APA anterior parou ou mesmo possa realizar funções relativas ao seu escopo na rede. Caso um APA não tenha informações suficientes para realizar a análise da política ou realizar ações que deveriam ser de seu escopo, este recorre ao APS que tem uma visão global do *policy flow*, sendo capaz de alterar o anel lógico ou reconfigurar o APA em questão para que este seja capaz de realizar as tarefas necessárias para a avaliação e execução da política.

O APS é publicado como um serviço web, de forma que os APAs podem descobrir seus recursos disponíveis. No passo 03 da Figura 23, um novo APA é instanciado na rede. Neste caso, uma situação é identificada pelo APA1, relativo à política de agregação de novos nós à AN. Desta forma, o APA1 realiza o processo de avaliação da política, realizando a avaliação das condições da política, executando uma ação interna de delegar a autenticação do novo APA para o APS, já que esta decisão não pode ser tomada por um APA. Conforme ilustrado no passo 04, o APS autentica o APA3, reconfigurando o anel lógico com o novo APA. O exemplo ilustrado é simples, porém o mecanismo proposto permite a implementação de políticas complexas, de forma distribuída.

#### 4.5.1.3 Modelo de políticas para SDN baseado em grafos

A maioria das arquiteturas de políticas propostas na literatura é baseada em um modelo de informações e um motor de políticas. Políticas são comumente definidas como *eventos*  $\Rightarrow$  *condições*  $\Rightarrow$  *ações*, tendo como principal objetivo mapear políticas abstratas, de alto nível, para configurações específicas nos elementos de rede. O conceito de SDN é mais extensivo, incluindo mais funções que uma plataforma de gerência com base em políticas. No entanto, arquiteturas SDN definem um sistema operacional de rede, e aplicações SDN que requerem definições de políticas para governar sua operação, da mesma forma que sistemas operacionais de computadores possuem, como por exemplo políticas de escalonamento, de alocação de memória e de segurança (SMALLEY *et al.*, 2001; LOVE, 2010).

O conceito de *policy flow* para análise de políticas é proposto por M. Siqueira *et al.* [C] como uma estratégia de análise distribuída de políticas complexas, utilizando uma abordagem similar a agentes móveis, onde cada *Local Policy Decision Point* (LPDP) executa parte do *policy flow*. Adicionalmente, a avaliação das políticas inclui o conceito de análise de situação, no qual uma análise mais ampla sobre a situação da rede é realizada em comparação às abordagens tradicionais que utilizam análise do tipo *eventos*  $\Rightarrow$  *condições*  $\Rightarrow$  *ações*.

No contexto SDN, é proposta a implementação do *policy flow* com uma abordagem orientada à análise de situação através da análise de grafos [A], de forma centralizada ou distribuída, através da utilização de bases de dados não relacionais, orientadas a grafos (*graph database*). Como os elementos da rede óptica, como ROADMs, são modelados em YANG, incluindo suas relações, restrições e interconexões, abstrações em formato de grafo podem ser geradas de forma automática através de transformações. Desta forma, as políticas podem ser modeladas através de operações de grafos como algoritmos para travessia de grafos, criação de subgrafos, grafos com eventos (*listenable graphs*), análise de atributos de vértices e enlacs.

Com o objetivo de distribuir a implementação do motor de políticas, de forma a permitir a implementação de controladores SDN distribuídos, o conceito de composição de grafos (BROCHELER *et al.*, 2010) pode ser utilizado, através da utilização de bases de dados de grafos distribuídas<sup>3</sup>, ou através da replicação de subgrafos abstratos no sistema operacional de rede, no nível do controlador SDN.

O *policy flow* proposto para a validação do conceito, que inclui *evento*  $\Rightarrow$  *identificação de situação*  $\Rightarrow$  *análise de condição*  $\Rightarrow$  *seleção e execução de ações*, é ilustrado na Figura 24. Como exemplo, após falha ou degradação em um receptor, como um transponder óptico, a verificação da situação é realizada através da análise do grafo com a visão das informações necessárias para a avaliação da política, através de operações sobre a base de dados de

<sup>3</sup> <http://thinkaurelius.github.io/titan/>

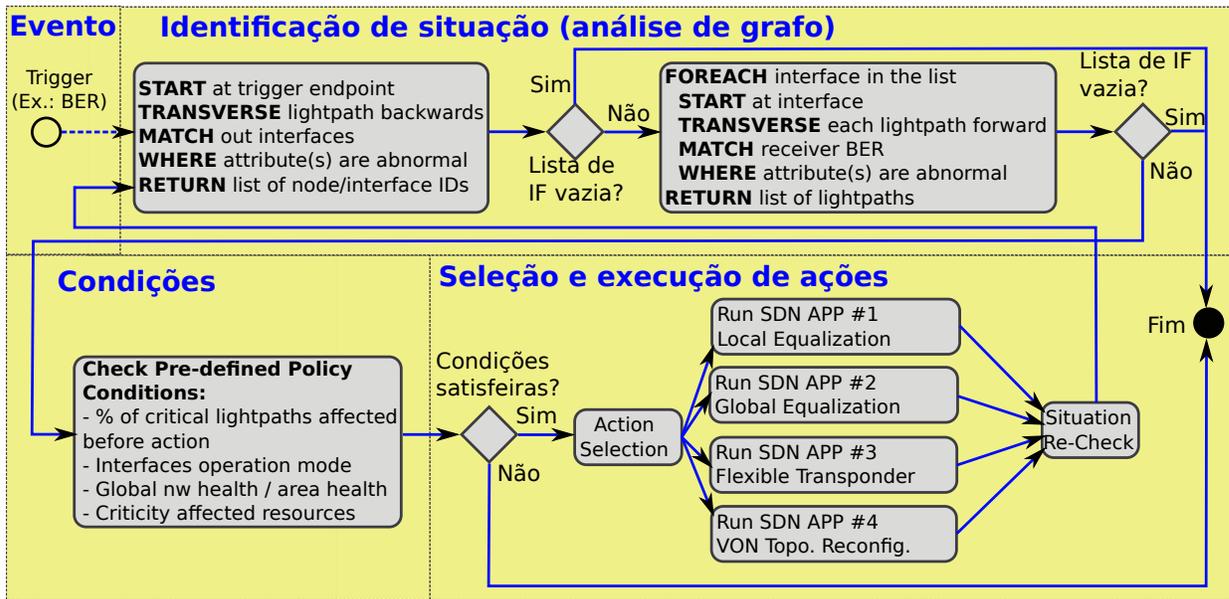


Figura 24 – Exemplo de um modelo de políticas para equalização. Fonte: [A].

grafo (mostrado com pseudocódigo, similar à linguagem de grafos Cypher<sup>4</sup>). A verificação da situação permite a verificação de potenciais interfaces que poderiam ser reconfiguradas com o objetivo de resolver a causa raiz do problema, bem como iniciar análise de outros canais potencialmente afetados. Após a identificação das interfaces alvo, políticas pré-configuradas são analisadas com o objetivo de determinar quais ações devem ser executadas. Após a execução das ações selecionadas, a situação é verificada novamente, de forma que seja avaliado se as ações resolveram o problema. Se não, outras ações podem ser selecionadas. Como exemplo, se o evento de BER fora dos limites não for resolvido através dos mecanismos de equalização e controle de ganho, se o canal é crítico (requisito de negócio) e se outras condições são satisfeitas, o *lightpath* deve ser re-roteado. Observa-se que diferentes aplicações SDN podem ser invocadas em um mesmo *policy flow*.

Desta forma, no contexto deste trabalho, o principal propósito do motor de políticas é prover controle automático da infraestrutura de rede e do substrato virtualizado. A arquitetura proposta permite que um provedor de VON exponha diferentes visões da rede virtualizada para diferentes clientes, com políticas distintas para cada *network slice*. Desta forma, cada cliente pode ser o responsável por gerenciar sua rede virtual, incluindo suas políticas, as quais são aplicadas pelo PEG do T-SDNC.

No capítulo 5 são apresentados resultados experimentais relativos à aplicação de políticas no contexto do T-SDNC, incluindo um caso de uso, descrito na seção 5.3, em que é realizada reconfiguração dinâmica de VON com base em políticas e uma aplicação SDN para

<sup>4</sup> <http://www.neo4j.org/learn/cypher>

reconfiguração dinâmica do formato de modulação que utiliza políticas para definição de seu comportamento, descrito na seção 5.4.

#### 4.5.2 Planejamento automatizado como suporte à análise de políticas

Planejamento é um processo que permite a escolha e organização de ações através da antecipação dos resultados esperados ou de seus efeitos. Conforme definido por Srivastava e Kambhampati (2005), um problema de planejamento  $PP$  é uma tupla  $\langle P, I, G, A \rangle$  onde  $P$  é um conjunto de predicados,  $(I \subseteq P)$  é a descrição do estado inicial,  $(G \subseteq P)$  é a descrição do estado objetivo, e  $A$  é o conjunto de ações a serem executadas. A especificação de uma ação consiste de pré-condições ( $A_i^{pre} \subseteq P$ ) e pós-condições ( $A_i^{pos} \subseteq P$ ). Um plano para  $PP$  é uma sequência de ações  $S$ , no qual se  $S$  é executado em  $I$ , o estado resultante do ambiente conterà  $G$ . Desta forma, um planejador determina um plano através da busca no espaço de possíveis configurações de estados ou sequenciamento de ações.

Planejamento automatizado (GHALLAB *et al.*, 2004) é uma área da IA que estuda mecanismos para a realização de planejamento utilizando auxílio computacional. O planejamento automatizado pode ser aplicado a sistemas que exigem comportamento autônomo em ambientes que possam ser modelados. Comumente, as soluções são complexas e envolvem o tratamento de múltiplas variáveis e/ou espaços.

Conforme analisado por Srivastava e Kambhampati (2005), os requisitos para planejamento em sistemas de computação autônoma podem ser resumidos como se segue:

- A representação de um plano pode ser tão geral quanto um *workflow*, contendo construções sequenciais, condicionais, paralelas e *loops*.
- Um plano é necessário, mesmo que informações sobre os estados iniciais, objetivos e ações não estejam disponíveis.
- Geração automática do plano é importante, mas estas também podem ser obtidos de operadores ou métodos dependentes do domínio.
- A geração de novos planos deve considerar o histórico de planos passados gerados e executados, obtidos através de um repositório.
- Tipicamente, os planos são executados de forma centralizada, porém podem ser distribuídos dependendo do cenário.

O conceito de planejamento automatizado é utilizado para o auxílio na definição de planos na camada de conhecimento do controlador SDN. Com o mapeamento da topologia

das diferentes camadas através de modelos, bem como descoberta automática de topologia, a abordagem SDN permite que ferramentas de dimensionamento de rede funcionem de forma acoplada ao controlador para que as premissas iniciais de dimensionamento sejam verificadas de forma contínua. Desta forma, no escopo desta tese foi desenvolvida uma ferramenta de planejamento de redes multicamadas que se integra ao controlador SDN.

A ferramenta, denominada *Network Planner*, foi desenvolvida utilizando-se a API *Java Universal Network/Graph Framework* (JUNG)<sup>5</sup>, de forma que pode obter a topologia do controlador em forma de grafo de propriedades, podendo realizar simulações com informações topológicas, de capacidade e demandas obtidas de forma automática do controlador. No entanto, a ferramenta também pode obter informações de arquivos do tipo *GraphML*, já que pode ser utilizada para o dimensionamento de redes não existentes. Operando de forma acoplada ao T-SDNC, a ferramenta de planejamento permite que situações na rede possam ser identificadas de antemão, permitindo que o PEG realize ajustes na rede de forma preventiva, conforme definição das políticas. Como exemplo, a ferramenta pode obter informações do controlador como: (i) lista de estações (sites); (ii) cabos ópticos interconectando as estações, incluindo identificação da região do cabo, para divisão da rede em regiões geográficas; (iii) lista de comprimentos de onda que interconectam os equipamentos da camada superior (roteadores, OTN *Switches*, etc), incluindo a associação destes com os cabos ópticos que utilizam, matriz de demandas; e (iv) estratégia de sobrevivência de cada demanda (Ex.: 1+1, 1+R, 1+1+R). Com estas informações, a ferramenta pode realizar simulação de falhas de enlaces conforme estratégias predefinidas na política, permitindo dimensionamento de capacidade adicional para restauração. Estratégias de falha suportadas incluem: (a) falhas simples em toda a rede ou por região de forma simultânea e (b) falhas duplas em toda a rede ou por região de forma simultânea.

Desta forma, o T-SDNC pode receber resultados de simulação da ferramenta e realizar a reconfiguração de forma automática da rede, incluindo: (1) alterações no mapeamento das camadas (Ex.: instanciação de novos caminhos ópticos entre comutadores elétricos, de forma a atender aos requisitos de sobrevivência especificado para as demandas); (2) pré-planejamento de caminhos para mecanismos de restauração distribuídos; (3) ajuste fino dos tipos de proteção e restauração das demandas, evitando que ocorra bloqueio na restauração de demandas prioritárias em situação de falhas.

Na seção 5.6 são apresentados resultados experimentais relativos à utilização da ferramenta de planejamento operando em conjunto com o T-SDNC para o ajuste dinâmico de políticas de sobrevivência através de planejamento em operação. Adicionalmente, na seção 5.7, são apresentados resultados da utilização da ferramenta de planejamento para o

---

<sup>5</sup> <http://jung.sourceforge.net/>

ajuste fino do modo de operação de mecanismos de controle locais (localizados nos NEs), que deverão atuar em caso de falha no controle centralizado.

No próximo capítulo são apresentados resultados experimentais e de simulações realizados para a validação dos conceitos propostos em um conjunto de casos de uso.

## 5 Aplicações e Casos de Uso

Este capítulo apresenta resultados experimentais obtidos com o principal objetivo de validar as propostas realizadas no contexto desta tese. Os experimentos foram organizados nos seguintes casos de uso:

- UC01 – Virtualização da rede com suporte a GMPLS e instanciação automática de VONs.
- UC02 – Reconfiguração dinâmica de VON com base em políticas.
- UC03 – Aplicação SDN para reconfiguração dinâmica do formato de modulação.
- UC04 – Plano de controle SDN com suporte a NEs flexíveis.
- UC05 – Ajuste dinâmico de políticas de sobrevivência através de planejamento em operação.
- UC06 – Ajuste fino de políticas locais via planejamento automatizado

Estes experimentos têm como objetivo a realização de provas de conceito, permitindo a validação de que a arquitetura proposta atende aos requisitos introduzidos no Capítulo 4, principalmente em relação à aplicabilidade de SDN em redes ópticas de transporte. No entanto, vale ressaltar que não objetivam a realização de testes funcionais da arquitetura de implementação com seus blocos de construção integrados, os quais deveriam ser realizados em uma eventual fase de desenvolvimento do projeto.

### 5.1 Ambiente experimental

Para a realização dos experimentos relativos aos casos de uso UC1 e UC2, foi utilizada a rede experimental do laboratório de redes ópticas da Fundação CPqD, ilustrada na Figura 25. A mesma é composta por cinco NEs e o controlador SDN. Cada NE inclui um ROADM multi-direcional, amplificadores ópticos, canais de supervisão e transponders ópticos. Apesar de as placas WSS suportarem configuração *flex-grid*, os experimentos foram realizados com canais na grade DWDM com espaçamento de 50GHz, padronizada pelo ITU-T. O controlador SDN que executa o T-NOS e as instâncias virtualizadas do XCP é equipado com um processador Intel Core i7, 2.8GHz, com 8GB de RAM.

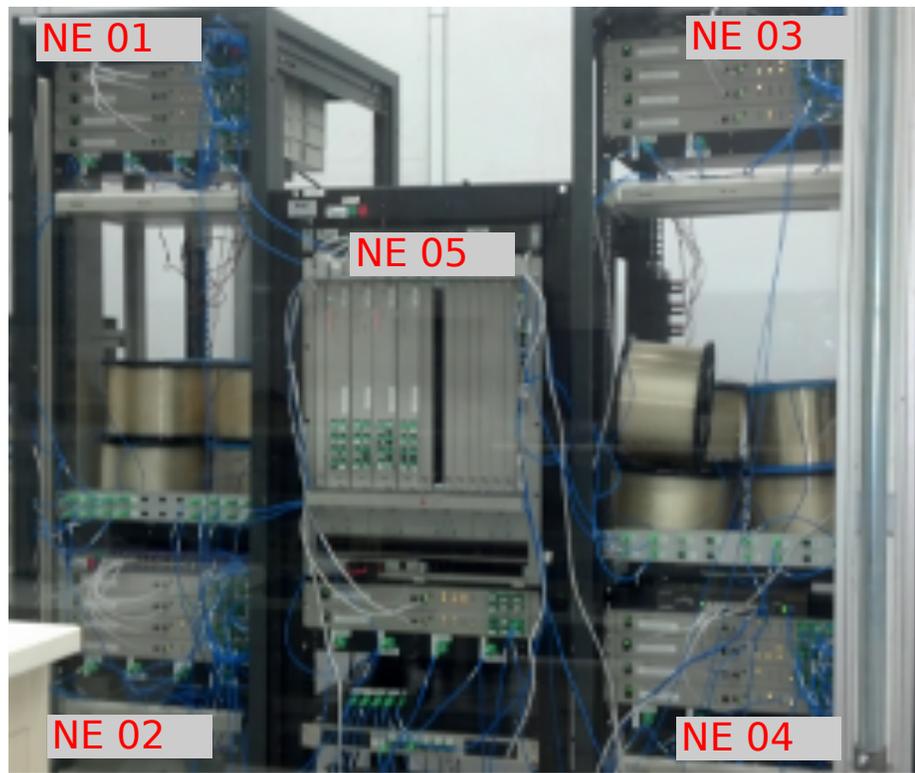


Figura 25 – *Testbed* composto por cinco ROADMs WSS. Fonte: [B].

## 5.2 UC01 – Virtualização da rede com suporte a GMPLS e instanci- ação automática de VONs

Este experimento, introduzido por M. Siqueira et al. [B], tem como objetivos: (i) validar a capacidade do controlador T-SDN de realizar a segregação dos recursos de rede (*network slicing*); (ii) validar o mecanismo de descoberta de topologia baseado em *Link Layer Discovery Protocol* (LLDP), com suporte a criação automática de VONs; e (iii) demonstrar a aplicabilidade da execução de diferentes instâncias do plano de controle GMPLS como aplicação SDN, em que cada instância tem visibilidade sobre sua VON. A implementação de GMPLS para redes ópticas do CPqD, denominada *eXtended Control Plane* (XCP), foi utilizada. Um dos fatores cruciais para a virtualização do XCP é a possibilidade deste receber informações de falhas em enlaces da mesma forma que ele receberia se estivesse rodando localmente no elemento de rede. Desta forma, para a identificação de falhas e a consequente atualização de estado dos TE-Links, através do protocolo LMP, as informações que causariam o LMP perceber as falhas são modeladas em YANG, de forma que, contando que haja priorização na sincronização destas informações entre o elemento de rede e o T-NOS, não há prejuízo funcional no LMP devido ao fato dele rodar virtualizado como aplicação do controlador SDN.

As instâncias do XCP foram virtualizadas através de *Linux Containers* (LXC)<sup>1</sup>, de forma que é possível rodar várias dezenas de instâncias em uma mesma máquina física. No entanto, nada impede que as instâncias de XCP virtualizado sejam distribuídas em mais de uma máquina física.

De modo a permitir a autoconfiguração da instância principal do XCP virtualizado, incluindo elementos de rede e TE-Links virtuais, os O-NEs rodam LLDP (IEEE, 2009), o qual troca mensagens de descoberta de vizinhos através do canal de supervisão da rede DWDM. Desta forma, novos elementos de rede podem ser iniciados com configurações mínimas, como por exemplo identificador e tipo. Cada NE possui um comutador Ethernet operando como canal de supervisão, com  $N$  portas do tipo 100BaseFX equipadas com *Small Form-factor Pluggables* (SFPs) que operam no comprimento de onda 1510nm, destinado ao canal de supervisão fora da banda. Cada direção do ROADM é configurada com uma VLAN específica, de acordo com a seguinte regra: (DIR 1, VLAN 100; DIR 2, VLAN200; DIR N, VLAN N).

O controlador é composto por um comutador Ethernet e uma placa de supervisão. A conexão entre estes é formada por um tronco do tipo 802.1q o qual transporta  $N$  VLANs. O Linux da placa supervisora é configurado com  $N$  sub-interfaces, uma para cada VLAN do tronco. Deste modo, forma-se um domínio de *broadcast* para cada conexão ponto-a-ponto entre NEs. Esta configuração permite que os processos LLDP executando em diferentes supervisoras descubram seus vizinhos, preenchendo tais informações na base de dados YANG local, de forma que as informações de topologia são consolidadas no T-NOS através de NETCONF. Desta forma, a aplicação *eXtended Control Plane Manager* (XCP-Mgr) lê as informações topológicas consolidadas no T-NOS, podendo, desta forma, instanciar os elementos de rede e TE-Links da VON principal.

A Figura 26 ilustra a captura de pacotes LLDP no NE02. As mensagens recebidas dos NEs 02 e 04 contêm uma TLV do tipo 127, identificando a origem. Utilizando estas informações, incluindo VLANs local e remota, o NE preenche sua base de dados YANG referente a informações de descoberta de topologia.

As informações de descoberta de topologia são consolidadas automaticamente no T-NOS, e utilizadas pelo XCP-Mgr para a instânciação da VON principal, contendo todos os NEs e suas interligações, bem como as instâncias virtuais do XCP e respectivos TE-Links, sendo que uma máquina virtual LXC é instanciada para cada NE real.

VONs adicionais podem ser instanciadas pelo XCP-Mgr com base em arquivos XML com a definição da topologia da VON. Desta forma, a Figura 27 ilustra a VON principal com seus respectivos NEs virtuais e TE-Links autoconfigurados pelo XCP-Mgr. Adicionalmente,

---

<sup>1</sup> <http://lxc.sourceforge.net/>

Frame 7: 135 bytes on wire (1080 bits), 135 bytes captured (1080 bits)

Ethernet II, Src: TaiwanCo\_04:51:58 (00:03:1d:04:51:58), Dst: LLDP\_Multicast (01:80:c2:00:00:0e)

802.1Q Virtual LAN, PRI: 0, CFI: 0, ID: 100

000. .... = Priority: Best Effort (default) (0)

...0 .... = CFI: Canonical (0)

.... 0000 0110 0100 = ID: 100

Type: 802.1 Link Layer Discovery Protocol (LLDP) (0x88cc)

Link Layer Discovery Protocol

Chassis Subtype = MAC address, Id: 00:03:1d:04:51:58

Port Subtype = Interface name, Id: mgmt0.300

Time To Live = 120 sec

Port Description = mgmt0.300

System Name = ame250.(none)

System Description = i686/Linux 3.2.0-29-generic-pae

Capabilities

Management Address

Unknown - Unknown

1111 111. .... = TLV Type: Organization Specific (127)

.... ..0 0000 1001 = TLV Length: 9

Organization Unique Code: Unknown (0x435051)

Unknown Subtype Content: 000118000000

End of LLDPDU

NE 2: Informações LLDP na base de dados YANG local

```
lldp-topo {
  local_node_id 2;
  local_if_index 20 {
    remote_node_id 1;
    local_vlan 100;
  }
  local_if_index 23 {
    remote_node_id 4;
    local_vlan 300;
  }
}
```

0000 01 80 c2 00 00 0e 00 03 1d 04 51 58 81 00 00 64 ..... ..QX...d

0010 88 cc 02 07 04 00 03 1d 04 51 58 04 0a 05 6d 67 ..... .QX...mg

0020 6d 74 30 2e 33 30 30 06 02 00 78 08 09 6d 67 6d mt0.300. ..x..mgm

0030 74 30 2e 33 30 30 0a 0d 61 6d 65 32 35 30 2e 28 t0.300.. ame250.(

0040 6e 6f 6e 65 29 0c 1f 69 36 38 36 2f 4c 69 6e 75 none)..i 686/Linu

0050 78 20 33 2e 32 2e 30 2d 32 39 2d 67 65 6e 65 72 x 3.2.0- 29-gener

0060 69 63 2d 70 61 65 0e 04 00 80 00 80 10 0c 05 01 ic-pae.. .....

0070 00 00 00 00 03 18 00 00 00 00 fe 09 43 50 51 00 ..... ..CPO

0080 01 18 00 00 00 00 00 Tipo 00: ROADM

NE ID remoto: 01

Indice da interface remota: 24 (0x18)

Unknown Subtype Content (lldp.u... Packets: 40 Displayed: 40 Marked: 0 Load time: 0:00.000

Figura 26 – Pacotes LLDP e modelo YANG. Fonte: [B].

a figura mostra a VON HS (VON de alta velocidade - *High Speed*) que inclui os NEs 01, 03 e 04. Na parte inferior da figura o estado dos TE-Links é ilustrado, através da CLI do T-NOS. Como pode ser observado, cada VON possui um subconjunto dos comprimentos de onda disponíveis para a instanciação de *lightpaths*.

Com o objetivo de demonstrar a operação do plano de controle (XCP) sobre o T-NOS, suportando duas VONs distintas, um conjunto de vinte LSPs foram estabelecidos na VON principal, onde dez são roteados diretamente entre os NEs 02 e 04, e outros dez passam pelos NEs 01 e 03. Por outro lado, na VON HS, um total de sete LSPs foram estabelecidos entre os NEs 01 e 04, passando pelo NE 03. Todos os procedimentos de RWA utilizados são descritos por Santos *et al.* (2009). A Figura 28 ilustra as mensagens RSVP-TE para estabelecimento de um dos caminhos no NE01 da VON HS.

A Figura 29 mostra o resultado do comando para listagem das cross-conexões de cada

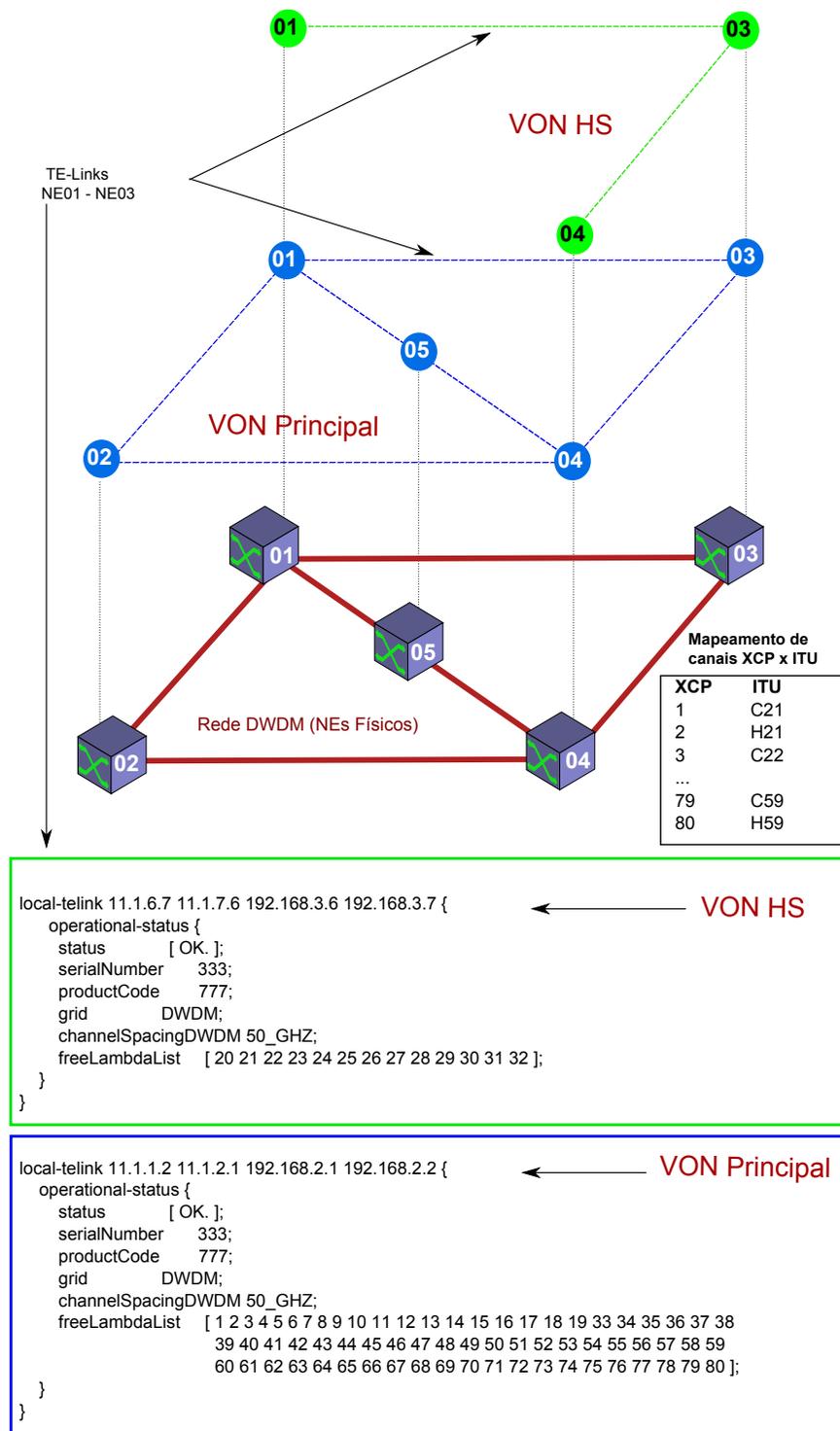


Figura 27 – VONs principal e HS, seus respectivos NEs, TE-Links e mapeamento. Fonte: [B].

VON realizado através da CLI do T-NOS. Tais cross-conexões são criadas em função do estabelecimento dos LSPs pelo XCP.

Como os ROADMs são *flex-grid*, as cross-conexões são mostradas utilizando a repre-

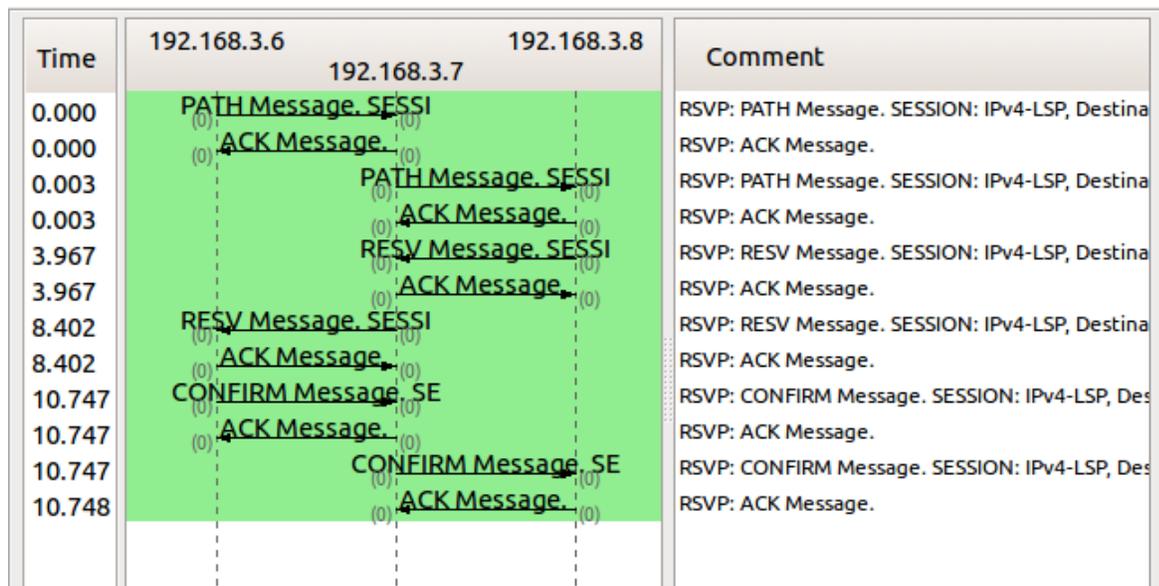


Figura 28 – Mensagens RSVP durante estabelecimento de *lightpath*. Fonte: [B].

sentação *flex-grid*, conforme definido na ITU-T G.694.1 (ITU-T, 2012). Para a grade flexível, os slots de espectro possuem uma frequência central (em THz) definido por  $193.1THz + (n * 0.00625THz)$ , onde “n” é um inteiro positivo, negativo ou zero e  $0.00625THz(6.25GHz)$  é a granularidade da frequência nominal central; e a largura do slot é definida por  $(12.5 * m)$ , onde “m” é um inteiro positivo e 12.5 é a granularidade da largura do slot em GHz. Qualquer combinação de slots de frequência é permitida, contanto que não haja sobreposição entre dois slots. Como exemplo, o canal -160 representa o canao C21 da grade ITU-T (192.1 THz). Este experimento considera  $m = 8$  para todos os canais (espaçamento de 50GHz).

Com o objetivo de demonstrar a operação das VONs, um conjunto de 40 canais modulados em 112 Gbps com *Dual Polarization Quadrature Phase Shift Keying* (DP-QPSK) foram conectados às portas ADD dos NEs 01 e 02. A topologia física, canais adicionados e removidos para ambas VONs são ilustrados na Figura 30.

```
% show config system von-ne roadm cross-connections |tab
```

VON ID	NODE ID	CC LABEL	CC CHANNEL	IN	OUT
1	1	D1-IN-D3-OUT-312	312	D1-IN	D3-OUT
		D1-IN-D3-OUT-328	328	D1-IN	D3-OUT
		D1-IN-D3-OUT-344	344	D1-IN	D3-OUT
		D1-IN-D3-OUT-360	360	D1-IN	D3-OUT
...					
1	2	D1-ADD-D1-OUT-312	312	D1-ADD	D1-OUT
		D1-ADD-D1-OUT-328	328	D1-ADD	D1-OUT
		D1-ADD-D1-OUT-344	344	D1-ADD	D1-OUT
		D1-ADD-D1-OUT-360	360	D1-ADD	D1-OUT
...					
1	3	D1-IN-D3-OUT-312	312	D1-IN	D3-OUT
		D1-IN-D3-OUT-328	328	D1-IN	D3-OUT
		D1-IN-D3-OUT-344	344	D1-IN	D3-OUT
		D1-IN-D3-OUT-360	360	D1-IN	D3-OUT
...					
1	4	D1-ADD-D1-OUT-312	312	D1-ADD	D1-OUT
		D1-ADD-D1-OUT-328	328	D1-ADD	D1-OUT
		D1-ADD-D1-OUT-344	344	D1-ADD	D1-OUT
		D1-ADD-D1-OUT-360	360	D1-ADD	D1-OUT
...					
2	1	D1-ADD-D1-OUT--8	-8	D1-ADD	D1-OUT
		D1-ADD-D1-OUT-24	24	D1-ADD	D1-OUT
		D1-ADD-D1-OUT-40	40	D1-ADD	D1-OUT
		D1-ADD-D1-OUT-56	56	D1-ADD	D1-OUT
...					
2	3	D1-IN-D3-OUT--8	-8	D1-IN	D3-OUT
		D1-IN-D3-OUT-24	24	D1-IN	D3-OUT
		D1-IN-D3-OUT-40	40	D1-IN	D3-OUT
		D1-IN-D3-OUT-56	56	D1-IN	D3-OUT
...					
2	4	D1-ADD-D1-OUT--8	-8	D1-ADD	D1-OUT
		D1-ADD-D1-OUT-24	24	D1-ADD	D1-OUT
		D1-ADD-D1-OUT-40	40	D1-ADD	D1-OUT
		D1-ADD-D1-OUT-56	56	D1-ADD	D1-OUT
...					

Figura 29 – Cross-conexões no ROADM por VON/NE. Fonte: [B].

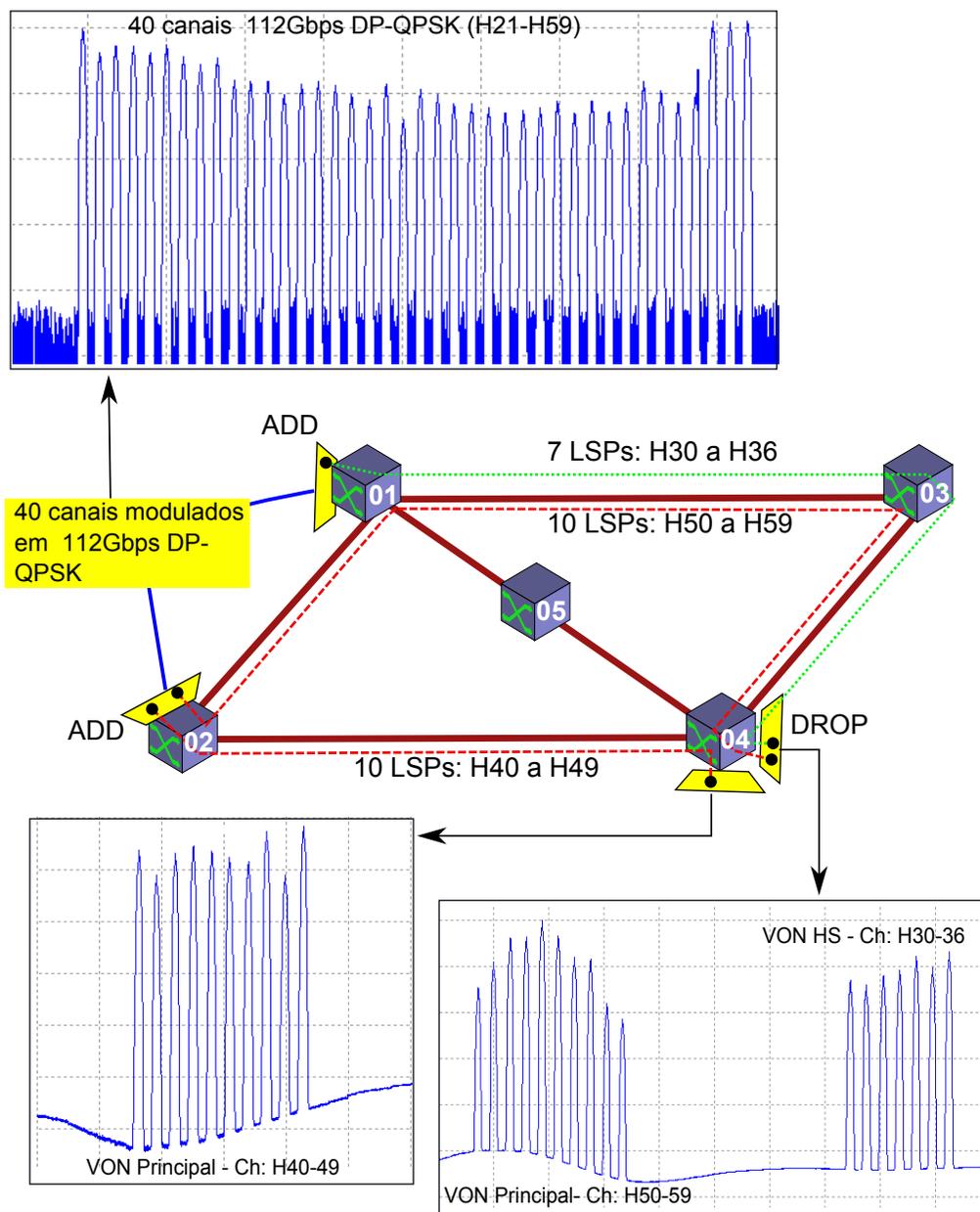


Figura 30 – Espectro óptico nos NEs 01, 02 e 04. Fonte: [B].

## 5.3 UC02 – Reconfiguração dinâmica de VON com base em políticas

No UC01 foi demonstrada a criação automática da VON principal, com mapeamento direto na topologia física da rede, com base em descoberta LLDP. Adicionalmente, é possível a instanciação de VONs estáticas a partir de arquivos XML.

Já no UC02, introduzido por M. Siqueira et al. [A], é validado o funcionamento de uma aplicação SDN denominada *Virtual Optical Network Configuration and Restoration SDN Application* (VONCR-APP), a qual é capaz de instanciar e manter VONs de forma dinâmica, com base nas seguintes informações por VON: (a) lista de pontos de ADD/DROP de sinais cliente; e (b) política de mapeamento da rede virtual na rede física. A aplicação é capaz de instanciar cada VON, alocando recursos e instanciando entidades de plano de controle para cada NE virtual de cada VON, podendo alterar o mapeamento de forma dinâmica caso ocorra mudança na lista de pontos de ADD/DROP, no estado da rede física (ex.: falha em enlaces ou NEs), ou na política de mapeamento.

A rede experimental, ilustrada na Figura 31c é composta de cinco NEs DWDM e o controlador T-SDN. Além da VON principal, criada de forma automática com base nas informações de descoberta de topologia, VONCR-APP permite a criação de VONs adicionais de forma automática, com base em arquivos XML contendo a descrição dos pontos de entrada (redes cliente), matriz de demanda e requisitos de redundância (políticas). A Figura 31a mostra duas VONs: *green* e *blue*, seus mapeamentos para a rede física, e respectivos TE-Links.

A Figura 31b ilustra os resultados esperados para a reconfiguração das VONs utilizadas como exemplo para o caso de uma falha no enlace que conecta os NEs 0 e 2. Os passos são descritos a seguir:

1. O sistema é inicializado e as VONs são instanciadas.
2. Uma falha, tal como um rompimento de fibra, ocorre no enlace que conecta os NEs 1 e 3.
3. Como as redes cliente são provisionadas com proteção, o tráfego é comutado para a rota alternativa.
4. O estado da falha é atualizado no modelo YANG do controlador T-SDN através de notificação NETCONF, de forma que a abstração em grafo que possui tal atributo é automaticamente atualizada.
5. Um evento é disparado no motor de políticas, que por sua vez, atravessa o grafo, analisando a situação da rede e as políticas, conforme detalhado na Figura 24. Neste caso,

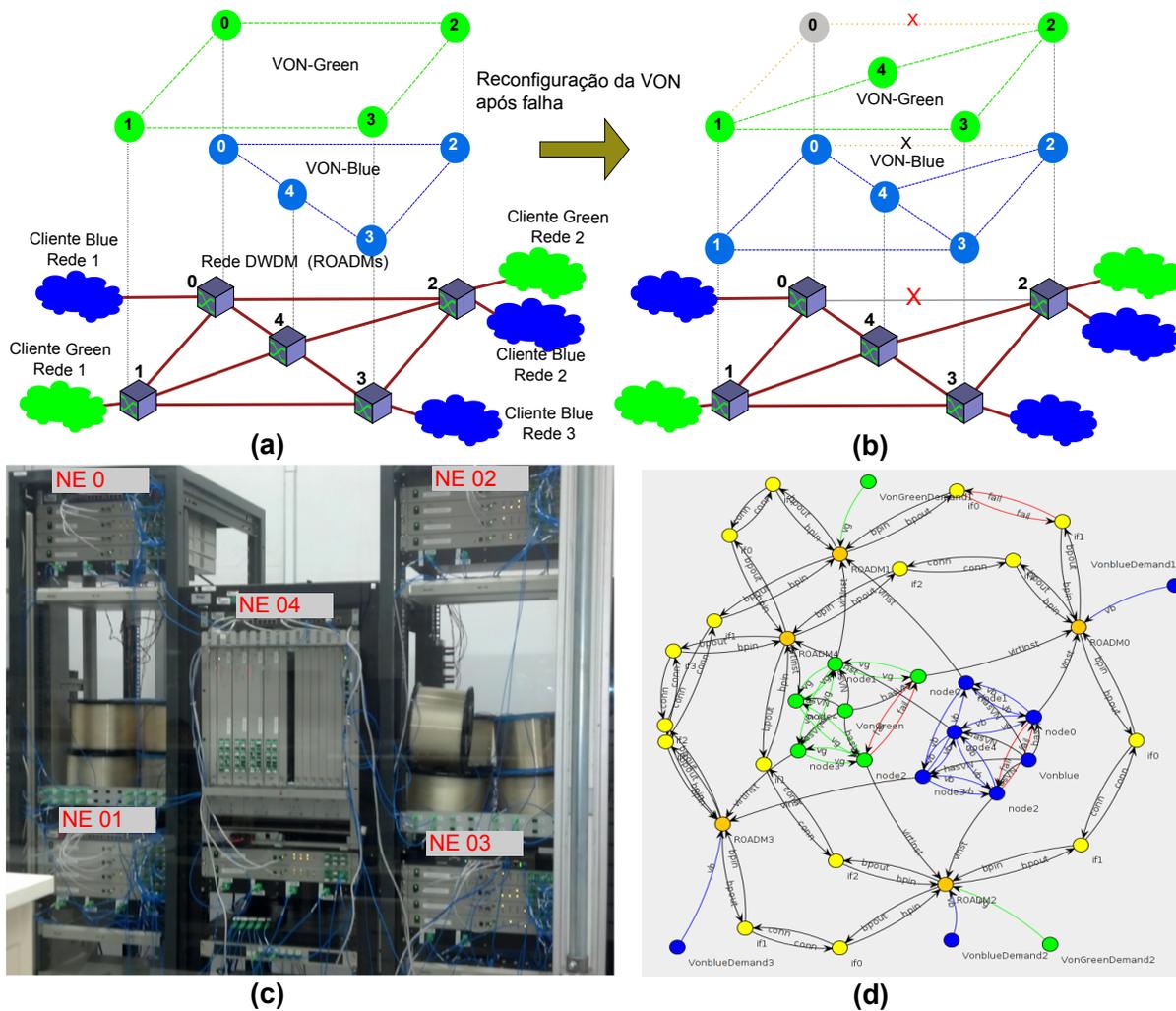


Figura 31 – Reconfiguração da VON após falha em enlace na rede física. Fonte: [A].

uma das políticas define que cada VON deve disponibilizar pelo menos dois caminhos alternativos para cada par de demandas.

6. Após a análise do grafo, as ações definidas pela política são executadas, dentre as quais, o acionamento da aplicação SDN VONCR-APP para reconfigurar as VONs, conforme ilustrado na Figura 31b.
7. A aplicação VONCR-APP adiciona novos recursos a cada VON, no T-NOS, com o objetivo de manter o requisito de rotas alternativas e disjuntas para cada par de redes cliente. Adicionalmente, novas instâncias de XCP e TE-Links são criadas, se aplicável.
8. No contexto deste caso de uso, é função do cliente de cada VON gerenciar comutação e re-roteamento de *lightpaths* após reconfiguração da VON.

A implementação da análise de situação através da análise de grafos foi realizada utilizando a API JUNG e inclui travessia de grafos e verificação de atributos. Para isso, um grafo abstrato da rede foi gerado, onde o ROADM é abstraído como um *switching fabric* genérico, dotado de um conjunto de interfaces. Desta forma, toda a complexidade dos componentes ópticos, como WSS, amplificadores, *multicast switch*, entre outros, é abstraída e padronizada, de forma que o mecanismo de análise de políticas possa ser reusado nos casos de mudança de arquitetura de NEs. A Figura 31d ilustra a visão abstrata do grafo, incluindo ambas VONs após a execução das ações de reconfiguração. A API Java JUNG foi utilizada para gerar a visualização a partir do grafo.

## 5.4 UC03 – Aplicação SDN para reconfiguração dinâmica do formato de modulação

A utilização de técnicas de processamento digital de sinais, aplicadas em receptores coerentes de transponders ópticos de alta taxa (Ex.: 100 a 1000 Gbps), permite que sejam implementados nos DSPs diferentes formatos de modulação, que podem ser mudados conforme as condições da rede (SCHMOGROW *et al.*, 2010). A visão holística viabilizada pela arquitetura T-SDN permite que o controle do formato de modulação dos transponders seja realizado somente em casos em que esta operação é realmente necessária, e de forma coordenada com a configuração dos demais equipamentos da rede óptica, como os ROADMs.

A Figura 32 ilustra o cenário do experimento [E], no qual um transponder com suporte a taxa de 448 Gbps com suporte a formato de modulação variável foi alinhado sobre uma rede óptica reconfigurável, controlada pelo T-SDNC. O sinal óptico é composto de 38 canais ópticos do tipo *Continuous Wave* (CW) com espaçamento de 100 GHz na grade DWDM do ITU-T, acoplados com o sinal do transponder flexível. A Tabela 1 apresenta as características do Transponder Flexível relativas a formatos de modulação.

Tabela 1 – Parâmetros do experimento com transponder flexível

Taxa (Gbps)	Form. de modulação	Larg. de Espectro	OSNR Mín.)
448	DP-16QAM	75GHz	29dB
448	DP-QPSK	150GHz	14dB

Os sinais foram transmitidos através da rede com ROADMs com grade flexível, interconectados por spans de 50 km de fibras ópticas, sendo que foram utilizados amplificadores EDFA para compensar perdas de enlaces e dos próprios ROADMs. Para emular uma degradação no sistema, foi acoplado *Amplified Spontaneous Emission* (ASE), utilizando-se um atenuador variável para mudar a degradação durante o experimento. O canal de 448Gbps foi

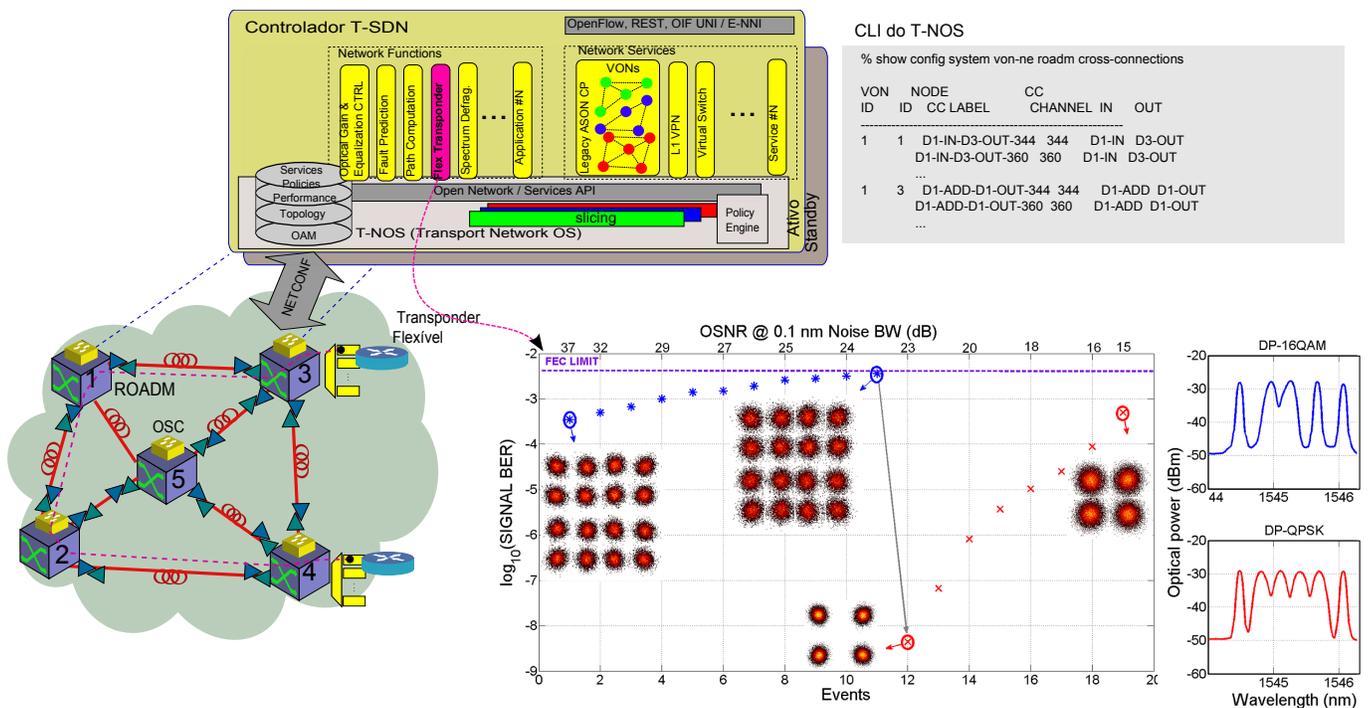


Figura 32 – Experimento com Transponder Flexível. Fonte: [E].

recebido através de um receptor coerente com suporte a duas polarizações, e a saída deste foi amostrada a 40-GS/s por um osciloscópio. A BER foi computada e processada *off-line* através de algoritmos para a recuperação da informação transmitida [E].

Conforme mostrado na Figura 32, no início do experimento o transponder flexível foi configurado com formato de modulação DP-16QAM, ocupando somente 75GHz de espectro. Com a degradação da OSNR, a aplicação SDN “*Flex Transponder*” atua na reconfiguração do transponder flexível para o formato de modulação DP-QPSK que requer valores menores de OSNR. Como este ocupa o dobro de espectro óptico (150 GHz), a aplicação coordena com a aplicação de desfragmentação do espectro para mover o canal vizinho para outra região, de forma que os ROADMs possam ser reconfigurados de forma automática para suportar o canal de 150 GHz.

Observa-se que a atuação da aplicação “*Flex Transponder*” deve ser controlada por políticas, de forma que esta somente deve ser acionada no caso de aplicações de controle de ganho de amplificadores ópticos e aplicações de equalização óptica não tenham sido efetivas em resolver a situação de falha ou degradação.

## 5.5 UC04 – Plano de controle SDN com suporte a NEs flexíveis

Este caso de uso, introduzido por M. Siqueira e C. E. Rothenberg [D], apresenta resultados obtidos através da implementação de uma aplicação SDN que implementa um plano de controle avançado, capaz de calcular rotas e estabelecer circuitos fim-a-fim na rede incluindo o grafo de conectividade interna de cada NE ao grafo global da rede. Desta forma, a aplicação é capaz de evitar bloqueio em nós com contenção, além de viabilizar a implementação de NEs flexíveis com suporte multicamadas.

Para validar o funcionamento da aplicação, foi concebido um NE flexível denominado *Software-defined Packet-optical Switching System* (SD-POSS). A arquitetura SD-POSS apresenta uma mudança de paradigma, já que, em vez de criar soluções multicamadas (Ex.: IP, OTN e DWDM), permite que a rede opere com uma única camada, possibilitando a eliminação de ineficiências, redução de custos e maior flexibilidade. Adicionalmente, a proposta já suporta diferentes funcionalidades que permitem a virtualização das redes e serviços. A solução proposta é composta por dois componentes principais, detalhados nesta seção: (i) arquitetura de referência para um equipamento com suporte à comutação multicamadas que objetiva permitir a implementação de equipamentos virtuais, seguindo o conceito NFV, e (ii) arquitetura de controle SDN utilizando abstração da topologia dos elementos de rede e da rede com recursividade, através de grafos direcionais.

### Arquitetura do SD-POSS

A Figura 33 ilustra a arquitetura de referência proposta para o elemento de rede que viabiliza a convergência das camadas óptica, OTN e IP. Na arquitetura proposta, todas as interfaces externas do elemento de rede (NE) são conectadas a um OXC (*Optical Cross Connect*) com  $K \times K$  portas, que distribui tais interfaces para blocos que compõem o NE. Tais blocos incluem módulos WSS (*Wavelength Selective Switch*) que compõem um ROADM com  $N$  direções, onde  $N < 20$ , incluindo um bloco de *ADD/DROP*. Adicionalmente, são conectados os lados cliente de interfaces de linha, baseadas em soluções de ASIC, FPGA de alta capacidade ou processadores de rede (*Network Processors*) que permitam a implementação do conceito NFV a nível das placas de linha, ou seja, as placas devem poder operar em modo IP/OpenFlow, em modo OTN *Switch*, ou ambos. Tais placas devem conectar-se a um conjunto de matrizes agnósticas, normalmente implementadas através da comutação de células, que permitem tanto a comutação de pacotes como de quadros OTN, através de adaptação feita nas placas de linha. Adicionalmente, observa-se que as matrizes externalizam algumas interfaces para o OXC, de forma que seja possível a criação de elementos de rede virtuais, distribuídos, ou seja, placas de linha espalhadas por diferentes sítios interconectadas através

de matrizes remotas.

Outros componentes da arquitetura podem ser os transponders multífluxos, os quais podem ser implementados no mesmo hardware das placas de linha agnósticas, ou de forma separada, conforme ilustrado na Figura 33. Finalmente, observa-se que o OXC possui algumas linhas em *loop* do lado cliente, de forma que seja possível a realização do mapeamento de sinais de um tipo em outro, de forma hierárquica, como por exemplo o mapeamento do sinal que sai de uma das placas de linha em uma porta de *ADD/DROP* do ROADM. Esta proposta passa a ser viável na medida em que OXCs baseados em tecnologia 3D-MEMS estão sendo disponibilizados no mercado com grande quantidade de portas (Ex.: 320x320) com relação custo-benefício não proibitiva.

Avalia-se que a proposta de plataforma de hardware modularizada do SD-POSS permitirá a implementação do conceito de NFV para a criação de elementos de comutação e transmissão multicamadas de alta capacidade. O objetivo é permitir que, através de modificações somente a nível de modelo, seja possível a implementação das seguintes funções de rede, de forma exclusiva ou em qualquer combinação de tais funções de forma simultânea: comutação de comprimentos de onda e/ou de faixas de espectro óptico, comutação de sub-lambda, incluindo diferentes níveis de ODU, comutação de pacotes e implementação de matriz de comutação agnóstica (ODU e pacotes) geograficamente distribuída. Adicionalmente, é proposta uma abordagem baseada no conceito SDN para a orquestração dos elementos e redes virtuais, bem como para a criação de um plano de controle baseado em grafos direcionais para o estabelecimento dos fluxos multicamadas, com suporte a recursividade entre os modelos dos equipamentos e da rede como um todo, além de implementar de forma nativa o suporte a virtualização de rede através de *network slicing* e virtualização de planos de controle legados.

## Implementação do plano de controle SDN para redes SD-POSS

Para implementar o suporte a virtualização e o conceito de NFV, idealmente deveria ser possível instanciar elementos de rede virtuais, com diferentes funções e suporte a isolamento de recursos em um mesmo hardware SD-POSS. Adicionalmente, a programação dos mecanismos de encaminhamento poderia ser padronizada, como por exemplo usando abordagens similares ao *pipeline* utilizado em processadores de rede (COMER, 2004), independentemente dos tipos de blocos internos utilizados. Desta forma, a implementação dos conceitos de virtualização de rede e NFV foi realizada em nível do sistema operacional de rede, permitindo a implementação dos elementos virtuais em um mesmo SD-POSS, além de permitir a implementação de *network slicing* através da separação de cada rede virtual em diferentes chassis virtuais.

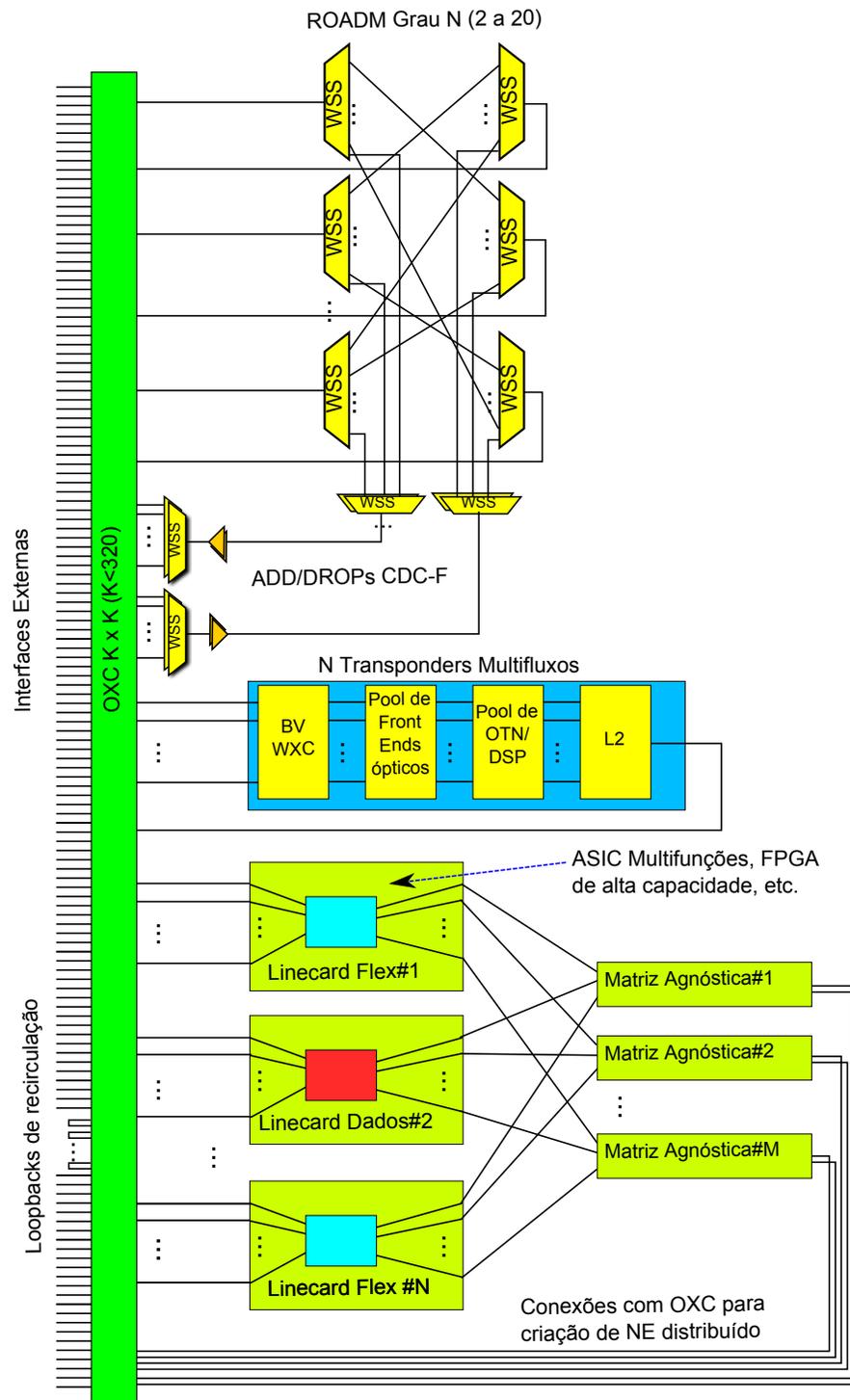


Figura 33 – Arquitetura proposta para elemento de rede SD-POSS. Fonte: [D].

O modelo recursivo de elementos de rede e da rede, criado utilizando-se a linguagem YANG, permite que seja realizada uma criação automática da topologia da rede no controlador, através de LLDP, conforme descrito por M. Siqueira et al. [B]. Desta forma, através de

transformação XML gera-se um grafo abstrato no formato *graphml* em que cada elemento de rede é representado por um comutador de uso geral, equipado com um conjunto de interfaces, o qual é utilizado para cálculo do caminho na rede. Tal grafo é composto com os grafos que detalham a parte interna de cada elemento de rede para que seja realizada a travessia já considerando os fluxos a serem configurados dentro de cada SD-POSS.

Para o cálculo de caminhos no cenário com integração de camadas, foi implementado um protótipo do mecanismo para cálculo de caminhos através de travessia de grafos utilizando-se a API de grafos JUNG, incluindo travessia de grafos de redes multicamadas, formadas por elementos do tipo SD-POSS, já com suporte a verificação de atributos como comparação de “tipo de demanda” x “tipo de aresta”.

## Resultados experimentais

A Figura 34(a) ilustra a rede de referência (baseada no mapa do *backbone* disponibilizado no sítio *web* da RNP) e 34(c) o grafo desta, incluindo a parte interna de cada SD-POSS, bem como o caminho calculado para determinado fluxo de uma demanda de um lambda entre os NEs SPO e SSA. A API JUNG foi utilizada para gerar as visualizações dos grafos. A Figura 34(b) ilustra o grafo do SD-POSS de RJO, incluindo as matrizes elétrica e óptica. Como a demanda é de um lambda, a figura ilustra o caminho escolhido internamente ao elemento passando pelo ROADM.

Para a avaliação de desempenho e escalabilidade da solução de cálculo de caminhos foi utilizado um computador equipado com processador Intel i5, com 8GB de RAM, rodando Linux Ubuntu 12.04LTS. A Tabela 2 apresenta um resumo dos resultados relativos ao tempo de cálculo de rotas, incluindo: (i) uma análise de desempenho relativa ao tempo de cálculo de travessia do grafo completo para um algoritmo que segue as arestas do grafo, compatíveis com a demanda (lambda), utilizando um mecanismo de *crankback* (T1), que realiza uma busca em árvore, não atravessando a mesma aresta mais que uma vez. O algoritmo é limitado ao maior diâmetro da rede, que no caso é 42 saltos, considerando 7 saltos multiplicado por 6 vértices internos em cada elemento de rede. Observa-se que o tempo médio para o cálculo do caminho para diversas quantidades de saltos é aproximadamente 500ms; (ii) a coluna T2 apresenta uma análise de desempenho relativa ao tempo de cálculo de travessia do grafo utilizando o algoritmo *Dijkstra* simples, o qual é aplicado sobre um subgrafo que contém somente as arestas e vértices compatíveis com a demanda, incluindo os vértices e arestas internos aos NEs. Observa-se que o tempo médio para o cálculo do caminho para diversas quantidades de saltos é aproximadamente 50ms, ou seja, dez vezes menor que o algoritmo de *crakback*; (iii) já a coluna T3 apresenta os tempos para o cálculo da rota utilizando-se um subgrafo simplificado, que não contém a parte interna dos NEs, obtendo-se tempo médio entre 35 e

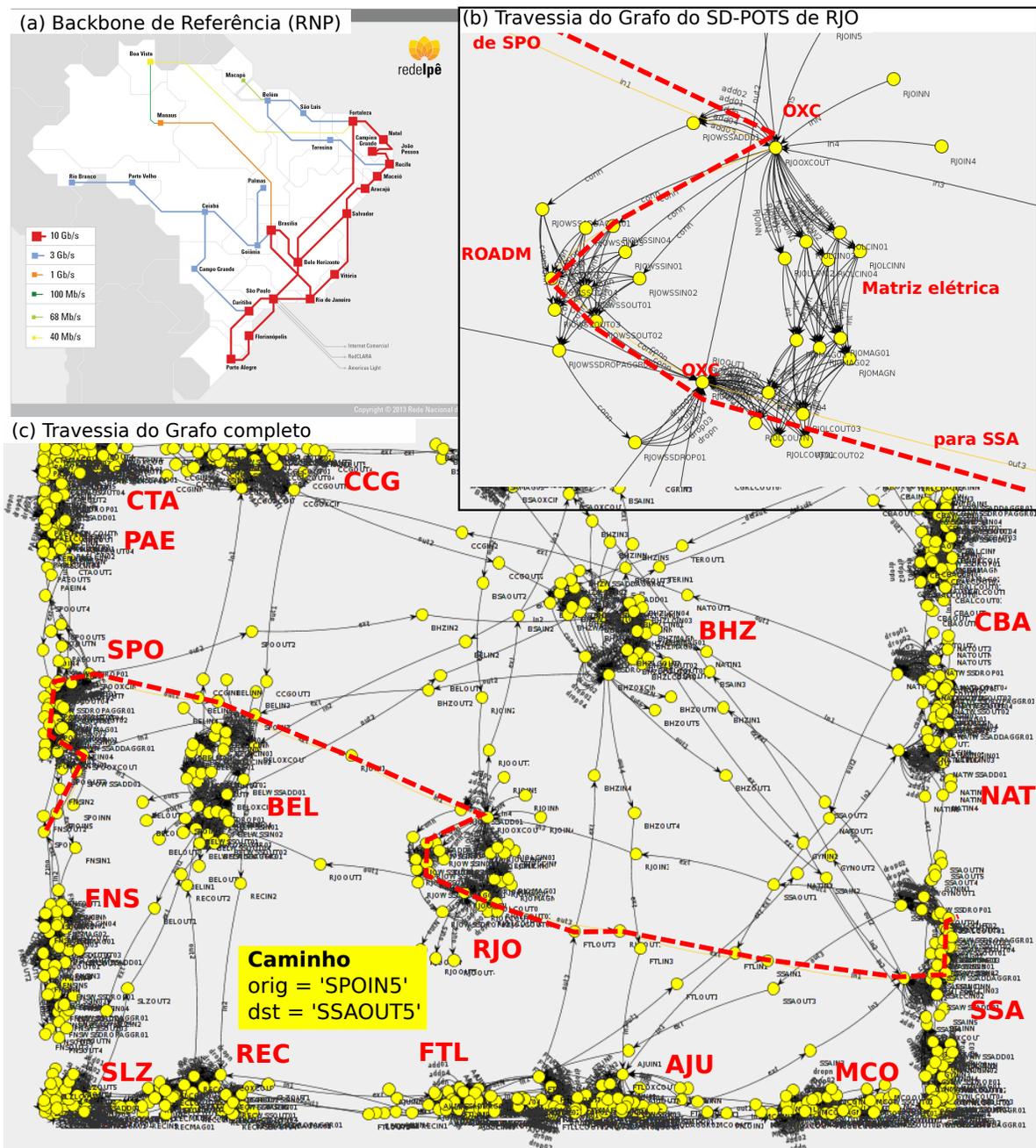


Figura 34 – Grafo completo ilustrando a rede de referência e o caminho calculado entre SPO e SSA. Fonte: [D].

39 ms.

Observa-se que quantidade de arestas na travessia não aumenta significativamente com o número de saltos nem com o diâmetro do algoritmo, sendo mais dependente da complexidade na topologia da rede. No caso, o cálculo da rota considerando todos os vértices e arestas internas ao nó levou aproximadamente 15ms a mais que o cálculo com a topologia

completa, quando o algoritmo Dijkstra foi utilizado. Adicionalmente, foram realizados testes de escalabilidade com cálculo de caminhos para crescimento com clusters auto-similares da rede de referência ( $N$  redes de referência interconectadas seguindo a mesma topologia de conectividade de nós da mesma), com baixo grau de conectividade (até 3 em cada cluster), para até 1200 vértices, o tempo de cálculo continua em torno de 50 ms.

Tabela 2 – Análise de desempenho para o cálculo de caminhos

Origem->Rota->Destino	T1(ms)	T2(ms)	T3(ms)
PAE->FNS	523	50	39
PAE->FNS CTA->SPO->BHZ->FTL->SLZ->BEL	488	52	35

## Avaliação da factibilidade de construção do SD-POSS através de análise de CAPEX

A realização da análise de CAPEX da arquitetura proposta pode ser realizada utilizando-se como referência alguns trabalhos que apresentam custos comparativos de equipamentos IP, OTN e DWDM. Uma proposta de re-arquitetura das redes IP foi apresentada por Das *et al.* (2013), a qual é baseada na realização de *bypass* em comutadores TDM baseados em OTN sempre que possível, criando uma topologia lógica *full mesh* entre roteadores IP de borda, utilizando um controle de roteamento unificado baseado em SDN. A proposta inclui uma análise detalhada de custos, que mostrou que a solução proposta oferece uma redução de CAPEX de 60% em relação a uma rede IP sobre DWDM tradicional e de 50% comparando-se com uma solução que utiliza IP com *bypass* óptico, já que é obtida uma grande economia de portas ópticas nos roteadores, somada com a redução de interfaces devido à unificação dos mecanismos de proteção das diferentes camadas. Os resultados foram validados para três matrizes de tráfego diferentes. Os custos relativos apresentados mostram que o valor de placas de linha em comutadores OTN é similar a equipamentos com comutação puramente óptica. Já as placas de linha em roteadores podem alcançar custos de até 8 vezes o custo de portam em equipamentos com comutação óptica.

No entanto, se forem consideradas demandas mais altas, requerendo a implementação em larga escala de interfaces de 100Gbps, pode-se considerar que esta diferença deverá diminuir. Conforme previsão apresentada por Inniss (2013), o custo de transponders 100Gbps com alcance compatível com redes metropolitanas (<800Km) deve reduzir para a faixa entre 20 a 30k USD em meados de 2016. Já os transponders de longa distância teriam o dobro do valor. Desta forma, um equipamento DWDM equipado com 50 placas de linha de 100G (5Tbps) do tipo Metro teria um custo de 1M USD. Desta forma, pode-se prever que a diferença na proporção de preços entre roteadores IP e os demais tipos de equipamentos, apresentada

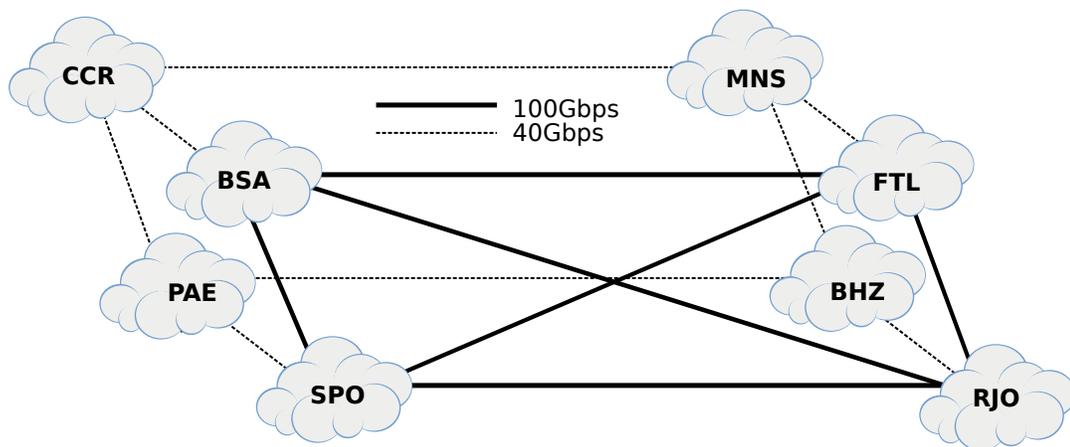


Figura 35 – Demandas fictícias (Ex.: entre *data centers* ou tráfego Internet)

por Das *et al.* (2013), deverá reduzir, já que um grande componente do preço será devido aos módulos ópticos, ficando uma menor parcela associada ao software embarcado e ao hardware.

Para a análise a ser realizada no contexto desta tese, serão considerados custos relativos, considerando um OXC com quantidade elevada de portas (custo=100 $up$ , onde  $up$  é uma unidade fictícia de preço), sistema DWDM/ROADM equipado com 50 transponders de 100Gbps (custo=3000 $up$ ), comutador OTN (custo=2800 $up$ ) e roteador (custo=6000 $up$ ), ambos equipados com 50 portas de 100Gbps. Observe que foi assumida a proporção de preço entre um equipamento DWDM/ROADM e um roteador IP, ambos com 50 portas de 100Gbps, de 2x, sendo este um fator muito menor ao utilizado no estudo realizado por Das *et al.* (2013).

Para realização da análise de CAPEX relativo utilizou-se uma variação da topologia da RNP apresentada, incluindo nós e enlaces adicionais, contendo 22 sites, 44 conexões de fibra óptica entre sites, 8 demandas de 40Gbps e 6 demandas de 100Gbps entre diferentes sites. As demandas (interesse de tráfego) são ilustradas na Figura 35. A rede foi dividida em três regiões geográficas, de forma que esta deveria ser dimensionada para tolerar quaisquer combinações de falhas duplas de enlaces nas três regiões (totalizando até 6 falhas, de forma simultânea).

Através do grafo da rede, tendo os enlaces modelados como arestas e os nós modelados como vértices, foi implementado um protótipo de uma ferramenta de dimensionamento de redes, utilizando a API JUNG, que realiza o roteamento das demandas, incluindo rota principal e de proteção para as demandas de 100G (sobrevivência do tipo  $1 + 1 + R$ ) e somente uma rota para cada demanda de 40G ( $1 + R$ ). A ferramenta permite a simulação das falhas duplas por região, conforme requisito de sobrevivência, re-roteando as demandas e dimensionando os enlaces conforme necessário. Desta forma, considerando uma rede IP sobre DWDM (interfaces integradas nos roteadores), ou com OTN *Switches*, equipados com

interfaces DWDM de longa distância, seriam necessárias 248 interfaces 100G, totalizando um custo de 9.920 $up$  para a rede construída com IP sobre óptica. Caso a rede seja projetada utilizando comutadores OTN em todos os sites, o custo relativo a tais comutadores seria de 4.712 $up$ , porém seria necessário adicionar 8 roteadores IP nos sites com demandas, somando um total de 38 interfaces 100G para atender às demandas, totalizando 6.232 $up$ .

Para a estimativa do custo relativo da arquitetura SD-POSS aplicada à rede em questão, considera-se um custo fixo dos OXCs (um por site), totalizando 2200 $up$ . Adicionalmente, será considerado que as placas de linha 100G, mesmo que com funcionalidades IP habilitadas possuem o mesmo custo que as com comutação de circuitos, já que, conforme abordado por Das *et al.* (2013), pode-se obter menor custo utilizando-se uma arquitetura SDN, com menor complexidade de software embarcado nos equipamentos. Adicionalmente, observa-se que, devido à versatilidade da solução com OXCs, não é necessário que o tráfego passe por uma placa de pacotes e depois por uma OTN no mesmo sítio, podendo ser mapeada no próximo. Desta forma, o custo da solução seria o equivalente a 6.912 $up$ . A Tabela 3 apresenta um resumo dos valores relativos, mostrando que a arquitetura SD-POSS apresenta uma redução de aproximadamente 30% comparada a uma solução IP sobre DWDM, porém sendo em torno de 10% mais custosa que a solução equivalente à apresentada por Das *et al.* (2013). No entanto, a arquitetura SD-POSS apresenta inúmeras vantagens oferecidas principalmente pela flexibilidade alcançada através da utilização do OXC.

Tabela 3 – Comparação de custos relativos entre a arquitetura SD-POSS, com soluções baseadas em IP e/ou OTN

Arquitetura	Valor ( $up$ )	(%)
IP sobre DWDM convencional	9.920	100%
IP nas bordas, com comutação OTN (DAS <i>et al.</i> , 2013)	6.232	62.8 %
SD-POSS	6.912	69.6 %

## 5.6 UC05 – Ajuste dinâmico de políticas de sobrevivência através de planejamento em operação

Em redes de transporte com suporte ao plano de controle GMPLS, a utilização de diferentes técnicas de restauração combinadas ou não com técnicas de proteção, permite a manutenção dos serviços em caso de uma ou mais falhas, com suporte à diferenciação de serviços para as demandas, dependendo da estratégia de sobrevivência de cada demanda.

Estratégias de sobrevivência incluem proteção 1 + 1, 1 : N, re-roteamento de LSPs, 1 + R, 1 + 1 + R e *shared mesh*. Cada estratégia de sobrevivência possui características

diferentes quanto ao dimensionamento da rede. Como exemplo, enquanto a estratégia do tipo  $1 + 1 + R$  requer o dobro da capacidade pré-reservada, além de banda residual para re-roteamento em caso de falha, a estratégia *shared mesh* já é mais eficiente, permitindo que várias demandas utilizem recursos de restauração compartilhados.

Desta forma, a rede deve ser planejada e dimensionada, conforme previsão de demandas e estratégia de sobrevivência desejada para cada demanda, bem como considerando qual ou quais camadas serão responsáveis por quais estratégias de sobrevivência (Ex.: rede óptica fixa e restauração na camada OTN).

Caso as demandas ativadas na rede não sejam as mesmas planejadas, em situações de falha algumas demandas sofrerão bloqueio na restauração por falta de recursos nos enlaces. Desta forma, este caso de uso consiste na avaliação de uma aplicação SDN, denominada *SDN - In Operation Network Planning* (S-IONP), que opera em malha fechada com o controlador SDN, podendo-se realizar “planejamento em operação”, utilizando técnicas adaptativas ou cognitivas para alterar a estratégia de sobrevivência das demandas de forma a não haver bloqueio. A aplicação suporta estratégias de sobrevivência dos tipos  $1 + 1$ ,  $1 + R$  e  $1 + 1 + R$ .

Para a validação da aplicação S-IONP utilizou-se uma topologia de uma operadora fictícia, com cabos ópticos interligando as principais capitais brasileiras, conforme ilustrado na Figura 36. Inicialmente, a aplicação é utilizada em modo planejamento *offline*, de forma que o dimensionamento dos enlaces seja realizado de forma que as demandas possam ser atendidas.

A Figura 37 ilustra a camada de núcleo da rede IP a ser construída, a qual é formada por oito roteadores conectados em uma topologia em formato de um cubo. Este *backbone* IP é o principal gerador de demandas para a rede de transporte de longa distância, a qual será construída com tecnologias de comutação de lambdas (ROADMs) e comutação de ODUs (OTN *Switch*).

Desta forma, é necessário criar uma topologia de OTN *Switches* interconectados através da rede de ROADMs, na qual, cada conexão de 100Gbps entre OTN *Switches* utiliza um conjunto de cabos ópticos, interconectados através de conexões a serem realizadas pelos ROADMs.

A Figura 38 ilustra a rede de OTN *Switches* com suas interconexões realizadas sobre a rede de cabos ópticos e de ROADMs. Observe que cada interconexão entre OTN *Switches* é realizada através de um ou mais cabos ópticos, que, em alguns casos específicos, são compartilhados por mais de uma conexão entre OTN *Switches*.

Conforme ilustrado na Figura 36, a rede foi dividida em três regiões. Os requisitos de sobrevivência da rede em questão são:

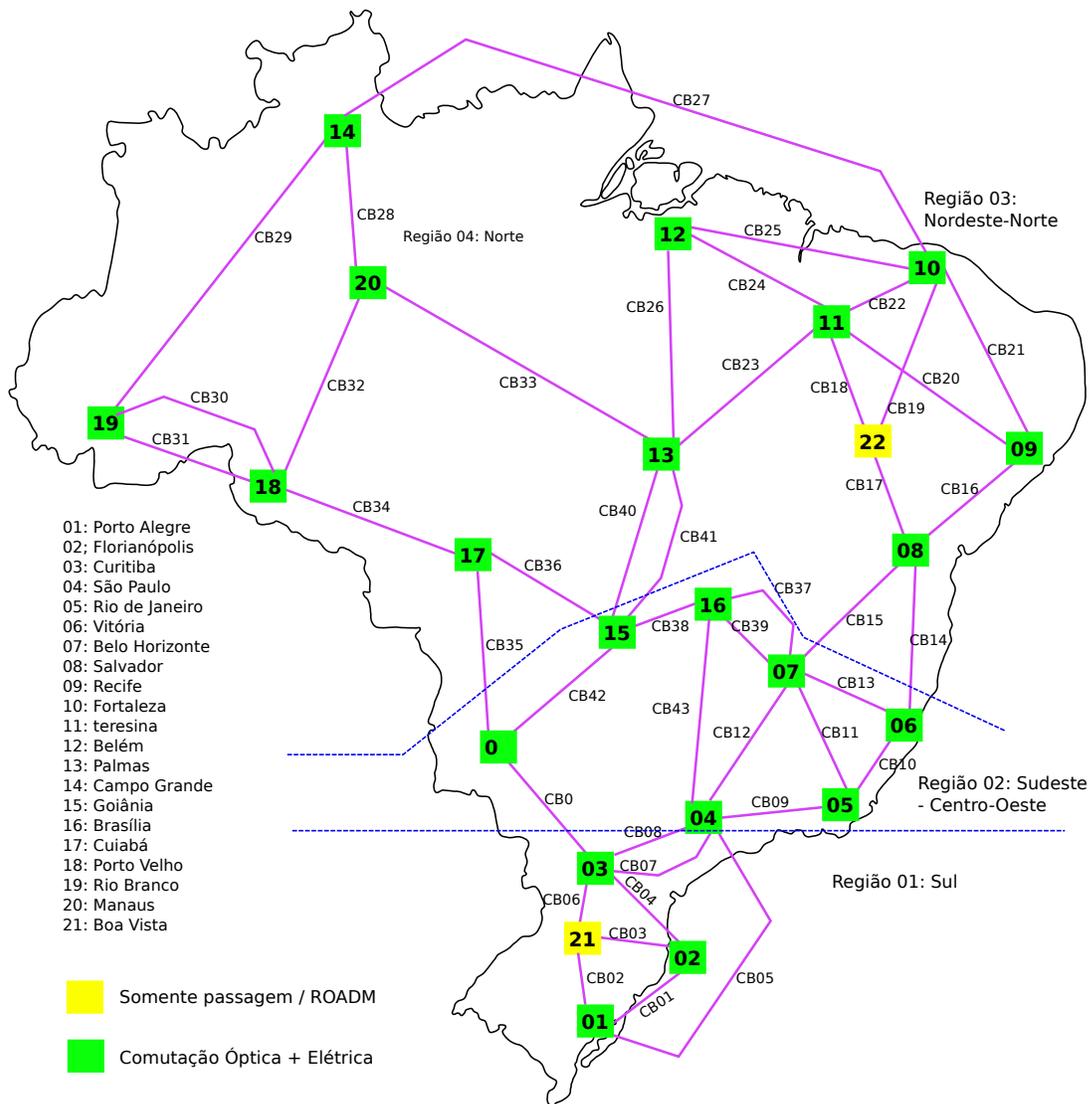


Figura 36 – Topologia de sites de cabos ópticos de referência

- Suporte a uma falha dupla de enlace por região, de forma simultânea. Desta forma, a rede deve suportar seis falhas simultâneas em enlaces.
- Demandas com terminação em RJO ou FTL possuem sobrevivência do tipo  $1 + 1 + R$ , onde a restauração só é ativada em caso de falha dupla que afete o caminho principal e a proteção.
- As outras demandas possuem sobrevivência do tipo  $1 + R$ .

As topologias física e lógica, as demandas e os requisitos de sobrevivência das demandas são passados à ferramenta S-IONP, que é utilizada para realizar uma primeira análise de sobrevivência da rede. A análise é realizada utilizando-se uma técnica de Monte Carlo, na qual

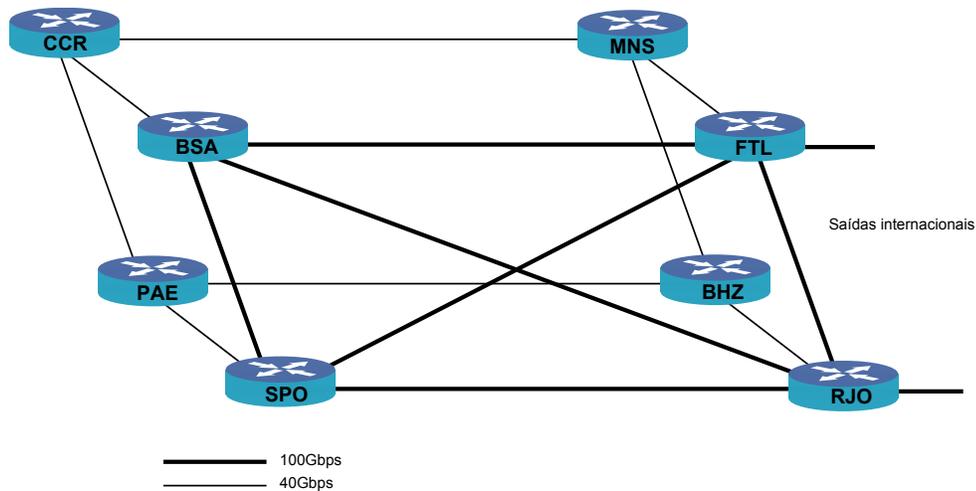


Figura 37 – Topologia do *Backbone* IP (demandas)

diversas combinações de falhas são geradas, verificando a capacidade da rede em rerrotear as demandas. As falhas são geradas de forma aleatória até que o dimensionamento da rede estabilize. Tal técnica foi utilizada para viabilizar o dimensionamento de redes com mais de 5 regiões e mais de 30 sítios, já que a avaliação de todas as possibilidades de falha pode requerer um tempo muito grande da ordem de vários anos, utilizando computadores disponíveis atualmente.

A ferramenta gera alertas no caso da topologia proposta não atender ao requisito de falhas nas diferentes regiões, gerando particionamento de demandas. A seção B.1 do Anexo B apresenta um exemplo de log saída textual da aplicação para a topologia em análise.

Desta forma, observou-se que o OTN *Switch* de CCR ficou isolado no caso de falha dupla nos cabos 0 e 42 da região 2, em conjunto com o cabo 35 da região 3, afetando de forma simultânea os enlaces 34, 33, e 30. Desta forma, assumindo que há disponibilidade de um novo cabo óptico entre SPO e CGR, este é adicionado à topologia, conforme ilustrado nas Figuras 39 (CB44) e 40 (Enlace 37).

A seção B.2 do Anexo B apresenta um resumo do log da aplicação S-IONP incluindo as etapas de roteamento inicial das demandas, incluindo caminhos de proteção, bem como simulação de falhas duplas em cada região, de forma simultânea, de forma a dimensionar as capacidades necessárias para restauração. A Figura 41 ilustra o grafo gerado como resultado do dimensionamento da rede em questão.

O funcionamento da aplicação em malha fechada com o controlador T-SDN é muito similar ao dimensionamento inicial da rede, no entanto, não altera a capacidade dos enlaces, mas sim o tipo de mecanismo de sobrevivência das demandas, caso haja alguma situação de

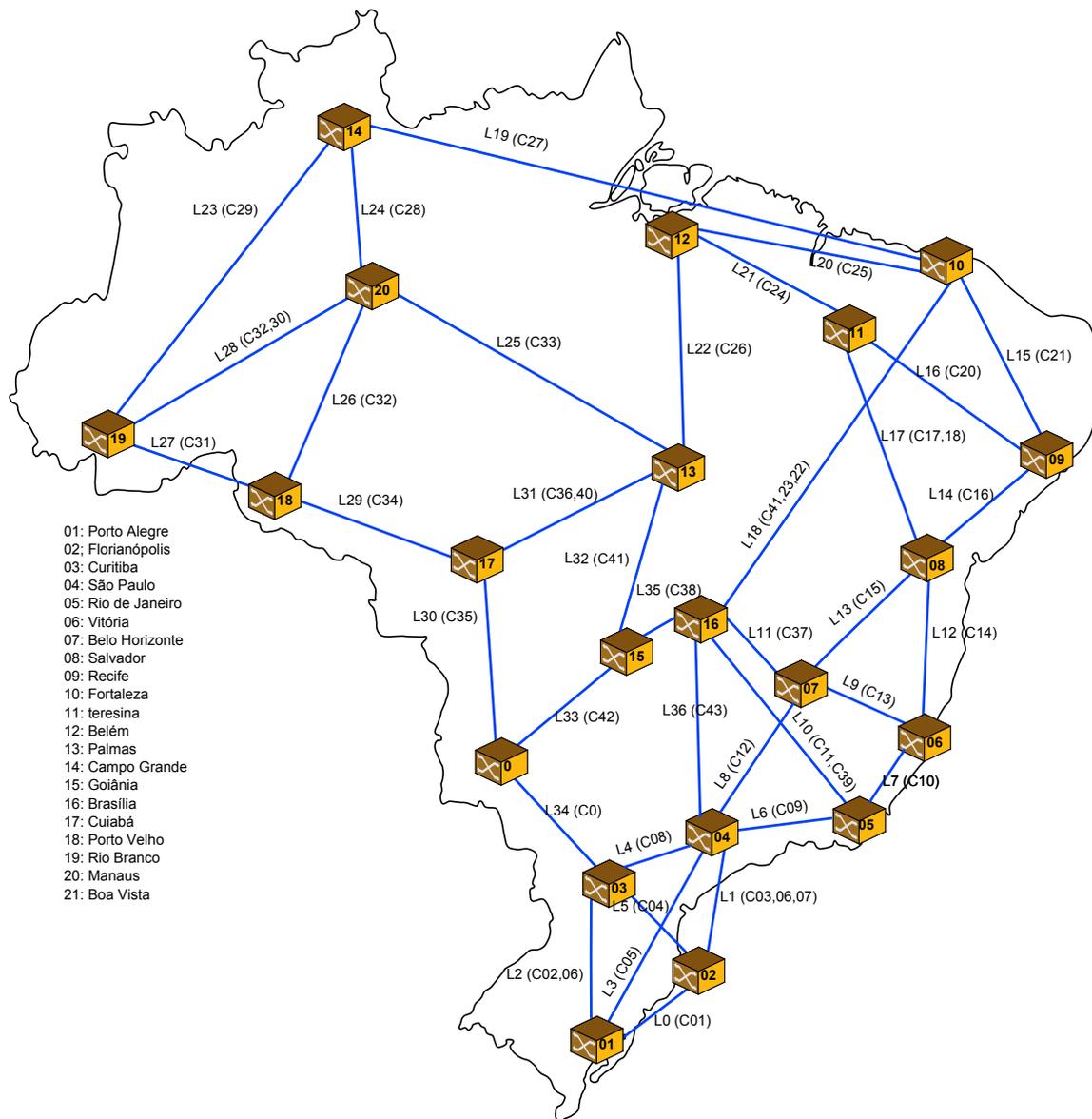


Figura 38 – Topologia da rede de OTN *Switches*

falha que causar bloqueio, conforme a política definida.

O “planejamento em operação” pode utilizar técnicas adaptativas ou cognitivas para alterar a estratégia de sobrevivência das demandas de forma a não haver bloqueio, bem como informações de histórico de falhas e SLA contratado dos serviços como entrada.

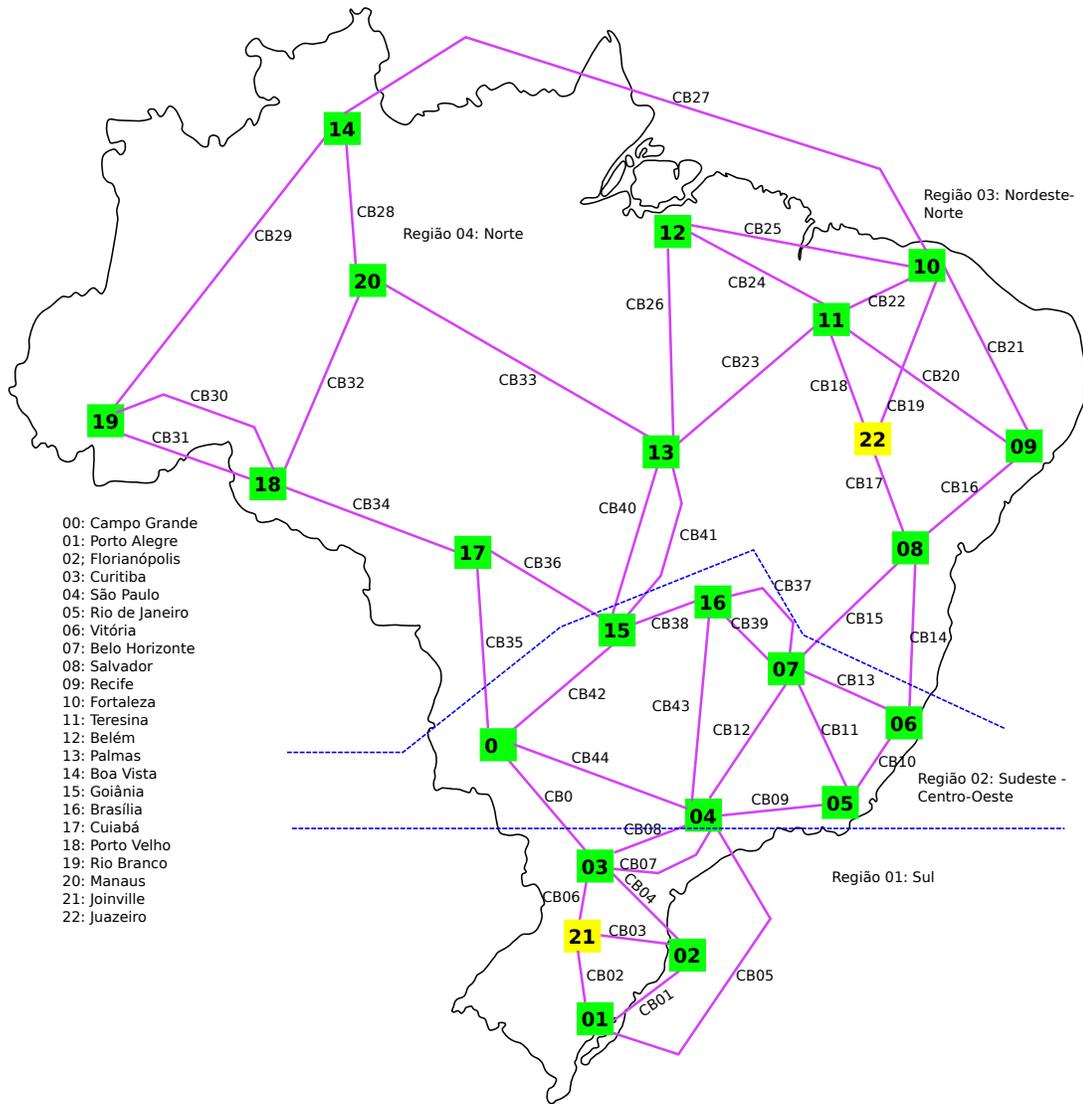


Figura 39 – Topologia de cabos ópticos readequada

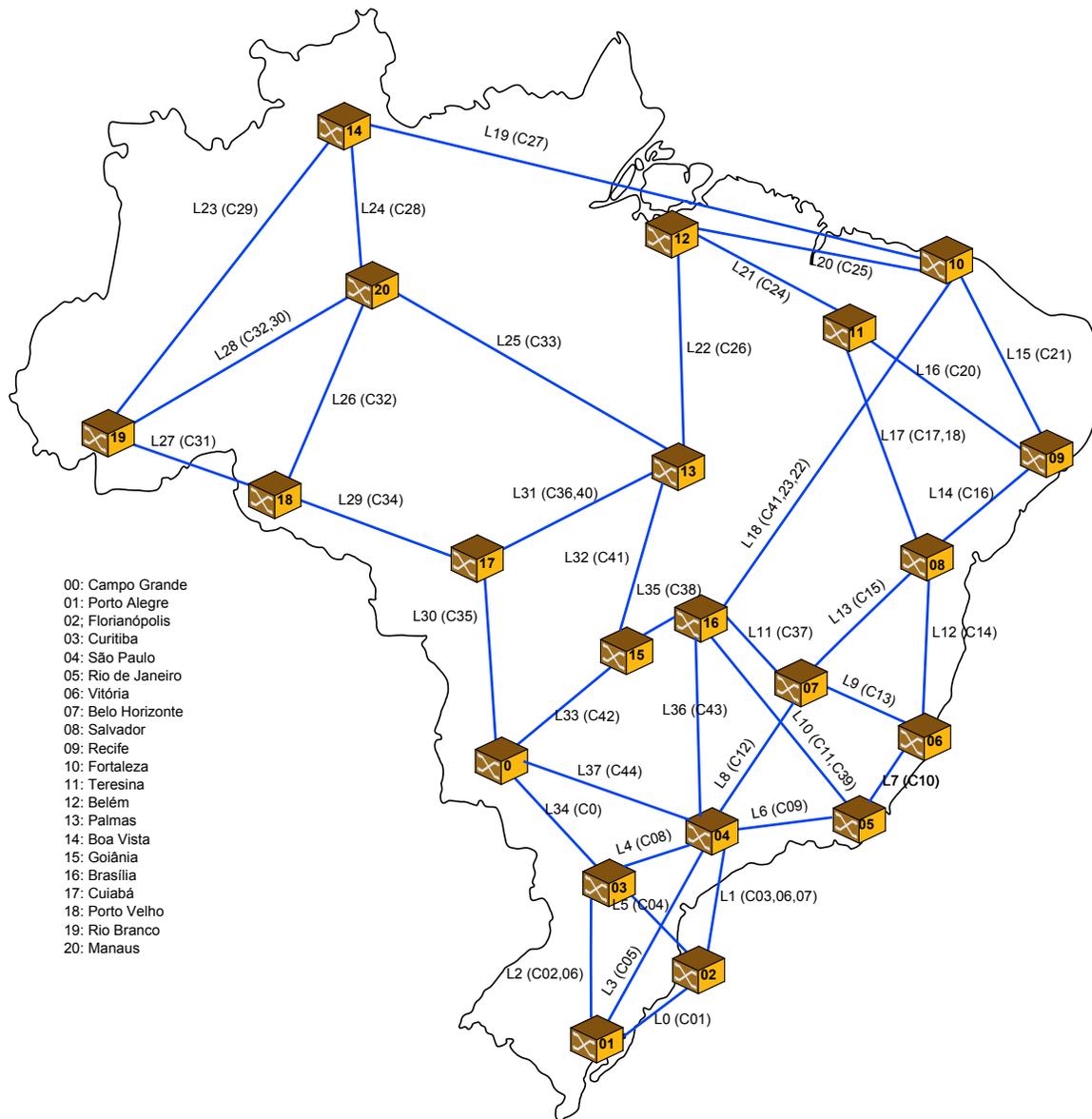
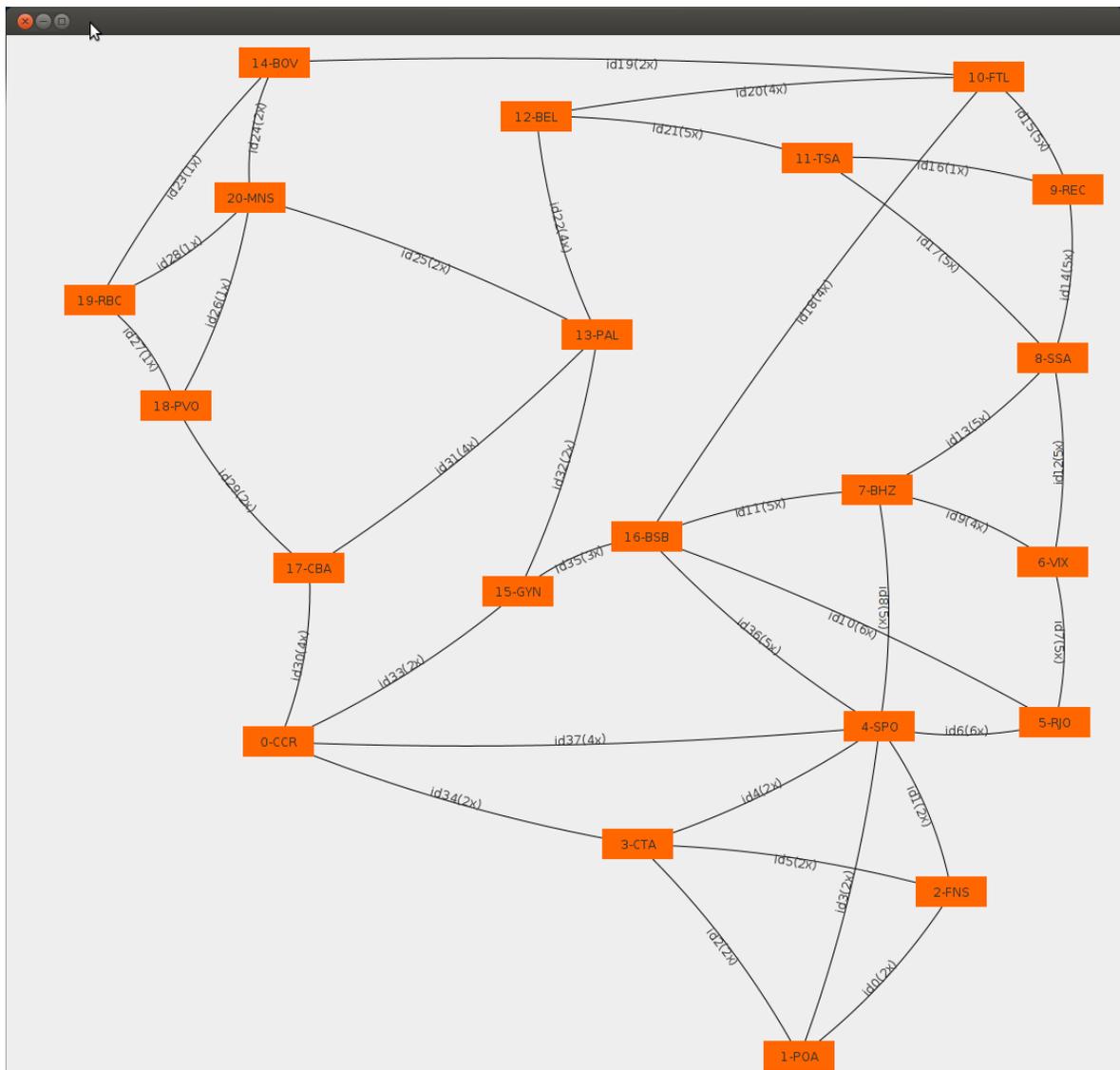


Figura 40 – Topologia da rede de OTN *Switches* readequada

Figura 41 – Grafo da rede gerado pela ferramenta *Network Planner*

## 5.7 UC06 – Ajuste fino de políticas locais via planejamento automatizado

O autor desta tese colaborou com E. Magalhães et al. [Q] na implementação de uma aplicação SDN para equalização óptica de redes com ROADMs, considerando os *lightpaths* fim-a-fim, de forma que o resultado final alcance o melhor desempenho possível. No entanto, um dos aspectos questionados quanto a aplicações SDN que atuam de forma reativa para o controle de funções vitais da rede é quanto à sua confiabilidade, devido à centralização, mesmo que lógica, da aplicação.

Com o grande aumento de capacidade viabilizado pela tecnologia DWDM, cada vez mais as redes ópticas de transporte possuem requisitos mais estritos com relação a confiabilidade. Desta forma, é fundamental que possam operar sem interrupção no caso de perda de conectividade entre NEs e o plano de gerência, ou mesmo em situações de falha no plano de controle. Desta forma, a arquitetura T-SDN proposta deve atender a este requisito, de forma que funções essenciais para a operação da rede sejam implementadas de forma distribuída nos elementos de rede, sendo altamente desejável que funcionem mesmo com situação de falha no canal de supervisão e/ou plano de controle.

Algumas destas funcionalidades poderiam ser implementadas tanto pelo controlador T-SDN como pelos NEs, de forma que o controlador implemente um conjunto mais rico de funcionalidades, se beneficiando do poder computacional disponível no controlador, bem como informações globais sobre a rede (com por exemplo a aplicação de equalização global, proposta por E. Magalhães et al. [Q]), já o NE poderia implementar um subconjunto de funcionalidades, suficientes para manter a rede operando no caso de falha nas comunicações entre NEs e o controlador T-SDN.

Com o objetivo de demonstrar este conceito, foi implementado um mecanismo no qual os ROADMs são capazes de realizar equalização de potência óptica forma sub-ótima, com base em instruções enviadas previamente pelo controlador T-SDN. Tais instruções são criadas por uma aplicação SDN que realiza simulações do tipo “what-if” de forma proativa, utilizando os parâmetros de operação da rede e possíveis cenários de falha. A aplicação obtém informações como sequência de ROADMs para cada *lightpath*, potência de entrada por ROADM e variação desta no tempo, além de requisitos (políticas de operação) como máximo *overshoot* e *undershoot* permitidos e máximo tempo de convergência. Utilizando estas variáveis, a aplicação evoca o simulador e obtém resultados, simulando diferentes condições de falha, incluindo atenuação em diferentes spans. Com base nos requisitos e resultados das simulações, o controlador T-SDN pré-configura os parâmetros do mecanismo de equalização a ser utilizado por cada ROADM em caso de falha da comunicação com o controlador. Esta

proposta foi introduzida por M. Siqueira et al. [A].

A Figura 42a apresenta o modelo de simulação criado utilizando-se a ferramenta *Matlab Simulink*, onde o mecanismo de equalização foi implementado através de simulação de eventos discretos. Cada ROADM tem um mecanismo independente, recebendo o sinal do ROADM anterior. O modelo considera que o valor recebido como entrada de cada bloco seja a potência óptica (em dBm) de cada canal. O modelo pode ser criado automaticamente, com base na topologia descoberta pelo controlador T-SDN. Por outro lado, requisitos como máximo *overshoot* e tempo de convergência devem ser configurados como políticas do sistema.

O mecanismo de equalização de cada ROADM é implementado como um controlador *Proportional, Integral, Derivative* (PID). Adicionalmente, a validação da estabilidade do sinal de entrada é realizada verificando-se o desvio padrão ( $\sigma$ ) dos últimos valores de potência medidos. Neste caso, o mecanismo de equalização somente atua se o sinal está estável por um determinado período de tempo. Para todas as simulações apresentadas, a potência de entrada é de 11dBm e o valor alvo é 4dBm. Atenuações de fibra e ROADM e amplificadores não são considerados no modelo, já que se assume que os amplificadores estão operando no modo *Automatic Gain Control* (AGC), de modo a manter ganho fixo, compensando as perdas. Para todas as simulações é inserida uma degradação de 10dB no sinal de entrada do sistema no momento  $t=20s$ . Adicionalmente, nas últimas quatro simulações, há outra degradação de 5dB no último span. Os gráficos apresentados na Figura 42 apresentam os resultados das simulações utilizados para pré-configurar os parâmetros de equalização local em caso de falha dos ROADMs. Os resultados obtidos são descritos a seguir:

- **Atuação simultânea:** Figura 42b mostra a potência óptica de saída versus tempo, com sinal de entrada variando até  $0.4dB$  de forma aleatória. Usando  $Kp = 0.6$ , sem validação de estabilidade ( $\sigma$ ), a execução não ordenada dos mecanismos de equalização dos diferentes ROADMs causou um *overshoot* de aproximadamente  $6dB$ , levando mais de 12 segundos para convergência após a falha.
- **Atuação sequenciada:** os resultados mostrados na Figura 42c foram obtidos através da aplicação de um mecanismo de sequenciamento, desenvolvido como uma das contribuições desta tese, em que cada ROADM somente atua após verificar critério de estabilidade, no qual o ROADM número  $N$  utiliza a medida de potência atual,  $N$  e  $N + 1$  para cálculo do  $\sigma$ . Este mecanismo simples garante que o segundo ROADM na cascata somente inicie a equalização quando o primeiro termina, sem que haja requisitos de coordenação distribuída. Desta forma, a figura mostra que, apesar de não apresentar *overshoot*, quando a degradação ocorre no último *span*, o tempo de convergência não é

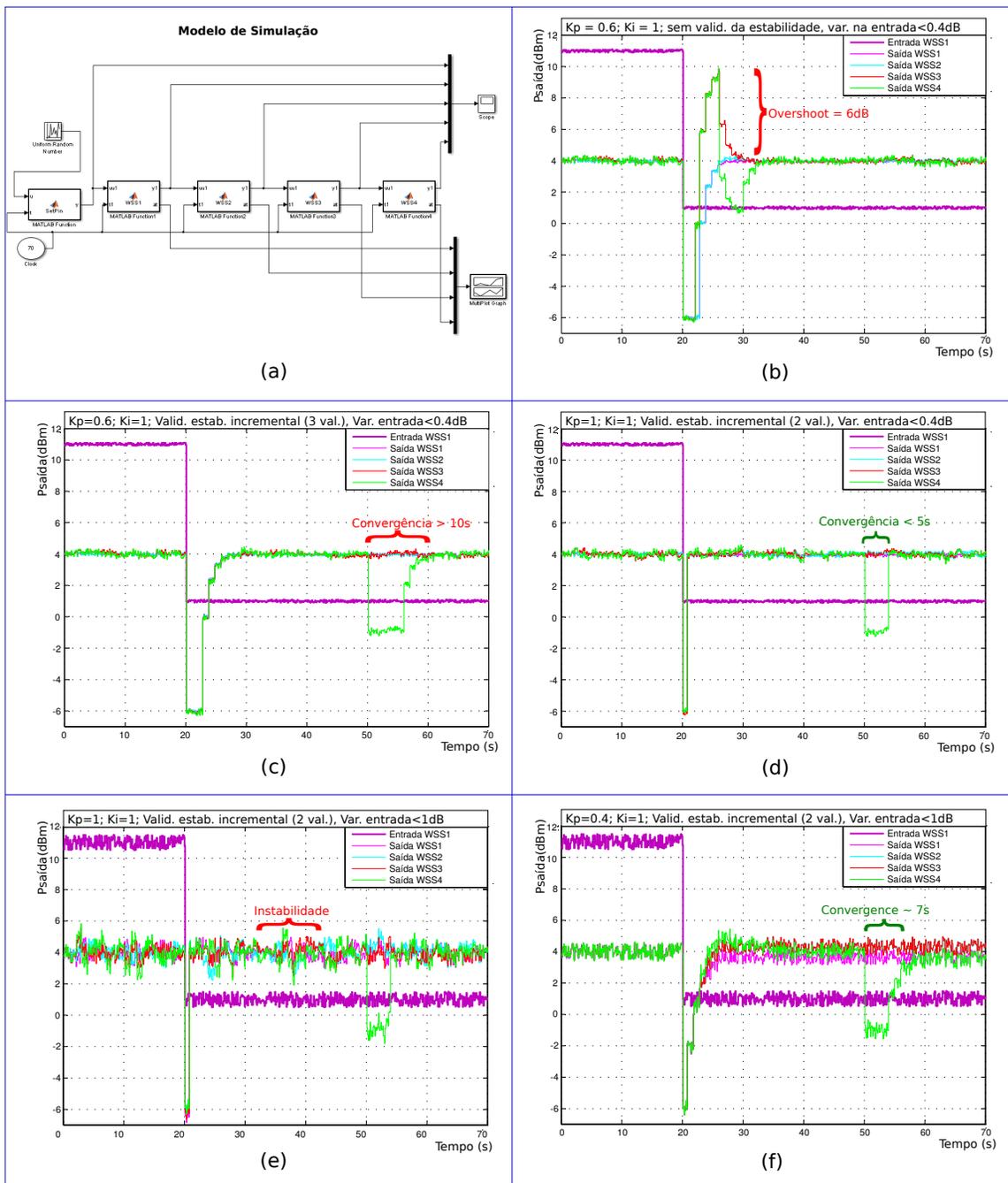


Figura 42 – Resultados de simulação de equalização. Fonte: [A].

aceitável, já que, devido à condição de estabilidade, o sistema requer no mínimo  $N + 1$  segundos para convergir.

- **Atuação sequencial acelerada:** como na simulação anterior o *overshoot* é zero, porém o tempo de convergência não atende, a aplicação de controle das simulações atua no aumento do  $K_p$ , de forma que seja mais agressivo com relação ao tempo de convergência.

Desta forma, com  $Kp = 1$  e validação  $\sigma$ , usando somente o valor atual e a medida  $N$ , onde  $N$  é o identificador do ROADM na sequência, a convergência para o pior caso é de 5 segundos, conforme mostrado na Figura 42d. Portanto, neste cenário, este resultado por ser utilizado para pré-configurar os mecanismos de equalização local dos ROADMs para a situação atual da rede e requisitos (políticas).

- **Sinal de entrada com grande variação:** se o sinal de entrada varia com maior intensidade (1 dB ao invés de 0,4 dB), a pré-configuração da Figura 42d deverá levar o sistema à instabilidade, como mostrado na Figura 42d. Portanto, novas interações devem ser simuladas para busca de novo ponto de ajuste.
- **Estabilização com variação no sinal de entrada:** com o objetivo de minimizar a instabilidade com variação na entrada,  $Kp$  foi reduzido para 0.4, e o limiar de equalização foi aumentado para 1 dB. Mesmo que os resultados mostrados na Figura 42f mostrem que o tempo de convergência sobe para 7 segundos, esta configuração permite que o sinal fique mais estável.

Desta forma, uma aplicação no nível do controlador SDN pode realizar uma comparação dos resultados de simulação com os valores alvo, ajustando as configurações dos elementos de rede para os parâmetros que permitirão a obtenção dos melhores resultados.



## Conclusão e Trabalhos Futuros

Os capítulos iniciais desta tese apresentaram o estado da arte em tecnologias para a construção de redes ópticas de transporte, bem como os desafios e lacunas ainda existentes, incluindo aspectos de controle e gerência destas redes. Adicionalmente, foram revisadas propostas e mecanismos para gerência de redes com base em políticas, redes autonômicas e foram discutidas novas propostas relacionadas a SDN e virtualização de redes.

A principal contribuição da tese consistiu na proposta de uma arquitetura SDN para redes ópticas de transporte de nova geração, com suporte a políticas. A proposta foi validada através de implementação e realização de experimentos, permitindo concluir-se que equipamentos de redes modelados através da linguagem YANG facilitam a abstração da rede a nível do controlador SDN, viabilizando a representação do equipamento e da rede como grafos através de transformações. Adicionalmente, observou-se que é essencial a implementação de sistema de políticas no contexto do sistema operacional de rede, de forma a controlar as requisições, concorrência e acesso a recursos das aplicações SDN. Neste contexto, a proposta do modelo *policy flow*, modelado sobre a abstração em grafos da rede, mostrou-se adequada para operar no escopo do controlador T-SDN. Finalmente, concluiu-se que a visão logicamente centralizada provida pela arquitetura SDN viabiliza a implementação de aplicações e políticas de forma mais simples e direta que em redes com plano de controle distribuído.

Vale ressaltar que o trabalho foi sustentado por diversas publicações em conferências e jornais, descritas no Apêndice A. No entanto, o tema SDN em redes ópticas de transporte, apesar de ainda ser embrionário, tem sido objeto de intensa pesquisa e desenvolvimento pela academia e indústria, de forma que muitas propostas têm sido feitas para extensão dos conceitos e protocolos. Neste contexto, muitos desafios ainda deverão ser superados, tanto no nível da solução SDN como dos equipamentos da camada de dados, para que as redes ópticas de transporte comerciais possam se beneficiar das vantagens propostas pelo paradigma SDN. No nível da arquitetura SDN, devem ser desenvolvidas soluções efetivas para a implementação de funções de provisionamento de serviços, automação na operação, proteção e restauração, entre outras, que superem em desempenho, robustez e em dimensionamento da rede as técnicas atuais, simplificando drasticamente a operação destas redes, incluindo cenários multicamadas. Já no nível dos elementos de rede, devem ser disponibilizadas APIs padronizadas, além de serem necessárias evoluções tecnológicas nas camadas de comutação e transmissão, como por exemplo, para que ROADMs suportem chaveamento sub-50ms e suporte a mudança de formato de modulação *hitless* por transponders flexíveis.

A associação entre SDN e PBNM realizada no contexto desta tese, reforçada por trabalhos recentes da Cisco (SMITH *et al.*, 2014) e da VmWare (BALLAND; HINRICHS, 2014), deverá permitir que a operação das redes tenha um maior grau de automação. No entanto, é necessário que todo o trabalho realizado em PBNM na última década seja revisitado com a ótica dos princípios SDN, incluindo a utilização de interfaces abertas e plano de controle logicamente centralizado. Em última instância, deverão ser propostos novos paradigmas para modelo, linguagem e motor de políticas. Adicionalmente, a aplicação de políticas em redes ópticas de transporte foi pouco explorada, já que tais redes eram implementadas de forma estática. No entanto, com a evolução das tecnologias no plano de dados, bem como com a integração entre camadas, surgem aplicações para o controle por políticas de parâmetros de operação das redes além dos abordados nesta tese, como o controle mais granular de parâmetros de operação de transmissores e receptores coerentes, controle de consumo dos equipamentos, utilizando políticas e visão global da rede, proteção e restauração, incluindo cenários multicamadas.

Finalmente, foi explorada a aplicação do conceito de virtualização de funções de redes (NFV) no contexto de redes ópticas de transporte. Observou-se que, apesar do paradigma SDN facilitar o controle destas redes, a efetiva construção de elementos de rede de transporte virtualizados, nos moldes de aplicações NFV em plataformas *x86*, ainda é um desafio, já que os blocos de construção de elementos de redes ópticas são muito específicos.

# Referências

ADI, A.; ETZION, O. The Situation Manager Rule Language. In: *RuleML*. [S.l.: s.n.], 2002. v. 60, p. 36–57. Citado na página 71.

ADVA (Ed.). *Converged Packet Transport From Juniper Networks and ADVA Optical Networking*. 2013. Disponível em: <[http://www.advaoptical.com/~media/AboutUs/Partners/Documents/Partner\\\_Juniper.ashx](http://www.advaoptical.com/~media/AboutUs/Partners/Documents/Partner\_Juniper.ashx)>. Citado na página 2.

ALCATEL-LUCENT (Ed.). *Alcatel-Lucent 1830 Photonic Service Switch*. [S.l.], 2013. Disponível em: <<http://www.alcatel-lucent.com/products/1830-photonic-service-switch>>. Citado na página 1.

AMAYA, N.; ZERVAS, G. S.; SIMEONIDOU, D. Architecture on Demand for Transparent Optical Networks. In: IEEE. *Transparent Optical Networks (ICTON), 2011 13th International Conference on*. [S.l.], 2011. p. 1–4. Citado na página 54.

AYANOGLU, E. Reduction of Restoration Capacity Requirements in Advanced Optical Networks. In: IEEE. *Global Telecommunications Conference, 1995. GLOBECOM'95., IEEE*. [S.l.], 1995. v. 2, p. 1018–1022. Citado na página 8.

BALLAND, P.; HINRICHS, T. Congress - a system for declaring, auditing, and enforcing policy in heterogeneous cloud environments. *OpenStack Summit, Atlanta-USA*, 2014. Citado 2 vezes nas páginas 59 e 112.

BEHRINGER, M.; PRITIKIN, M.; BJARNASON, S.; CLEMM, A. *A Framework for Autonomic Networking*. IETF, 2013. Informational. (Internet Draft). Disponível em: <<http://tools.ietf.org/html/draft-behringer-autonomic-network-framework-01>>. Citado na página 31.

BOHN, M.; NAPOLI, A.; D'ERRICO, A.; FERRARIS, G.; GUNKEL, M.; JIMENEZ, F.; FERNANDEZ-PALACIOS, J. P.; LAYEC, P.; ZERVAS, G.; AMAYA, N. *et al.* Elastic Optical Networks: The Vision of the ICT Project IDEALIST. In: IEEE. *Future Network and Mobile Summit (FutureNetworkSummit), 2013*. [S.l.], 2013. p. 1–9. Citado na página 11.

BROADCOM (Ed.). *Broadcom BCM88650 Series, 200G Integrated Packet Processor, Traffic Manager and Fabric Interface Single-Chip Device*. [S.l.], 2013. (Product Sheet). Disponível em: <<http://www.broadcom.com/collateral/pb/88650-PB200-non-nda.pdf>>. Citado na página 56.

BROCHELER, M.; PUGLIESE, A.; SUBRAHMANIAN, V. COSI: Cloud Oriented Subgraph Identification in Massive Social Networks. In: IEEE. *Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on*. [S.l.], 2010. p. 248–255. Citado 2 vezes nas páginas 68 e 74.

CHEN, Y.; QIAO, C.; YU, X. Optical Burst Switching: a New Area in Optical Networking Research. *Network, IEEE*, IEEE, v. 18, n. 3, p. 16–23, 2004. Citado na página 16.

- CHOWDHURY, N.; BOUTABA, R. A Survey of Network Virtualization. *Computer Networks*, Elsevier, v. 54, n. 5, p. 862–876, 2010. Citado na página 48.
- CIENA (Ed.). *Is It Finally Time for Network as a Service*. 2013. Ciena Whitepaper. Disponível em: <<http://www.ciena.com/resources/white-papers/>>. Citado 2 vezes nas páginas 7 e 50.
- CISCO (Ed.). *Cisco Visual Networking Index: Forecast and Methodology, 2012–2017*. 2012. Disponível em: <[www.cisco.com/go/vni](http://www.cisco.com/go/vni)>. Citado na página 1.
- CISCO (Ed.). *Building Cost Effective and Scalable CORE Networks Using an Elastic Architecture*. 2013. Disponível em: <[http://www.cisco.com/en/US/prod/collateral/routers/ps5763/white\\\_paper\\\_c11-727983.pdf](http://www.cisco.com/en/US/prod/collateral/routers/ps5763/white\_paper\_c11-727983.pdf)>. Citado na página 2.
- TAIL-F (Ed.). *Tail-f ConfD Datasheet*. 2013. Citado na página 61.
- CLAUBERG, A. Open INTERNET Standards DT. In: . [S.l.]: Presented at the IETF87, August 2013, Berlin - Germany, 2013. Citado na página 41.
- COMER, D. Network Processors: Programmable Technology for Building Network Systems. *The Internet Protocol Journal*, v. 7, n. 4, p. 2–12, 2004. Citado na página 92.
- DAMIANOU, N. *A Policy Framework for Management of Distributed Systems*. Tese (Doutorado) — Imperial College London, March 2002. Disponível em: <<http://pubs.doc.ic.ac.uk/policy-framework-distrib-systems/>>. Citado na página 36.
- DAS, S. *PAC. C: A Unified Control Architecture for Packet and Circuit Network Convergence*. Tese (Doutorado) — Stanford University, 2012. Citado na página 6.
- DAS, S.; PARULKAR, G.; MCKEOWN, N. Rethinking IP Core Networks. *Journal of Optical Communications and Networking*, Optical Society of America, v. 5, n. 12, p. 1431–1442, 2013. Citado 3 vezes nas páginas 96, 97 e 98.
- DUNLOP, N. *Dynamic Policy-based Management in Open Distributed Environments*. Ph.D — University of Queensland, 2002. Citado na página 33.
- ENNS, R.; BJORKLUND, M.; SCHOENWAELDER, J.; BIERMAN, A. *Network Configuration Protocol (NETCONF)*. IETF, 2011. RFC 6241 (Proposed Standard). (Request for Comments, 6241). Disponível em: <<http://www.ietf.org/rfc/rfc6241.txt>>. Citado na página 43.
- ESTEVEZ, R. P.; GRANVILLE, L. Z.; BOUTABA, R. On the Management of Virtual Networks. *Comm. Magazine, IEEE*, IEEE, v. 51, n. 7, 2013. Citado 2 vezes nas páginas 7 e 60.
- ETSI Group Specification (Ed.). *Autonomic Network Engineering for the Self-managing Future Internet (AFI); Scenarios, Use Cases and Requirements for Autonomic/Self-Managing Future Internet*. 2011. (ETSI GS AFI 001 V1.1.1). Citado na página 31.

- ETSI Group Specification (Ed.). *Autonomic Network Engineering for the Self-managing Future Internet (AFI); Generic Autonomic Network Architecture (An Architectural Reference Model for Autonomic Networking, Cognitive Networking and Self-Management)*. 2013. (ETSI GS AFI 002 V1.1.1). Citado na página 31.
- CHIOSI, M. (Ed.). *Network Functions Virtualisation - An Introduction, Benefits, Enablers, Challenges & Call for Action*. 2012. (White Paper). Disponível em: <[http://portal.etsi.org/NFV/NFV\\_White\\_Paper.pdf](http://portal.etsi.org/NFV/NFV_White_Paper.pdf)>. Citado 2 vezes nas páginas 2 e 51.
- ETSI Group Specification (Ed.). *Network Functions Virtualisation (NFV) Architectural Framework*. 2013. (ETSI GS NFV 002 V1.1.1). Disponível em: <[http://www.etsi.org/deliver/etsi\\_gs/NFV/001\\_099/002/01.01.01\\_60/gs\\_NFV002v010101p.pdf](http://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.01.01_60/gs_NFV002v010101p.pdf)>. Citado 2 vezes nas páginas 51 e 52.
- FARREL, A.; VASSEUR, J.-P.; ASH, J. *A Path Computation Element (PCE)-Based Architecture*. IETF, 2006. RFC 4655 (Informational). (Request for Comments, 4655). Disponível em: <<http://www.ietf.org/rfc/rfc4655.txt>>. Citado 2 vezes nas páginas 23 e 25.
- FEAMSTER, N.; REXFORD, J.; ZEGURA, E. The road to sdn: an intellectual history of programmable networks. *ACM SIGCOMM Computer Communication Review*, ACM, v. 44, n. 2, p. 87–98, 2014. Citado na página 49.
- FORGY, C. L. Rete: A Fast Algorithm for the Many Pattern/many Object Pattern Match Problem. *Artificial intelligence*, Elsevier, v. 19, n. 1, p. 17–37, 1982. Citado na página 34.
- FOSTER, N.; GUHA, A.; REITBLATT, M.; STORY, A.; FREEDMAN, M.; KATTA, N.; MONSANTO, C.; REICH, J.; REXFORD, J.; SCHLESINGER, C.; WALKER, D.; HARRISON, R. Languages for Software-Defined Networks. *Communications Magazine, IEEE*, v. 51, n. 2, p. 128–134, 2013. ISSN 0163-6804. Citado 2 vezes nas páginas 3 e 7.
- GARRICH, M.; AMAYA, N.; ZERVAS, G. S.; GIACCONE, P.; SIMEONIDOU, D. Architecture on Demand: Synthesis and Scalability. In: IEEE. *Optical Network Design and Modeling (ONDM), 2012 16th International Conference on*. [S.l.], 2012. p. 1–6. Citado na página 55.
- GERLACH, C. (Ed.). *OIF Carrier WG Requirements on Transport Networks in SDN Architectures*. 2013. (OIF Carrier Working Group Documents). Disponível em: <[http://www.oiforum.com/public/carrier\\_wg\\_documents.html](http://www.oiforum.com/public/carrier_wg_documents.html)>. Citado na página 45.
- GERSTEL, O.; JINNO, M.; LORD, A.; YOO, S. B. Elastic Optical Networking: A New Dawn for the Optical Layer? *Communications Magazine, IEEE, IEEE*, v. 50, n. 2, p. s12–s20, 2012. Citado na página 11.
- GHALLAB, M.; NAU, D.; TRAVERSO, P. *Automated Planning: Theory & Practice*. [S.l.]: Elsevier, 2004. Citado na página 76.
- GLINGENER, C. Optical Networking Trends and Evolution. In: OPTICAL SOCIETY OF AMERICA. *Optical Fiber Communication Conference*. [S.l.], 2011. p. OThAA1. Citado na página 1.

- GREDLER, H.; MEDVED, J.; PREVIDI, S.; FARREL, A.; RAY, S. *North-Bound Distribution of Link-State and TE Information using BGP*. IETF, 2014. (Internet Drafts). Disponível em: <draft-ietf-idr-ls-distribution-05>. Citado na página 41.
- GRINGERI, S.; BITAR, N.; XIA, T. J. Extending Software Defined Network Principles to Include Optical Transport. *Communications Magazine, IEEE*, IEEE, v. 51, n. 3, p. 32–40, 2013. Citado 2 vezes nas páginas 2 e 6.
- GUTZ, S.; STORY, A.; SCHLESINGER, C.; FOSTER, N. Splendid Isolation: A Slice Abstraction for Software-Defined Networks. In: ACM. *Proceedings of the first workshop on Hot topics in software defined networks*. [S.l.], 2012. p. 79–84. Citado na página 49.
- HAMMAD, A.; NEJABATI, R.; SIMEONIDOU, D. Novel Approaches for composition of online virtual optical networks utilizing O-OFDM technology. *ECOC 2013 Mo.4.E.4*, 2013. Citado na página 50.
- HOLZSCHUHER, F.; PEINL, R. Performance of Graph Query Languages: Comparison of Cypher, Gremlin and Native Access in Neo4j. In: ACM. *Proceedings of the Joint EDBT/ICDT 2013 Workshops*. [S.l.], 2013. p. 195–204. Citado na página 69.
- HOOFT, F.; MADEIRA, E. Multi-layer Network Equipment Configuration Based on Multi-property Graph Descriptions. *Unpublished*, 2013. Citado na página 62.
- IBM (Ed.). *IBM Software Defined Network for Virtual Environments*. 2013. (Datasheet). Disponível em: <<http://public.dhe.ibm.com/common/ssi/ecm/en/qcd03023usen/QCD03023USEN.PDF>>. Citado na página 49.
- IEEE (Ed.). *IEEE 802.1AB-2009 - Station and Media Access Control Connectivity Discovery (LLDP)*. 2009. Citado na página 81.
- INFINERA (Ed.). *Infinera DTN-X Multi-Terabit Packet Optical Network Platform*. [S.l.], 2013. Disponível em: <<http://www.infinera.com>>. Citado na página 1.
- SCHMITT, A. (Ed.). *The Fast Approaching 100G Era*. 2011. Citado na página 1.
- SCHMITT, A. (Ed.). *Integrated OTN Switching Virtualizes Optical Networks*. 2012. Citado na página 50.
- INNISS, D. Winning Metro 100G. In: *ECOC 2013, Market Focus*. [S.l.: s.n.], 2013. Citado na página 96.
- BJORKLUND, M. (Ed.). *YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)*. IETF, 2010. RFC 6020 (Proposed Standard). (Request for Comments, 6020). Disponível em: <<http://www.ietf.org/rfc/rfc6020.txt>>. Citado 2 vezes nas páginas 5 e 44.
- ITU-T (Ed.). *Spectral Grids for WDM Applications: DWDM Frequency Grid*. 2012. Citado 2 vezes nas páginas 11 e 84.
- ITU-T (Ed.). *Interfaces for the Optical Transport network (OTN)*. 2009. Citado na página 17.

- JINNO, M.; HIRANO, A. Toward Deeply Virtualized Elastic Optical Networks. In: OPTICAL SOCIETY OF AMERICA. *National Fiber Optic Engineers Conference*. [S.l.], 2013. Citado na página 55.
- JOHN, W. *Split Architecture for Large Scale Wide Area Networks, Deliverable D3.3*. 2012. Disponível em: <<http://www.fp7-sparc.eu/project/>>. Citado na página 7.
- KARAGIANNIS, G.; LIU, W.; TSOU, T.; SUN, Q.; LOPEZ, D.; YEGANI, P.; TREMBLAY, J. *Problem Statement for Shared Unified Policy Automation (SUPA)*. IETF, 2014. (Internet Drafts). Disponível em: <[draft-karagiannis-sup-a-problem-statement-00](#)>. Citado na página 8.
- KEPHART, J. O.; CHESS, D. M. The Vision of Autonomic Computing. *Computer*, IEEE, v. 36, n. 1, p. 41–50, 2003. Citado na página 29.
- KERIVIN, H.; MAHJOUB, A. R. Design of Survivable Networks: A Survey. *Networks*, Wiley Online Library, v. 46, n. 1, p. 1–21, 2005. Citado na página 8.
- KIM, I.; PALACHARLA, P.; WANG, X.; BIHON, D.; FEUER, M. D.; WOODWARD, S. L. Performance of Colorless, Non-Directional ROADMs with Modular Client-side Fiber Cross-connects. In: OPTICAL SOCIETY OF AMERICA. *National Fiber Optic Engineers Conference*. [S.l.], 2012. Citado na página 15.
- KOMPELLA, K.; REKHTER, Y. *Routing Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)*. IETF, 2005. RFC 4202 (Proposed Standard). (Request for Comments, 4202). Updated by RFCs 6001, 6002. Disponível em: <<http://www.ietf.org/rfc/rfc4202.txt>>. Citado na página 22.
- KREUTZ, D.; RAMOS, F.; VERISSIMO, P.; ROTHENBERG, C. E.; AZODOLMOLKY, S.; UHLIG, S. Software-Defined Networking: A Comprehensive Survey. *arXiv preprint arXiv:1406.0440*, 2014. Citado 3 vezes nas páginas 2, 37 e 58.
- LEE, Y.; BERNSTEIN, G.; SO, N.; FANG, L.; CECCARELLI, D.; LOPEZ, D.; DIOS, O. G. de. *Network Control Function Virtualization for Transport SDN*. IETF, 2013. Informational. (Internet Draft). Disponível em: <<https://tools.ietf.org/html/draft-lee-network-control-function-virtualization-01>>. Citado 2 vezes nas páginas 66 e 67.
- LEE, Y.; ROUX, J. L.; KING, D.; OKI, E. *Path Computation Element Communication Protocol (PCEP) Requirements and Protocol Extensions in Support of Global Concurrent Optimization*. IETF, 2009. RFC 5557 (Proposed Standard). (Request for Comments, 5557). Disponível em: <<http://www.ietf.org/rfc/rfc5557.txt>>. Citado na página 27.
- OPENDAYLIGHT (Ed.). *OpenDayLight Technical Overview*. OpenDayLight Project, 2015. Disponível em: <<http://www.opendaylight.org/project/technical-overview>>. Citado 2 vezes nas páginas xix e 42.
- LIU, C. Is there a Role for GMPLS in Transport SDN? *Techs in Paradise 2013*, ESCC/Internet2, 2013. Citado na página 6.

- LIU, L.; MUÑOZ, R.; CASELLAS, R.; TSURITANI, T.; MARTÍNEZ, R.; MORITA, I. OpenSlice: an OpenFlow-based Control Plane for Spectrum Sliced Elastic Optical Path Networks. In: OPTICAL SOCIETY OF AMERICA. *European Conference and Exhibition on Optical Communication*. [S.l.], 2012. Citado na página 6.
- LOTHBERG, P. TeraStream – A Simplified IP Network Service Delivery Model. In: . [S.l.]: Presented at the RIPE67, October 2013, Athens - Greece, 2013. Citado na página 42.
- LOVE, R. *Linux Kernel Development*. Pearson Education, 2010. (Developer's Library). ISBN 9780768696790. Disponível em: <<http://books.google.com.br/books?id=3MWRMYRwulIC>>. Citado na página 74.
- LUPU, E.; SLOMAN, M.; DULAY, N.; DAMIANOU, N. Ponder: Realising Enterprise Viewpoint Concepts. In: IEEE. *Enterprise Distributed Object Computing Conference, 2000. EDOC 2000. Proceedings. Fourth International*. [S.l.], 2000. p. 66–75. Citado na página 33.
- MAGALHAES, E. C.; NASCIMENTO, M. R.; OLIVEIRA, V. G. de; ANGELI, B. C. C.; OLIVEIRA, J. R. F. de; OLIVEIRA, J. C. R. F. de; BORDONALLI, A. C. Node Architectures for Next Generation ROADMs: A Comparative Study Among Emergent Optical Solutions. *Journal of Microwaves, Optoelectronics and Electromagnetic Applications*, v. 12, p. 156–166, 2013. Citado 3 vezes nas páginas 11, 12 e 14.
- MAGRATH, S. *An Inquiry Into Pbnm System Performance Required For Massive Scale Telecommunication Applications*. Ph.D — University of Technology, Sydney, 2006. Citado 4 vezes nas páginas 33, 34, 35 e 70.
- MAHALINGAM, M.; DUTT, D.; DUDA, K.; AGARWAL, P.; KREEGER, L.; SRIDHAR, T.; WRIGHT, M. B. and C. *VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks*. IETF, 2013. (Internet Drafts). Disponível em: <[draft-mahalingam-dutt-dcops-vxlan-05](http://draft-mahalingam-dutt-dcops-vxlan-05)>. Citado na página 48.
- MANNIE, E.; PAPADIMITRIOU, D. *Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)*. IETF, 2006. RFC 4427 (Informational). (Request for Comments, 4427). Disponível em: <<http://www.ietf.org/rfc/rfc4427.txt>>. Citado 2 vezes nas páginas 19 e 20.
- MARGARIA, C.; DIOS, O. G. de; ZHANG, F. *PCEP extensions for GMPLS*. IETF, 2014. (Internet Draft). Disponível em: <<http://tools.ietf.org/html/draft-ietf-pce-gmpls-pcep-extensions-09>>. Citado na página 28.
- MCCAULEY, J.; PANDA, A.; CASADO, M.; KOPONEN, T.; SHENKER, S. Extending SDN to Large-Scale Networks. *ONS13 Research Track*, 2013. Citado na página 67.
- MCKEOWN, N.; ANDERSON, T.; BALAKRISHNAN, H.; PARULKAR, G.; PETERSON, L.; REXFORD, J.; SHENKER, S.; TURNER, J. OpenFlow: Enabling Innovation in Campus Networks. *ACM SIGCOMM Computer Communication Review*, ACM, v. 38, n. 2, p. 69–74, 2008. Citado 2 vezes nas páginas 37 e 49.

- MELLE, S.; DEORE, A.; TURKCU, O.; AHUJA, S.; HAND, S. Comparing Optical & OTN Switching Architectures in Next-Gen 100Gb/s Networks. In: OPTICAL SOCIETY OF AMERICA. *National Fiber Optic Engineers Conference*. [S.l.], 2013. Citado na página 17.
- MEZHOUDI, M.; HU, Y. Optical Backbone Network Evolution: Design, Optimization and Evaluation of NG-OTN. In: IEEE. *Wireless and Optical Communications Conference (WOCC), 2010 19th Annual*. [S.l.], 2010. p. 1–6. Citado na página 17.
- MIGUEL, I. de; DURÁN, R. J.; JIMÉNEZ, T.; FERNÁNDEZ, N.; AGUADO, J. C.; LORENZO, R. M.; CABALLERO, A.; MONROY, I. T.; YE, Y.; TYMECKI, A. *et al.* Cognitive Dynamic Optical Networks [invited]. *Journal of Optical Communications and Networking*, Optical Society of America, v. 5, n. 10, p. A107–A118, 2013. Citado na página 32.
- MOK, W.; GASS, K. *OTN Over Packet Fabric Protocol (OFP) Implementation Agreement*. 2011. (Implementation Agreement). Disponível em: <<http://www.oiforum.com/public/documents/oif-ofp-01.0.pdf>>. Citado na página 17.
- MONSANTO, C.; REICH, J.; FOSTER, N.; REXFORD, J.; WALKER, D. Composing Software Defined Networks. *NSDI, Apr*, 2013. Citado na página 7.
- MRAZE, K.; WAKELIN, S. *Enabling OTN Switching over Packet/Cell Fabrics*. 2011. (White Paper). Disponível em: <<http://pmcs.com/cgi-bin/document.pl?docnum=2114338>>. Citado na página 17.
- NADEAU, T.; GRAY, K. *Sdn: Software Defined Networks*. O'Reilly & Associates Incorporated, 2013. ISBN 9781449342302. Disponível em: <<http://books.google.com.br/books?id=l01LmwEACAAJ>>. Citado 2 vezes nas páginas xix e 40.
- NAKAO, A. Network Virtualization as Foundation for Enabling new Network Architectures and Applications. *IEICE transactions on communications*, The Institute of Electronics, Information and Communication Engineers, v. 93, n. 3, p. 454–457, 2010. Citado na página 48.
- OKI, E.; TAKEDA, T.; FARREL, A. *Extensions to the Path Computation Element Communication Protocol (PCEP) for Route Exclusions*. IETF, 2009. RFC 5521 (Proposed Standard). (Request for Comments, 5521). Disponível em: <<http://www.ietf.org/rfc/rfc5521.txt>>. Citado na página 26.
- O'MAHONY, M. J.; SIMEONIDOU, D.; HUNTER, D. K.; TZANAKAKI, A. The Application of Optical Packet Switching in Future Communication Networks. *Communications Magazine, IEEE*, IEEE, v. 39, n. 3, p. 128–135, 2001. Citado na página 16.
- ONF (Ed.). *OF-CONFIG 1.2: OpenFlow Management and Configuration Protocol*. [S.l.]: ONF, 2014. Citado na página 66.
- OTANI, T.; OGAKI, K.; CAVIGLIA, D.; ZHANG, F.; MARGARIA, C. *Requirements for GMPLS Applications of PCE*. IETF, 2013. RFC 7025 (Informational). (Request for

- Comments, 7025). Disponível em: <<http://www.ietf.org/rfc/rfc7025.txt>>. Citado na página 28.
- PAPADIMITRIOU, D.; MANNIE, E. *Analysis of Generalized Multi-Protocol Label Switching (GMPLS)-based Recovery Mechanisms (including Protection and Restoration)*. IETF, 2006. RFC 4428 (Informational). (Request for Comments, 4428). Disponível em: <<http://www.ietf.org/rfc/rfc4428.txt>>. Citado na página 20.
- PAPADIMITRIOU, D.; VIGOUREUX, M.; SHIOMOTO, K.; BRUNGARD, D.; ROUX, J. L. *Generalized MPLS (GMPLS) Protocol Extensions for Multi-Layer and Multi-Region Networks (MLN/MRN)*. IETF, 2010. RFC 6001 (Standards Track). (Request for Comments, 6001). Disponível em: <<http://www.ietf.org/rfc/rfc6001.txt>>. Citado na página 23.
- PENG, S.; NEJABATI, R.; SIMEONIDOU, D. Impairment-Aware Optical Network Virtualization in Single-Line-Rate and Mixed-Line-Rate WDM Networks. *Journal of Optical Communications and Networking*, Optical Society of America, v. 5, n. 4, p. 283–293, 2013. Citado na página 7.
- PERELLO, J.; SPADARO, S.; GARCIA-ESPIN, J.; RIERA, J. F.; FIGUEROLA, S. *et al.* Optimal Allocation of Virtual Optical Networks for the Future Internet. In: IEEE. *Optical Network Design and Modeling (ONDM), 2012 16th International Conference on*. [S.l.], 2012. p. 1–6. Citado na página 7.
- PERLIN, M.; DEBAUD, J.-M. Match Box: Fine-grained Parallelism at the Match Level. In: IEEE. *Tools for Artificial Intelligence, 1989. Architectures, Languages and Algorithms, IEEE International Workshop on*. [S.l.], 1989. p. 428–434. Citado na página 34.
- PERRIN, S. The Need for Next Generation ROADM Networks. *Heavy Reading White Paper*, 2010. Citado na página 1.
- ROTHENBERG, C. E.; NASCIMENTO, M. R.; SALVADOR, M. R.; CORRÊA, C. N. A.; LUCENA, S. Cunha de; RASZUK, R. Revisiting Routing Control Platforms with the Eyes and Muscles of Software-Defined Networking. In: *Proceedings of the first workshop on Hot topics in software defined networks*. New York, NY, USA: ACM, 2012. (HotSDN '12), p. 13–18. ISBN 978-1-4503-1477-0. Citado na página 7.
- ROUX, J. L.; PAPADIMITRIOU, D. *Evaluation of Existing GMPLS Protocols Against Multi-Layer and Multi-Region Networks (MLN/MRN)*. IETF, 2008. RFC 5339 (Informational). (Request for Comments, 5339). Disponível em: <<http://www.ietf.org/rfc/rfc5339.txt>>. Citado na página 23.
- ROUX, J. L.; VASSEUR, J.; IKEJIRI, Y.; ZHANG, R. *OSPF Protocol Extensions for Path Computation Element (PCE) Discovery*. IETF, 2008. RFC 5088 (Proposed Standard). (Request for Comments, 5088). Disponível em: <<http://www.ietf.org/rfc/rfc5088.txt>>. Citado na página 25.
- ROUX, J. L.; VASSEUR, J.; LEE, Y. *Encoding of Objective Functions in the Path Computation Element Communication Protocol (PCEP)*. IETF, 2009. RFC 5541 (Proposed Standard). (Request for Comments, 5541). Disponível em: <<http://www.ietf.org/rfc/rfc5541.txt>>. Citado na página 26.

- SANTOS, G.; SANTOS, F.; OLIVEIRA, J.; MOREIRA, R.; SALVADOR, M.; MAGALHÃES, M. Quantitative Bandwidth Analysis of Automatic Circuit Provisioning on ROADM-Based Networks Using GMPLS Control Plane. In: IEEE. *Microwave and Optoelectronics Conference (IMOC), 2009 SBMO/IEEE MTT-S International*. [S.l.], 2009. p. 664–668. Citado na página 82.
- SCHMOGROW, R.; HILLERKUSS, D.; DRESCHMANN, M.; HUEBNER, M.; WINTER, M.; MEYER, J.; NEBENDAHL, B.; KOOS, C.; BECKER, J.; FREUDE, W. *et al.* Real-time Software-defined Multiformat Transmitter Generating 64QAM at 28 GBd. *Photonics Technology Letters, IEEE, IEEE*, v. 22, n. 21, p. 1601–1603, 2010. Citado na página 89.
- SCHOENTGEN, A. Network Sharing: a Hot Topic for Operators and Regulators in Sub-Saharan Africa. *Communications and Strategies*, n. 86, p. 43, 2012. Citado na página 50.
- SCHOENWAELDER, J. *Overview of the 2002 IAB Network Management Workshop*. IETF, 2003. RFC 3535 (Informational). (Request for Comments, 3535). Disponível em: <<http://www.ietf.org/rfc/rfc3535.txt>>. Citado na página 43.
- SHAFER, P. *An Architecture for Network Management Using NETCONF and YANG*. IETF, 2011. RFC 6244 (Informational). (Request for Comments, 6244). Disponível em: <<http://www.ietf.org/rfc/rfc6244.txt>>. Citado na página 43.
- SHEN, G.; LI, Y.; PENG, L. How Much can Colorless, Directionless and Contentionless (CDC) of ROADM Help Dynamic Lightpath Provisioning? In: OPTICAL SOCIETY OF AMERICA. *National Fiber Optic Engineers Conference*. [S.l.], 2012. Citado na página 16.
- SHERWOOD, R.; CHAN, M.; COVINGTON, A.; GIBB, G.; FLAJSLIK, M.; HANDIGOL, N.; HUANG, T.-Y.; KAZEMIAN, P.; KOBAYASHI, M.; NAOUS, J. *et al.* Carving Research Slices out of your Production Networks with OpenFlow. *ACM SIGCOMM Computer Communication Review, ACM*, v. 40, n. 1, p. 129–130, 2010. Citado na página 49.
- SHIOMOTO, K.; PAPADIMITRIOU, D.; ROUX, J. L.; VIGOUREUX, M.; BRUNGARD, D. *Requirements for GMPLS-Based Multi-Region and Multi-Layer Networks (MRN/MLN)*. IETF, 2008. RFC 5212 (Informational). (Request for Comments, 5212). Disponível em: <<http://www.ietf.org/rfc/rfc5212.txt>>. Citado na página 21.
- SMALLEY, S.; VANCE, C.; SALAMON, W. Implementing SELinux as a Linux Security Module. *NAI Labs Report*, v. 1, p. 43, 2001. Citado na página 74.
- SMITH, M.; DVORKIN, M.; LARIBI, Y.; PANDEY, V.; GARG, P.; WEIDENBACHER, N. *OpFlex Control Protocol*. IETF, 2014. (Internet Drafts). Disponível em: <[draft-smith-opflex-00](#)>. Citado 4 vezes nas páginas 3, 46, 59 e 112.
- SRIVASTAVA, B.; KAMBHAMPATI, S. The Case for Automated Planning in Autonomic Computing. In: *ICAC*. [S.l.: s.n.], 2005. v. 5, p. 331–332. Citado na página 76.
- STRASSNER, J. DEN-ng: Achieving Business-driven Network Management. In: IEEE. *Network Operations and Management Symposium, 2002. NOMS 2002. 2002 IEEE/IFIP*. [S.l.], 2002. p. 753–766. Citado na página 33.

- STRASSNER, J. *Policy-Based Network Management: Solutions for the Next Generation*. [S.l.]: Morgan Kaufmann, 2003. Citado 2 vezes nas páginas 2 e 33.
- STRASSNER, J.; SCHLEIMER, S. *Policy Framework Definition Language*. IETF, 1998. (Internet Draft). Disponível em: <<https://tools.ietf.org/html/draft-ietf-policy-framework-pfdl-00>>. Citado na página 35.
- TAIL-F (Ed.). *Creating and Modifying Network Services Using Tail-f NCS*. 2014. Citado na página 43.
- TANG, L.; BILLENHALLI, S.; HUANG, W.; RAZO, M.; SIVASANKARAN, A.; VARDHAN, H.; MONTI, P.; TACCA, M.; FUMAGALLI, A. The PlaNet-OTN Module: A Double Layer Design Tool for Optical Transport Networks. In: IEEE. *Transparent Optical Networks, 2009. ICTON'09. 11th International Conference on*. [S.l.], 2009. p. 1–5. Citado na página 8.
- TANG, L.; HUANG, W.; RAZO, M.; SIVASANKARAN, A.; MONTI, P.; TACCA, M.; FUMAGALLI, A. A Customizable Two-step Framework for General Equipment Provisioning in Optical Transport Networks. In: IEEE. *Optical Network Design and Modeling (ONDM), 2011 15th International Conference on*. [S.l.], 2011. p. 1–6. Citado na página 8.
- TMFORUM (Ed.). *TMF518-FMW Framework Document Delivery Package Business Agreement*. [S.l.], 2008. Disponível em: <<http://www.tmforum.org/DownloadCenter/7549/home.html#mtosi>>. Citado na página 58.
- VASSEUR, J.; LEROUX, J.; YASUKAWA, S.; PREVIDI, S.; PSENAK, P.; MABBEY, P. *Routing Extensions for Discovery of Multiprotocol (MPLS) Label Switch Router (LSR) Traffic Engineering (TE) Mesh Membership*. IETF, 2007. RFC 4972 (Proposed Standard). (Request for Comments, 4972). Disponível em: <<http://www.ietf.org/rfc/rfc4972.txt>>. Citado na página 22.
- VASSEUR, J.; ROUX, J. L. *Path Computation Element (PCE) Communication Protocol (PCEP)*. IETF, 2009. RFC 5440 (Proposed Standard). (Request for Comments, 5440). Disponível em: <<http://www.ietf.org/rfc/rfc5440.txt>>. Citado na página 26.
- VMWARE (Ed.). *VMware NSX - The Platform for Network Virtualization*. [S.l.], 2015. Disponível em: <<https://www.vmware.com/files/pdf/products/nsx/VMware-NSX-Datasheet.pdf>>. Citado na página 49.
- WAY, W. I.; JI, P. N.; PATEL, A. N. Contention Resolution within Colorless and Directionless ROADMs. *OFC/NFOEC Technical Digest 2013 OSA*, 2013. Citado na página 15.
- WEI, W.; WANG, C.; YU, J. Cognitive Optical Networks: Key Drivers, Enabling Techniques, and Adaptive Bandwidth Services. *Communications Magazine, IEEE*, IEEE, v. 50, n. 1, p. 106–113, 2012. Citado na página 32.
- WOOD, P. T. Query Languages for Graph Databases. *ACM SIGMOD Record*, ACM, v. 41, n. 1, p. 50–60, 2012. Citado na página 69.

WU, T.-H. Emerging Technologies for Fiber Network Survivability. *Communications Magazine, IEEE, IEEE*, v. 33, n. 2, p. 58–59, 1995. Citado na página 8.

YE, Y.; DIXIT, S.; ALI, M. On Joint Protection/Restoration in IP-centric DWDM Based Optical Transport Networks. *Communications Magazine, IEEE, IEEE*, v. 38, n. 6, p. 174–183, 2000. Citado na página 8.

ZALUSKI, B.; RAJTAR, B.; HABJANIC, H.; BARANEK, M.; SLIBAR, N.; PETRACIC, R.; SUKSER, T. Terastream Implementation of All IP New Architecture. In: IEEE. *Information & Communication Technology Electronics & Microelectronics (MIPRO), 2013 36th International Convention on*. [S.l.], 2013. p. 437–440. Citado na página 41.

ZERVAS, G.; SIMEONIDOU, D. Cognitive Optical Networks: Need, Requirements and Architecture. In: IEEE. *Transparent Optical Networks (ICTON), 2010 12th International Conference on*. [S.l.], 2010. p. 1–4. Citado na página 31.



# Anexos



# ANEXO A – Publicações do autor relevantes para esta tese

Este apêndice apresenta a lista de publicações que formam os pilares desta tese. A Figura 43 apresenta uma visão de como cada publicação pode ser mapeada nos blocos de construção que formam a visão de alto nível de uma arquitetura SDN. As publicações são apresentadas em ordem de importância e relevância no contexto desta tese.

- [A] M. A. Siqueira, F. Hooft, J. Oliveira, E. Madeira, C. E. Rothenberg, “Providing Optical Network as a Service with Policy-based Transport SDN”, *Journal of Network and Systems Management - JNSM*, DOI 10.1007/s10922-014-9320-x, 2014.
- [B] M. A. Siqueira, J. Oliveira, G. Curiel, A. Hirata, F. Hooft, M. Nascimento, C. E. Rothenberg, J. Oliveira, Julio, “An Optical SDN Controller for Transport Network Virtualization and Autonomic Operation”, in *IEEE Globecom Workshops (GC Wkshps)*, pp. 1198-1203, 2013.

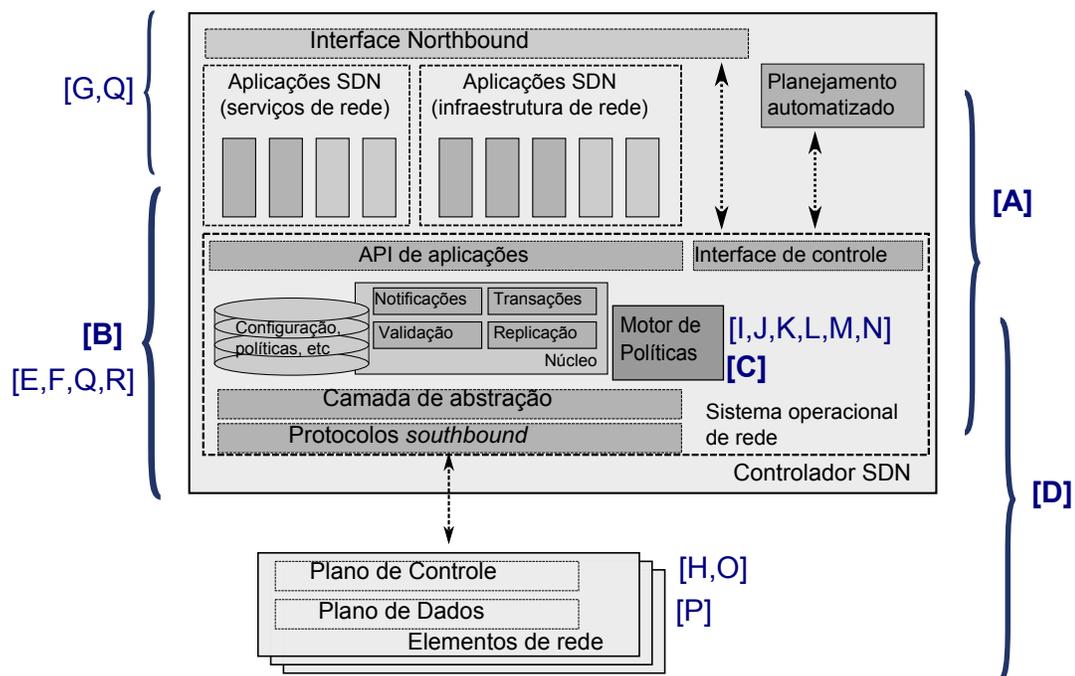


Figura 43 – Artigos publicados pelo autor relacionados com os blocos de construção do modelo SDN

- [C] M. A. Siqueira, F. L. Verdi, R. Pasquini, M. F. Magalhaes, “An Architecture for Autonomic Management of Ambient Networks”, in *Proceedings of Autonomic Networking AN06*, Springer, pp.255-267, 2006.
- [D] M. A. Siqueira, C. E. Rothenberg, “SD-POSS: Um Computador Agnóstico Definido por Software com Suporte à Virtualização Nativa de Funções de Redes Multicamada”, Relatório Técnico, INTRIG (Information & Networking Technologies Research & Innovation Group), FEEC-Unicamp, 2014.
- [E] J. Oliveira, J. Oliveira, M. A. Siqueira, R. Scaraficci, M. Salvador, L. Mariote, N. Guerrero, L. Carvalho, F. Hooft, G. Santos, E. Magalhães, J. Januário, “Towards Software Defined Autonomic Terabit Optical Networks” in *Invited Paper - Optical Fiber Communication Conference, OFC2014*, pp. M3H. 5, 2014.
- [F] J. Oliveira, J. Oliveira, E. Magalhães, J. Januário, M. A. Siqueira, R. Scaraficci, M. Salvador, L. Mariote, N. Guerrero, L. Carvalho, F. Hooft, G. Santos, and M. Garrich, “Towards Terabit Autonomic Optical Networks Based on a Software Defined Adaptive/Cognitive Approach [Invited]”, *J. Opt. Commun. Netw.* 7, A421-A431 (2015).
- [G] J. Oliveira, M. A. Siqueira, G. Curiel, A. Hirata, F. V. Hooft, D. Macedo, M. Colazza, C. E. Rothenberg, “Experimental Testbed of Reconfigurable Flexgrid Optical Network with Virtualized GMPLS Control Plane and Autonomic Controls Towards SDN”, in *Proceedings of International Microwave and Optoelectronics Conference - IMOC2013*, 2013.
- [H] M. A. Siqueira, R. Pasquini, F. L. Verdi, R. Duarte, F. C. Lacerda, M. F. Magalhães, E. Madeira, “Um Mecanismo Baseado em Web Services para Divulgação de Topologias Virtuais Inter-Domínios em Redes GMPLS”, em *Anais do WGRS, SBRC2005*, 2005.
- [I] M. A. Siqueira, F. L. Figueiredo, F. M. Rocha, J. A. Martins, M. C. Castro, “Policy-based Dynamic Reconfiguration of Mobile ad hoc Networks”, in *Proceedings of ICN05*, 2005.
- [J] M. A. de Siqueira, N. A. Nassif, R. A. Resende, M. Lima-Marques, “Policy-Based Architecture for QoS Management in Enterprise IP Networks”, in *Proceedings of IFIP/IEEE International Symposium on Integrated Network Management*, 2005.
- [K] M. A. Siqueira, M. F. Magalhães, L. Farias, M. C. Castro, “A BGP/MPLS PPVPN Management Information Model and a J2EE-based Implementation Architecture for Policy and Web-Based Configuration Management Systems”, in *Proceedings of IEEE ICN04-3rd International Conference on Networking*, 2004.

- 
- [L] M. A. Siqueira, M. F. Magalhães, Eleri Cardozo, “A Policy Management Architecture for MPLS Networks”, in *Proceedings of the 15th International Conference on Parallel and Distributed Computing Systems (PDCS 2002)*, 2002.
- [M] M. A. Siqueira, M. F. Magalhães, Eleri Cardozo, “Implementação de uma Arquitetura de Políticas para Gerência de Redes MPLS”, in *Anais do Workshop de Gerência de Redes e Serviços de Telecomunicações (WTMN2002)*, SBRC, 2002.
- [N] Resende, R.A.; Nassif, N.A.; de Siqueira, M.A.; da Silva, A.E.; Lima-Marques, M., "Quality of Service Control in IP Networks Using Fuzzy Logic for Policy Condition Evaluation," *Fuzzy Systems*, 2005. FUZZ '05. The 14th IEEE International Conference on , vol., no., pp.448,453, 25-25 May 2005, doi: 10.1109/FUZZY.2005.1452435
- [O] A. A. Avancini, A. C. Sachs, J. C. Magro, M. A. Siqueira, M. Rossi, J. P. Oliveira, Implementation of a User to Network Interface for IP/MPLS over WDM Networks, in *Proceedings of the 2003 SBMO/IEEE MTT-S Int. Microwave and Optoelectronics Conference, IMOC03*, 2003.
- [P] M. Garrich, J. R. F. Oliveira, M. A. Siqueira, N. Amaya, G. Zervas, D. Simeonidou, J. C. Oliveira, “Flexibility of Programmable Add/Drop Architecture for ROADMs”, in *Proceedings of Optical Fiber Communication Conference - OFC2014*, 2014.
- [Q] E. Magalhães, J. Oliveira, H. Carvalho, M. Magalhães, M. G. Alabarce, M. A. Siqueira, A. Bordonalli, J. Oliveira, “Global ROADM-Based Spectrum Equalizer in SDN Architecture for QoT Optimization at DWDM Networks”, in *Proceedings of OFC2014*, 2014.
- [R] R. Rosa, M. Siqueira, E. Barea, C. Marcondes, C. Rothenberg, Minicurso: Network Function Virtualization: Perspectivas, Realidades e Desafios, SBRC2014, Florianópolis - Brasil, 2014



# ANEXO B – Logs de execução da ferramenta de planejamento de redes multicamadas

## B.1 Fase 1 - Análise da Topologia

```

...
*** Falhas causaram isolamento de segmento de Rede *****
Links em falha:
id0(1x); id3(2x); id34(3x); id33(2x); id30(4x); id31(3x);
Impedindo Restauracao da Demanda: D9 / CCR / BSB
*** Falhas causaram isolamento de segmento de Rede *****
Links em falha:
id0(1x); id3(2x); id34(3x); id33(2x); id30(4x); id31(3x);
Impedindo Restauracao da Demanda: D10 / CCR / MNS
*** Falhas causaram isolamento de segmento de Rede *****
Links em falha:
id0(1x); id3(2x); id34(3x); id33(2x); id30(4x); id18(5x); id32(3x);
Impedindo Restauracao da Demanda: D11 / POA / CCR
*** Falhas causaram isolamento de segmento de Rede *****
Links em falha:
id0(1x); id3(2x); id34(3x); id33(2x); id30(4x); id18(5x); id32(3x);
Impedindo Restauracao da Demanda: D9 / CCR / BSB
...

```

## B.2 Fase 2 - Dimensionamento da Rede

A seguir é apresentado um resumo do log da ferramenta *Network Planner*, incluindo as etapas de roteamento inicial das demandas, incluindo caminhos de proteção, bem como simulação de falhas duplas em cada região, de forma simultânea, de forma a dimensionar as capacidades necessárias para restauração.

```

*****
*****
** Resultados da primeira etapa - Alocao e falhas duplas**
*****

```

```
*****
////////////////////////////////////
IDs dos NEs de rede:
NE: BSB ID: 16
NE: CCR ID: 0
NE: TSA ID: 11
NE: GYN ID: 15
NE: BEL ID: 12
NE: BOV ID: 14
NE: REC ID: 9
NE: POA ID: 1
NE: FNS ID: 2
NE: RJO ID: 5
NE: SSA ID: 8
NE: PVO ID: 18
NE: CTA ID: 3
NE: CBA ID: 17
NE: BHZ ID: 7
NE: FTL ID: 10
NE: RBC ID: 19
NE: SPO ID: 4
NE: PAL ID: 13
NE: VIX ID: 6
NE: MNS ID: 20
////////////////////////////////////
// Capacidades dos Links para atender demandas 1+1 ///

////////////////////////////////////

////////////////////////////////////Roteamento das Demandas - Resumo //////////////////////////////////
//////////////////////////////////// ROTAS 1+1 das demandas //////////////////////////////////
Demanda: D0 / SPO / RJO-Rota principal: [id6(1x)]
Backup: D0 / SPO / RJO-Rota backup [id36(1x),id10(1x)]

Demanda: D1 / RJO / FTL-Rota principal: [id10(1x),id18(1x)]
Backup: D1 / RJO / FTL-Rota backup [id7(1x),id12(1x),id14(1x),id15(1x)]

Demanda: D2 / BSB / FTL-Rota principal: [id18(1x)]
Backup: D2 / BSB / FTL-Rota backup [id35(1x),id32(1x),id22(1x),id20(1x)]
```

Demanda: D3 / BSB / SPO-Rota principal: [id36(1x)]

Demanda: D4 / BSB / RJO-Rota principal: [id10(2x)]

Backup: D4 / BSB / RJO-Rota backup [id36(2x),id6(1x)]

Demanda: D5 / SPO / FTL-Rota principal: [id36(3x),id18(2x)]

Backup: D5 / SPO / FTL-Rota backup [id8(1x),id13(1x),d14(1x),id15(1x)]

Demanda: D6 / POA / SPO-Rota principal: [id3(1x)]

Demanda: D7 / BHZ / RJO-Rota principal: [id8(1x),id6(2x)]

Backup: D7 / BHZ / RJO-Rota backup [id11(1x),id10(3x)]

Demanda: D8 / FTL / MNS-Rota principal: [id19(1x),id24(1x)]

Backup: D8 / FTL / MNS-Rota backup [id20(1x),id22(1x),id25(1x)]

Demanda: D9 / CCR / BSB-Rota principal: [id37(1x),id36(4x)]

Demanda: D10 / CCR / MNS-Rota principal: [id30(1x),id31(1x),id25(1x)]

Demanda: D11 / POA / CCR-Rota principal: [id2(1x),id34(1x)]

Demanda: D12 / BHZ / MNS-Rota principal: [id11(1x),id18(3x),id19(1x),id24(1x)]

Demanda: D13 / POA / BHZ-Rota principal: [id3(1x),id8(2x)]

////////////////////////////////////

Simulando double Failures da secao 1 com a 2 com a 3

//Capp. dos Links para atender demandas 1+1 + rest. de falhas duplas//

L32;GYN/PAL;Cap:200;Res:100;Princ:0;Prot:100;Bw Rest:100

L14;SSA/REC;Cap:500;Res:300;Princ:0;Prot:200;Bw Rest:220

L21;TSA/BEL;Cap:500;Res:500;Princ:0;Prot:0;Bw Rest:420

L15;REC/FTL;Cap:500;Res:300;Princ:0;Prot:200;Bw Rest:220

L24;BOV/MNS;Cap:200;Res:120;Princ:80;Prot:0;Bw Rest:100

L3;POA/SPO;Cap:200;Res:120;Princ:80;Prot:0;Bw Rest:120

L35;BSB/GYN;Cap:300;Res:200;Princ:0;Prot:100;Bw Rest:180

L18;FTL/BSB;Cap:400;Res:60;Princ:340;Prot:0;Bw Rest:40

L22;BEL/PAL;Cap:400;Res:260;Princ:0;Prot:140;Bw Rest:240

L27;PVO/RBC;Cap:100;Res:100;Princ:0;Prot:0;Bw Rest:80

L11;BHZ/BSB;Cap:500;Res:420;Princ:40;Prot:40;Bw Rest:400

L4;CTA/SPO;Cap:200;Res:200;Princ:0;Prot:0;Bw Rest:200

L2;POA/CTA;Cap:200;Res:160;Princ:40;Prot:0;Bw Rest:120

L23;BOV/RBC;Cap:100;Res:100;Princ:0;Prot:0;Bw Rest:80

L17;SSA/TSA;Cap:500;Res:500;Princ:0;Prot:0;Bw Rest:420

```
L31;CBA/PAL;Cap:400;Res:360;Princ:40;Prot:0;Bw Rest:340
L7;RJO/VIX;Cap:500;Res:400;Princ:0;Prot:100;Bw Rest:320
L5;FNS/CTA;Cap:200;Res:200;Princ:0;Prot:0;Bw Rest:200
L28;RBC/MNS;Cap:100;Res:100;Princ:0;Prot:0;Bw Rest:80
L6;SPO/RJO;Cap:600;Res:360;Princ:140;Prot:100;Bw Rest:280
L0;POA/FNS;Cap:200;Res:200;Princ:0;Prot:0;Bw Rest:160
L34;CCR/CTA;Cap:200;Res:160;Princ:40;Prot:0;Bw Rest:120
L20;FTL/BEL;Cap:400;Res:260;Princ:0;Prot:140;Bw Rest:200
L37;CCR/SPO;Cap:400;Res:360;Princ:40;Prot:0;Bw Rest:320
L36;BSB/SPO;Cap:500;Res:60;Princ:240;Prot:200;Bw Rest:40
L16;REC/TSA;Cap:100;Res:100;Princ:0;Prot:0;Bw Rest:100
L10;RJO/BSB;Cap:600;Res:260;Princ:200;Prot:140;Bw Rest:240
L26;PVO/MNS;Cap:100;Res:100;Princ:0;Prot:0;Bw Rest:100
L8;SPO/BHZ;Cap:500;Res:320;Princ:80;Prot:100;Bw Rest:240
L1;FNS/SPO;Cap:200;Res:200;Princ:0;Prot:0;Bw Rest:200
L13;BHZ/SSA;Cap:500;Res:400;Princ:0;Prot:100;Bw Rest:320
L33;CCR/GYN;Cap:200;Res:200;Princ:0;Prot:0;Bw Rest:180
L29;PVO/CBA;Cap:200;Res:200;Princ:0;Prot:0;Bw Rest:120
L19;FTL/BOV;Cap:200;Res:120;Princ:80;Prot:0;Bw Rest:100
L30;CCR/CBA;Cap:400;Res:360;Princ:40;Prot:0;Bw Rest:340
L25;PAL/MNS;Cap:200;Res:120;Princ:40;Prot:40;Bw Rest:120
L9;VIX/BHZ;Cap:400;Res:400;Princ:0;Prot:0;Bw Rest:340
L12;VIX/SSA;Cap:500;Res:400;Princ:0;Prot:100;Bw Rest:320
////////////////////////////////////
Total de Interfaces 100G Rede: 248
////////////////////////////////////
```