

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
ENGENHARIA ELETRÔNICA E DE TELECOMUNICAÇÕES

DOUGLAS RAFAELL PEREIRA

REDES ÓPTICAS DEFINIDAS POR SOFTWARE: UM ESTUDO DO CONTROLE DE
EQUIPAMENTOS NETCONF/YANG USANDO OS MODELOS OPENROADM

PATOS DE MINAS
2020

DOUGLAS RAFAELL PEREIRA

REDES ÓPTICAS DEFINIDAS POR SOFTWARE: UM ESTUDO DO CONTROLE DE
EQUIPAMENTOS NETCONF/YANG USANDO OS MODELOS OPENROADM

Trabalho apresentado como um dos requisitos parciais para aprovação na disciplina de Trabalho de Conclusão de Curso I do curso de Engenharia Eletrônica e de Telecomunicações da Universidade Federal de Uberlândia – Campus Patos de Minas.

Orientador: Prof. Dr. André Luiz Aguiar da Costa

PATOS DE MINAS
2020

DOUGLAS RAFAELL PEREIRA

**REDES ÓPTICAS DEFINIDAS POR SOFTWARE: UM ESTUDO DO CONTROLE DE
EQUIPAMENTOS NETCONF/YANG USANDO OS MODELOS OPENROADM**

Trabalho apresentado como um dos requisitos parciais
para aprovação na disciplina de Trabalho de Conclusão
de Curso II do curso de Engenharia Eletrônica e de
Telecomunicações da Universidade Federal de
Uberlândia – Campus Patos de Minas.

Patos de Minas, 18 de dezembro de 2020.

COMISSÃO EXAMINADORA:

Prof. Dr. André Luiz Aguiar da Costa
Universidade Federal de Uberlândia
Orientador

Prof. Dr. Alexandre Coutinho Mateus
Universidade Federal de Uberlândia
Examinador

Dra. Indayara Bertoldi Martins
Ekinops SA
Examinador

Resumo

A constante evolução e aumento de equipamentos de rede para satisfazer a demanda de tráfego vem influenciando os provedores de serviços de telecomunicações a adotar soluções alternativas de forma a automatizar as operações e a reduzir o custo de monitoração. Para isso, as redes de transporte óptica são aplicadas com conceitos de redes definidas por software vem se tornando uma alternativa viável pois, além de separar a camada de controle da camada de dados, permite a construção da topologia unificada e flexibiliza a configuração de equipamentos de diferentes fabricantes. Para isso, é necessário desagregar os equipamentos e abstrair suas funcionalidades por modelos de dados padronizados. Os modelos OpenROADM vem se tornando uma opção viável para desagregar os elementos da rede de transporte óptica porque providencia um grau bem realístico de funções, camadas, interfaces, alarmes e métricas do domínio óptico, necessário para otimizar a utilização de recurso. Este trabalho tem o objetivo de estudar como os modelos OpenROADM desagregarão os equipamentos ópticos e suas funções.

Palavras-chaves: SDN, SDN transporte, redes ópticas desagregadas, NETCONF/YANG, redes de transporte ópticas, OpenROADM, ConfD.

Abstract

The constant evolution and increasing over the network elements to keep the traffic demand has made service providers to search for other alternative solutions in a way to automate network operations and reduce the monitoring cost. Therefore, the optical transport network integrated with software defined network concepts has become an alternative solution because it allows the unification of the network topology, adapts the configuration on different equipment vendors, and separate the control and data plans. To do so, the network elements must be disaggregated e its functionalities be abstracted by standard or common data models. OpenROADM has become a feasible solution to disaggregate network elements in the optical transport network because it provides a deep and realist optical domain level for functions, layers, interfaces, alarms and metrics, which it is needed to optimize the resource utilization. The objective of this paper is to study how the OpenROADM models will disaggregate the network equipment and its functions.

Keywords: SDN, transport SDN, disaggregated optical networks, NETCONF/YANG, optical transport network, OpenROADM, ConfD.

Agradecimentos

Agradeço a minha família por ter me abrigado e me dado todo suporte necessário para que eu pudesse prosseguir meus estudos superiores. A motivação de não desistir de estudar foi o que me impulsionou a fazer o vestibular, mesmo sabendo da remota possibilidade da admissão. Por isso, em especial gostaria de agradecer minha tia, a Gislene, guerreira, sonhadora, com personalidade única que me auxiliou bastante nessa jornada.

Agradeço também a oportunidade de aprender e evoluir com os professores da faculdade em diversas disciplinas. Foi um ambiente admirável para estimular o aprendizado apesar das dificuldades iniciais que passamos juntos no primeiro semestre, mas o importante é que nunca desistimos e lutamos juntos para conseguir melhores condições. Acredito que todos os professores foram essenciais para minha formação, e quero deixar um agradecimento especial para os professores: Dra. Elise, Dr. William, Dra. Karine, Dra. Gláucia, Dra. Fabrícia, Dr. Alexandre, Dr. Laurence e Dr. Peterson.

Agradeço a Dra. Magda da Universidade da Califórnia de Irvine por me permitir o estágio na área de rede de computadores em seu laboratório.

Agradeço ao meu orientador, Dr. André, por ter me indicado na empresa Padtec. Consegui um estágio na empresa e foi quando comecei a minha carreira profissional. Sem a indicação dele, isso não seria possível. Por isso, muito obrigado também por ter a paciência e tempo para me orientar nesse trabalho que tenho orgulho.

Agradeço à Universidade Federal de Uberlândia por me conceder a oportunidade de realizar meu curso de graduação na faculdade de Engenharia Elétrica (FEELT).

Agradeço ao governo brasileiro e a CAPES por ter oferecido o programa Ciência sem Fronteiras que pude participar. Foi uma experiência de um ano e meio incrível que contribuiu na minha carreira pessoal e profissional. Pude aprender inglês, estudar disciplinas em faculdade de excelência e prestar estágio na área de redes de computadores.

Sumário

Capítulo 1	13
1.1. Problematização	18
1.2. Objetivo Geral.....	19
1.2.1. Objetivos Específicos	19
1.3. Justificativa	19
1.4. Proposta e Organização do Trabalho	19
Capítulo 2	21
2.1. OpenDayLight	26
2.2. ONOS.....	27
Capítulo 3	29
3.1. Rede de Transporte Aberta e Desagregada (ODTN)	34
3.2. ROADM.....	36
Capítulo 4	39
4.1. Preparação dos Simuladores	39
4.2. Modelo de Informações Gerais: Módulo <i>info</i>	41
4.3. Representação do Equipamento tipo ROADM.....	44
4.4. Modelo do Sub-Rack: Módulo <i>shelves</i>	46
4.5. Modelo da Placa: Módulo <i>circuit-pack</i>	48
4.5.1. Modelo <i>circuit-pack</i> : Amplificador.....	49
4.5.2. Modelo <i>circuit-pack</i> : WSS	50
4.5.3. Modelo <i>circuit-pack</i> : MUX/DEMUX	52
4.5.4. Modelo da Posição do Sub-Rack: Módulo <i>slot-info</i>	53
4.6. Modelo da Direção: Módulo <i>degree</i>	55
4.7. Modelo do SRG: Módulo <i>shared-risk-group</i>	57
4.8. Modelo das Ligações Internas: Módulo <i>internal-link</i>	58
4.9. Modelo das Ligações Físicas: Módulo <i>physical-link</i>	60
4.10. Modelo das Ligações Externas: Módulo <i>external-link</i>	64
4.11. Representação do Equipamento tipo Transponder	65
Capítulo 5	75
5.1. Simulação.....	75
5.1.1. Ambiente da Simulação.....	75
5.1.2. Nó Transponder de Transmissão	76
5.1.3. Nó ROADM da Ligação de Adição	83
5.1.4. Nó ROADM da Ligação Expressa	93
5.1.5. Nó ROADM da Ligação da Remoção.....	96
5.1.6. Nó Transponder de Recepção.....	99
5.2. Resultados	100
Capítulo 6	102
6.1. Conclusões	102
6.2. Estudos Futuros.....	103

Lista de Figuras

Figura 1 – Camadas da arquitetura SDN.	22
Figura 2 – Arquitetura SDN distribuída de controladores.	23
Figura 3 – Arquitetura do controlador OpenDayLight.	26
Figura 4 – Arquitetura do controlador ONOS.	27
Figura 5 – Redes de transporte convencionais.	29
Figura 6 – Arquitetura SDON para redes de transporte usando OpenFlow+ e OpenFlow.	31
Figura 7 – Arquitetura HT-SDN para rede de transportes de múltiplos domínios.	32
Figura 8 – Diagrama de blocos resumido do projeto ODTN.	35
Figura 9 – Diagrama de blocos de um equipamento ROADM.	36
Figura 10 – Nó ROADM de 2 graus ou direções com banco de adição e remoção.	38
Figura 11 – Diagrama de blocos de um nó ROADM.	44
Figura 12 – Composição de elementos DWDM dos blocos de direção e SRG em um nó ROADM.	45
Figura 13 – Placas físicas instaladas em um sub-rack de 20 posições.	46
Figura 14 – Ligações internas de cada tipo de <i>circuit-pack</i> (a) amplificador, (b) ROADM e (c) MUX/DEMUX.	59
Figura 15 – Ligações físicas entre os blocos de direções e SRG de um nó ROADM.	61
Figura 16 – Ligações externas entre dois nós ROADMs de localizações diferentes.	64
Figura 17 – Representação em blocos de <i>circuit-packs</i> e portas do nó transponder modelado. ...	66
Figura 18 – Ambiente de simulação.	76
Figura 19 – Interface OCh da porta rede do nó transponder.	77
Figura 20 – Interface OTU da porta rede do nó transponder.	79
Figura 21 – Interface ODU da porta rede do nó transponder.	80
Figura 22 – Interface ODU da porta cliente do nó transponder.	82
Figura 23 – Crossconexão entre as interfaces ODUs do nó transponder.	83
Figura 24 – Representação das terminações lógicas das ligações de adição do nó SP-ROADM-1.	84
Figura 25 – Exemplo das entidades canal de mídia (MC) e canal de mídia da rede (NMC).	86
Figura 26 – Interfaces OMS e OTS no bloco de direção.	89
Figura 27 – Interface MC-TTP do bloco de direção.	90
Figura 28 – Interface NMC-CTP do bloco de direção.	91

Figura 29 – Crossconexão entre as interfaces NMC-CTP do bloco de direção.	92
Figura 30 – Configuração física do nó BH-ROADM-1.	93
Figura 31 – Crossconexão e interfaces dos blocos de direção.	96
Figura 32 – Ligação de remoção do bloco de direção 2 do nó RIO-ROADM-1	96
Figura 33 – Representação de todas as camadas do serviço fim-a-fim ODU4.	101

Lista de Quadros

Quadro 1 – Lista dos arquivos YANG utilizados na simulação.....	40
Quadro 2 – Iniciação do ConfD e da sua interface CLI.	41
Quadro 3 – Parte do modelo YANG módulo <i>info</i>	41
Quadro 4 – Árvore YANG reduzida de atributos do módulo <i>info</i>	42
Quadro 5 – Mensagem RPC do NETCONF cliente para consulta do módulo <i>info</i>	43
Quadro 6 – Resposta RPC do ConfD com as todas as informações do módulo <i>info</i>	43
Quadro 7 – Árvore YANG reduzida de atributos do módulo <i>shelves</i>	46
Quadro 8 – Mensagem RPC do NETCONF cliente para consulta do módulo <i>shelves</i>	47
Quadro 9 – Resposta RPC do ConfD com as todas as informações do módulo <i>shelves</i>	47
Quadro 10 – Árvore YANG reduzida de atributos do módulo <i>circuit-pack</i>	48
Quadro 11 – Resposta RPC do ConfD com informações do <i>circuit-pack</i> Dir1-RxAMP.	49
Quadro 12 – Resposta RPC do ConfD com informações do <i>circuit-pack</i> Dir1-WSS.	50
Quadro 13 – Resposta RPC do ConfD com informações do <i>circuit-pack</i> Dir1-MUX-DEMUX.	52
Quadro 14 – Resposta RPC do ConfD do módulo <i>shelves</i> com informações de posições.	54
Quadro 15 – Árvore YANG reduzida de atributos do módulo <i>degree</i>	55
Quadro 16 – Resposta RPC do ConfD com as todas as informações do módulo <i>degree</i>	55
Quadro 17 – Resposta RPC do ConfD com as todas as informações do módulo <i>mc-capability-profile</i>	56
Quadro 18 – Árvore YANG reduzida de atributos do módulo <i>shared-risk-group</i>	57
Quadro 19 – Resposta RPC do ConfD com as todas as informações do módulo <i>shared-risk-group</i>	57
Quadro 20 – Resposta RPC do ConfD com informações do módulo <i>internal-link</i>	59
Quadro 21 – Árvore YANG reduzida de atributos do módulo <i>physical-link</i>	62
Quadro 22 – Resposta RPC do ConfD com as informações do módulo <i>physical-link</i>	62
Quadro 23 – Resposta RPC do ConfD com as informações do módulo <i>external-link</i>	65
Quadro 24 – Resposta RPC do ConfD com as informações do módulo <i>circuit-pack</i> do transponder.	67
Quadro 25 – Resposta RPC adaptada do ConfD com as informações de capacidades do transponder.	68
Quadro 26 – Resposta RPC adaptada do ConfD com as informações de capacidade das portas.	70
Quadro 27 – Resposta RPC adaptada do ConfD com as informações de caixas de formato.	72

Quadro 28 – Resposta RPC do ConfD com as informações da matriz de crossconexão.	73
Quadro 29 – Resposta RPC do ConfD com as informações do módulo transponder.	74
Quadro 30 – Resposta RPC adaptada do ConfD para a interface OCh da porta rede.	77
Quadro 31 – Resposta RPC adaptada do ConfD para a interface OTU da porta rede.	78
Quadro 32 – Resposta RPC adaptada do ConfD para a interface ODU da porta rede.	80
Quadro 33 – Resposta RPC adaptada do ConfD para a interface ODU da porta cliente.	81
Quadro 34 – Resposta RPC do ConfD da crossconexão de interfaces do nó SP-Transponder-1.	82
Quadro 35 – Resposta RPC do ConfD com dados de terminações lógicas nas portas de <i>circuit-packs</i>	85
Quadro 36 – Resposta RPC do ConfD para a interface NMC-CTP do bloco SRG 1.	87
Quadro 37 – Resposta RPC do ConfD para a interface OTS do bloco de direção 4.	88
Quadro 38 – Resposta RPC do ConfD para a interface OMS do bloco de direção 4.	89
Quadro 39 – Resposta RPC do ConfD para a interface MC-TTP do bloco de direção 4.	90
Quadro 40 – Resposta RPC do ConfD para a interface NMC-TTP do bloco de direção 4.	91
Quadro 41 – Resposta RPC do ConfD da crossconexão de interfaces do nó SP-ROADM-1.	92
Quadro 42 – Respostas RPC do ConfD para as interfaces da porta TTP-RX da direção 1.	94
Quadro 43 – Resposta RPC do ConfD da crossconexão de interfaces do nó BH-ROADM-1.	95
Quadro 44 – Respostas RPC do ConfD para as interfaces das portas do nó RIO-ROADM-1.	97
Quadro 45 – Resposta RPC do ConfD da crossconexão de interfaces do nó RIO-ROADM-1.	99
Quadro 46 – Resposta RPC do ConfD da crossconexão de interfaces do nó RIO-Transponder-1.	100

Lista de Siglas

Sigla	Definição (<i>Definição em Inglês</i>)
ANSI	Instituto Nacional de Padrões Americano (<i>American National Standards Institute</i>)
API	Interface de Programação de Aplicativos (<i>Application Programming Interfaces</i>)
BGP	<i>Border Gateway Protocol</i>
CFP	<i>C Form-Factor Pluggable</i>
CLI	Interfaces de Linhas de Comandos (<i>Command Line Interface</i>)
CO-OFDM	Técnica Óptica Coerente de Multiplexação Ortogonal de Divisão de Frequência (<i>Coherent Optical Orthogonal Frequency-Division Multiplexing</i>)
CP	Ponto de Conexão (<i>Connection Point</i>)
CTP	Ponto de Terminação de Conexão (<i>Connection Termination Point</i>)
DAPI	Identificadores de Ponto de Acesso de Destino (<i>Destination Access Point Identifiers</i>)
DEMUX	Demultiplexador (<i>Demultiplexer</i>)
DP-QPSK	Chaveamento por Deslocamento de Fase em Quadratura de Duas Polarizações (<i>Dual-Polarization Quadrature Phase-Shift Keying</i>)
DSF	Fibra de Dispersão Deslocada (<i>Dispersion-shifted Fibers</i>)
DSP	Processador de Sinal Digital (<i>Digital Signal Processor</i>)
DWDM	Multiplexação Densa por Comprimento de Onda (<i>Dense Wavelength Division Multiplexing</i>)
EDFA	Amplificadores Ópticos de Fibra Dopada de Érbio (<i>Erbium-Doped Fiber Amplifier</i>)
FEC	Código de Correção de Erro (<i>Forward Error Correction</i>)
ForCES	Separação do Elemento de Controle e Envio (<i>Forwarding and Control Element Separation</i>)
FWM	Efeito de Mistura de Quatro Ondas (<i>Four Wave Mixing</i>)
HT-SDN	Redes de Transporte Definidas por Software Hierárquica (<i>Hierarchical Transport Defined Network</i>)

IP	Protocolo de Internet (<i>Internet Protocol</i>)
ITU	União Internacional de Telecomunicações (<i>International Telecommunication Union</i>)
MC	Interfaces de Canal de Mídia (<i>Media Channel</i>)
MC-TTP	Interfaces de Canal de Mídia - Ponto de Terminação de Trilha (<i>Media Channel - Trail Termination Point</i>)
MIB	Base de Informação de Gerenciamento (<i>Management Information Base</i>)
MPLS	Comutação de Rótulos Multiprotocolo (<i>Multiprotocol Label Switching</i>)
MUX	Multiplexador (<i>Multiplexer</i>)
NBI	Interface de Comunicação Norte (<i>NorthBound Interface</i>)
NDSF	Fibra de Dispersão Deslocada não nula (<i>Non-dispersion Shifted Fiber</i>)
NE	Elementos da Rede (<i>Network Element</i>)
NETCONF	Protocolo de Configuração de Redes (<i>Network Configuration Protocol</i>)
NMC	Interfaces de Canal de Mídia da Rede (<i>Network Media Channel</i>)
NMC-CTP	Interfaces de Canal de Mídia da Rede - Ponto de Terminação de Conexão (<i>Network Media Channel - Connection Termination Point</i>)
NMS	Sistema de Gerenciamento de Redes (<i>Network Management System</i>)
NOS	Sistema Operacional da Rede (<i>Network Operating System</i>)
NSO	Orquestrador de Serviços da Rede (<i>Network Service Orchestration</i>)
OADM	Multiplexador de Adição e Remoção Óptica (<i>Optical Add Drop Multiplexer</i>)
OCh	Camada de Canal Óptica (<i>Optical Channel</i>)
ODTN	Redes de Transporte Aberta e Desagregada (<i>Open and Disaggregated Transport Network</i>)
ODU	Camada de Dados Óptica (<i>Optical Data Unit</i>)
OF+	OpenFlow Estendido (<i>Extended OpenFlow</i>)
OLA	Amplificador de Linha Óptico (<i>Optical Line Amplifier</i>)
OLT	Terminal de Linha Óptico (<i>Optical Line Terminal</i>)
OMS	Camada de Multiplexação Óptica (<i>Optical Multiplexed Section</i>)
ONF	Fundação Aberta de Redes (<i>Open Network Foundation</i>)
ONOS	Sistema Operacional de Rede Aberta (<i>Open Network Operating System</i>)
OPU	Camada de Carga Óptica (<i>Optical Payload Unit</i>)
OSNR	Relação Sinal-Ruído Óptico (<i>Optical Signal to Noise Ratio</i>)

OSS	Sistemas de Suporte de Operações (<i>Operations Support System</i>)
OTN	Rede de Transporte Óptico (<i>Optical Transport Network</i>)
OTS	Camada da Seção de Transmissão Óptica (<i>Optical Transmission Section</i>)
OTU	Camada de Transporte Óptica (<i>Optical Transport Unit</i>)
OXC	Crossconector Óptico (<i>Optical Crossconnect</i>)
PDH	Hierarquia Digital Plesiócrona (<i>Plesiochronous Digital Hierarchy</i>)
PP	Ponto de Pares de Portas (<i>Port Pairs</i>)
QAM	Modulação de Amplitude em Quadratura (<i>Quadrature Amplitude Modulation</i>)
QPSK	Chaveamento por Deslocamento de Fase em Quadratura (<i>Quadrature Phase-Shift Keying</i>)
REST	<i>Representational State Transfer</i>
ROADM	Multiplexador de Adição e Remoção Óptica Reconfigurável (<i>Reconfigurable Optical Add Drop Multiplexer</i>)
RPC	Procedimentos de Chamadas Remotas (<i>Remote Procedure Call</i>)
SAPI	Identificadores de Ponto de Acesso de Origem (<i>Source Access Point Identifiers</i>)
SBI	Interfaces de Comunicação Sul (<i>SouthBound Interface</i>)
SDH	Hierarquia Digital Síncrona (<i>Synchronous Digital Hierarchy</i>)
SDN	Redes Definidas por Software (<i>Software Defined Network</i>)
SDON	Redes de Transporte Definidas por software monolítica (<i>Software Defined Optical Networking</i>)
SFP	<i>Small form-factor Pluggable Transceiver</i>
SFP+	<i>Small form-factor Pluggable Transceiver plus</i>
SNMP	Protocolo Simples de Gerência de Rede (<i>Simple Network Management Protocol</i>)
SONET	Rede Óptica Síncrona (<i>Synchronous Optical Network</i>)
SRG	Grupo Compartilhado de risco (<i>Shared Risk Group</i>)
TAPI	Interface de Programação de Aplicativos de Transporte (<i>Transport Application Programming Interfaces</i>)
TDM	Técnica de Multiplexação por Divisão do Tempo (<i>Time Division Multiplexing</i>)
T-SDN	Redes de Transporte definidas por Software (<i>Transport Defined Network</i>)

TTP	Ponto de Terminação de Trilha (<i>Trail Termination Point</i>)
WDM	Técnica de Multiplexação por Divisão de Comprimento de Onda (<i>Wavelength Division Multiplexing</i>)
WSS	Comutador Seletivo de Comprimento de Onda (<i>Wavelength Selective Switch</i>)
XML	<i>Extensible Markup Language</i>
YANG	<i>Yet Another Next Generation Data Modeling Language</i>

Capítulo 1

Introdução

As primeiras redes de transporte foram implantadas para interligar os equipamentos do sistema de telegrafia: os telégrafos e os comutadores analógicos [1]. No século 19, foi demonstrado a transmissão de uma mensagem telegráfica em uma rede de transporte de aproximadamente 80 km nos Estados Unidos [2]. Dessa forma, as redes de transporte foram projetadas para estabelecer comunicação regional e nacional entre as estações de base nas cidades, servindo como um enlace de comunicação por centena de quilômetros.

Por volta de 1960 as redes de transporte utilizaram os sistemas de transmissão analógica e digital por cabo coaxial e par trançado para transportar sinais de voz [1]. Devido à alta sensibilidade ao ruído desses meios físicos, as redes de transporte necessitavam de vários estágios de regeneração e amplificação, tipicamente existiam repetidores a cada 1 km [3]. Portanto o custo de instalação e manutenção das redes de transporte era elevado. Além disso, os sinais de voz passaram a serem transmitidos nas redes de transporte com a técnica de multiplexação por divisão do tempo (TDM, *Time Division Multiplexing*), onde cada sinal de voz compartilha o mesmo meio físico em intervalos de tempo específicos [1].

A hierarquia digital plesiócrona (PDH, *Plesiochronous Digital Hierarchy*) foi a primeira arquitetura padronizada para organizar os níveis de multiplexação assíncrona dos canais de voz, que eram transmitidos com a técnica TDM, onde o canal de menor hierarquia é igual a 64 kbps. Japão, Europa e a América do Norte fizeram sua própria especificação de PDH, portanto, equipamentos de interoperabilidade foram desenvolvidos e instalados para interligar as redes de transporte de diferentes padrões PDH. Consequentemente, o custo de infraestrutura das redes de transporte era alto e o mercado era dividido entre regiões [1]. Na Europa, os sistemas eram capazes de agregar até 30 canais de voz somando uma taxa de 2.048 Mbps, enquanto no Japão e na América do Norte a hierarquia multiplexava até 24 canais gerando 1.544 Mbps [3].

A primeira evolução comercial das redes de transporte caracterizou-se pela utilização dos componentes ópticos por volta de 1980, destacando-se a invenção do laser semiconductor operando na faixa de 800 nm como transmissor e da fibra óptica como meio físico [4]. A fibra óptica tornou-se atrativa após 1966 quando Charles K. Kao e George Hockham provaram que era possível construir uma fibra óptica com perda característica de 20 dB/km [5]. Logo depois em

1970, a empresa Corning desenvolveu a primeira fibra óptica com atenuação de 20 dB/km [6]. Sendo assim, as redes de transporte ópticas necessitam de poucos estágios de regeneração e amplificação em comparação com as redes com sistemas de par trançado e de cabo coaxial. De certa forma, o custo da infraestrutura e de manutenção é menor e incentivou o mercado a aderir a tecnologia de comunicação óptica [1]. Nesse período, as redes de transporte óptico operavam à uma taxa de 45 Mbps com repetidores a cada 10 km [3].

Após o aprendizado dos problemas encontrados com o PDH, o Instituto Nacional de Padrões Americano (ANSI, *American National Standards Institute*) e a União Internacional de Telecomunicações (ITU, *International Telecommunication Union*) especificaram respectivamente os protocolos de transporte: Rede Óptica Síncrona (SONET, *Synchronous Optical Network*) e o Hierarquia Digital Síncrona (SDH, *Synchronous Digital Hierarchy*) para o controle da rede de transporte óptico, por volta de 1980, definindo portanto, a estrutura do quadro síncrono para transmitir sinais digitais TDM [1]. Tais hierarquias apresentam protocolos com taxas compatíveis permitindo a direta interoperabilidade entre os equipamentos, com funções específicas para o monitoramento e o gerenciamento da rede, facilitando a localização efetiva de falhas [3]. O SDH, então, tornou-se o protocolo de transporte predominante nas redes dos provedores de serviços de telecomunicações, apesar que o SONET foi mais utilizado na América do Norte [6].

Ainda por volta de 1980, a comunidade científica posteriormente desenvolveu e demonstrou sistemas ópticos operando na região de 1300 nm, onde a perda da fibra era abaixo de 1 dB/km e a dispersão era menor [4]. Por isso, desencadeou-se o desenvolvimento de lasers e detectores ópticos operando nessa faixa de 1300 nm [5], combinado paralelamente com o desenvolvimento das fibras ópticas monomodo padrão G.652 [7]. Desta forma, tais sistemas comerciais suportavam taxas de 1.7 Gbps com repetidores a cada 50 km [3].

Na terceira geração de sistemas ópticos, foi possível desenvolver sistemas comerciais operando à uma taxa de 2.5 Gbps com repetidores a cada 100 km, taxa equivalente aos níveis específicos de multiplexação dos quadros de transporte SDH de STM-16 e do SONET de OC-48 [3]. Além do mais, a comunidade científica levantou que os sistemas ópticos coerentes poderiam aumentar a distância entre os estágios de regeneração por causa da alta sensibilidade dos receptores coerentes, capazes de recuperar a fase do sinal óptico e também introduzir técnicas de modulação. Entretanto, não houve aderência da comunidade para desenvolver sistemas ópticos coerentes nessa geração [8]. O principal motivo para a não aderência foi o surgimento dos amplificadores ópticos de fibra dopada de érbio (EDFA, *Erbium-Doped Fiber Amplifier*), o qual, era capaz de compensar a atenuação da fibra amplificando parte do espectro óptico, especificamente nas banda C (1530 a

1565 nm) e na banda L (1565 a 1625 nm), sem a necessidade de converter o sinal do domínio elétrico para o domínio óptico e vice-versa [3].

A invenção dos amplificadores ópticos e do emprego da técnica de multiplexação por divisão de comprimento de onda (WDM, *Wavelength Division Multiplexing*), caracterizou a terceira evolução da rede de transporte óptico, por volta de 1990. A técnica WDM consiste em transmitir canais em diferentes comprimentos de onda em uma única fibra óptica e com o avanço da tecnologia devido ao aumento do número de canais transmitidos, o termo multiplexação densa por comprimento de onda (DWDM, *Dense Wavelength Division Multiplexing*) surgiu [4]. O instituto ITU padronizou o espaçamento entre canais e as faixas de comprimentos de onda na norma G.694.1 [9].

Além das regiões de 800 nm e 1300 nm, a região de 1550 nm surgiu como uma oportunidade para aumentar o produto taxa versus distância pois tal região oferecia baixa atenuação, e contudo, descobriu-se que a tecnologia óptica da época apresentava alta dispersão nessa região [4]. Para resolver esse problema, foram desenvolvidos o laser com modo único longitudinal e a fibra de dispersão deslocada (DSF, *Dispersion-shifted Fibers*) especificada pela norma G.653 [10]. Os amplificadores EDFA impulsionaram a evolução da rede de transporte óptico para a região de 1550 nm devido a sua operação nas bandas C e L [5], e deste modo, contribuíram significativamente para a redução do custo dos repetidores nas redes de transporte óptico pois sua amplificação é totalmente óptica. Os sistemas precisavam tipicamente de um repetidor a cada 70 a 80 km [6].

Os elementos tradicionais de uma rede DWDM são: o terminal de linha óptico (OLT, *Optical Line Terminal*) também conhecido como transponder, o multiplexador de adição e remoção óptica (OADM, *Optical Add Drop Multiplexer*) que pode ser reconfigurável (ROADM, *Reconfigurable Optical Add Drop Multiplexer*), o amplificador de linha óptico (OLA, *Optical Line Amplifier*), o multiplexador (MUX, *Multiplexer*), o demultiplexador (DEMUX, *Demultiplexer*), e por fim, o crossconector óptico (OXC, *Optical Crossconnect*) [6] [11]. Primeiramente, o transponder é responsável por ajustar o comprimento de onda do sinal e espaçamento adequado. O OADM permite o controle dos canais ópticos através da retirada e da adição de comprimentos de ondas específicos e o ROADM tem essa mesma função só que permitindo operações remotas de reconfiguração e são projetados com um módulo comutador seletivo de comprimento de onda (WSS, *Wavelength Selective Switch*) servindo como uma matriz de comutação no domínio óptico. Já o MUX e o DEMUX são equipamentos passivos, que não energizados geralmente, responsáveis por respectivamente unir e separar os diferentes comprimentos de onda na mesma fibra óptica. Por fim, o OXC é um elemento da rede que permite o roteamento de um conjunto de comprimento de

onda de entrada para outro conjunto na saída, de modo a estabelecer conexões ópticas em um sistema DWDM [11].

Ao mesmo tempo, devido ao crescimento exponencial do tráfego por causa da popularização dos serviços da Internet, um novo protocolo de transporte foi especificado pelo instituto ITU denominado rede de transporte óptico (OTN, *Optical Transport Network*) com o objetivo de facilitar o gerenciamento, controle e manutenção das redes ópticas [1]. A arquitetura genérica do OTN é definida na recomendação G.709 [12] onde especifica que OTN é capaz de multiplexar, supervisionar e transportar canais ópticos que encapsulam sinais de clientes, como o SONET e o SDH, de forma transparente não interferindo no sinal transportado. Além disso, existem outras recomendações que complementam a arquitetura OTN, como por exemplo, a G.872 [13] que define o formato do quadro e o mapeamento dos sinais clientes. Em adição, OTN foi projetado para ser uma rede assíncrona que transporta sinais clientes síncronos e assíncronos, diferente das redes SDH e SONET, reduzindo a complexidade do sincronismo na rede. Outra funcionalidade introduzida no protocolo OTN é a inclusão do código de correção de erro (FEC, *Forward Error Correction*) o qual melhora expressivamente a relação sinal-ruído óptico (OSNR, *Optical Signal to Noise Ratio*) diminuindo a taxa de erro de bit, e como efeito, permitindo a diminuição no número de repetidores na rede de transporte óptico [4].

A hierarquia de camadas OTN é dividida entre os domínios elétrico e óptico. No elétrico estão a camada de transporte (OTU, *Optical Transport Unit*), a camada de dados (ODU, *Optical Data Unit*) e a camada de carga (OPU, *Optical Payload Unit*). As principais funções dessas camadas são de encapsular o sinal cliente, monitorar a taxa de erro de bit, gerar alarmes de falha de rede e de sinais e providenciar dados de monitoração de conexão. No domínio óptico encontra-se as camadas da seção de transmissão (OTS, *Optical Transmission Section*), a camada de multiplexação (OMS, *Optical Multiplexed Section*) e a camada de canal óptico (OCh, *Optical Channel*). Além disso, a hierarquia de multiplexação de sinais é executada em diferentes níveis da camada ODU: começando com ODU0 à taxa de aproximadamente 1.25 Gbps, ODU1 de 2.5 Gbps, ODU2 de 10 Gbps, ODU3 de 40.3 Gbps, ODU4 de 104.8 Gbps, e ODUCn de múltiplos inteiros de n de 104.8 Gbps. A multiplexação de diferentes taxas de ODU, chamados tributários de baixa ordem, é bastante flexível nos tributários de mais alta ordem de OTU, por exemplo, é possível multiplexar 8 ODU0 em OTU2 e 10 ODU2 em OTU4, de acordo com a recomendação G.709 [12].

A transição da terceira para a quarta geração das redes de transporte óptico foi um desperdício de investimento de infraestrutura para alguns casos, pois certas redes foram implantadas com fibras DSF que permitiu a transmissão de comprimentos de ondas com baixa atenuação e dispersão praticamente nula. Entretanto, com a introdução da técnica DWDM

observou-se que essas redes apresentavam alta degradação por efeitos não lineares, principalmente pelo efeito de mistura de quatro ondas (FWM, *Four Wave Mixing*), inviabilizando toda a infraestrutura de fibras DSF com dispersão nula instaladas em campo [14]. Para resolver esse problema, surgiu a fibra de dispersão deslocada não nula (NDSF, *Non-dispersion Shifted Fiber*) especificada pela norma G.655 [15]. Logo, toda a infraestrutura de DSF foi dedicada para o uso de sistemas TDM na região 1550 nm devido a sua ineficiência nas redes DWDM, enquanto a nova infraestrutura de NDSF era adequada para a nova geração de redes de transporte óptica. Houve, portanto, uma demanda da indústria da rede de transporte que a quinta geração reusesse a infraestrutura de fibra óptica já instalada em campo, de forma a reduzir o custo de instalação [6]. Contudo os novos sistemas da posterior evolução mantiveram a infraestrutura óptica não substituindo amplificadores e multiplexadores em campo, mas somente trocando os transponders com taxa de 2.5 Gbps para os novos transponders de 10 Gbps com espaçamento entre canais de 100 GHz [3].

A quinta geração dos sistemas de transporte óptico foi caracterizada pela otimização do espectro óptico que o sistema DWDM opera, destacando-se a invenção dos amplificadores Raman, que apresenta menor figura de ruído, melhorando portanto, a figura de mérito do sistema e a extensão da transmissão de canais nas bandas C e L [3].

Por volta do ano 2000, a comunidade científica retomou experimentos com sistemas ópticos coerentes, buscando projetar sistemas com formatos de modulação avançada como o chaveamento por deslocamento de fase em quadratura (QPSK, *Quadrature Phase-Shift Keying*) e a modulação de amplitude em quadratura (QAM, *Quadrature Amplitude Modulation*). Após a implementação dos primeiros receptores coerentes, observou-se que processadores de sinais digitais (DSP, *Digital Signal Processor*) poderiam contribuir com o processo de detecção e de recuperação do sinal no domínio elétrico, compensando a degradação e executando códigos de erros customizados [8]. Nessa geração, foi possível desenvolver transponders comerciais operando a uma taxa de 40 Gbps com modulação diferencial QPSK, e com espaçamento entre canais de 50 GHz e 100 GHz [3].

Por volta de 2010 ocorreu a evolução de DSP robustos, sendo possível então, o desenvolvimento de transponders comerciais operando em 100 Gb/s usando técnicas de modulação avançadas de multiplexação da polarização QPSK [16]. Além disso, observou-se a forte adesão do mercado em transceptores, ou plugáveis, que são módulos compactos externos projetados para transmitir sinal óptico em diferentes formatos de protocolos, seja o protocolo Ethernet ou SDH. Para evitar problemas incompatibilidade no formato e tamanho dos transceptores, os fabricantes fizeram acordos de múltiplas fontes que servem para padronizar todas as características

necessárias. Para exemplificar, o CFP (*C form-factor pluggable*) [17] foi inicialmente especificado para interfaces 100 Gbps para substituir o SFP+ (*small form-factor pluggable transceiver plus*) [18] que opera nas interfaces 10 Gbps.

Em 2012, ITU padronizou o espaçamento flexível nas redes WDM pela atualização da recomendação G.694.1 [9], permitindo a otimização da utilização do espectro óptico de modo que sinais de diferentes taxas possam ocupar diferentes larguras de banda no espectro, ou seja, o espaçamento entre canais é determinado pelo formato de modulação e pela taxa de transmissão dos canais. Essa recomendação define que o espaçamento flexível mínimo entre canais deve ser múltiplo de 12.5 GHz, não restringindo o espaçamento de 50 ou 100 GHz como especificado para os equipamentos ópticos da geração anterior.

A evolução dos transponders com taxas ainda maiores como 400 Gbps e 1.2 Tbps, mantendo o custo comercial atrativo, seria investir em modulações avançadas como entre 8QAM à 64QAM [16] e nas técnicas ópticas de super canal que consiste na multiplexação transparente de vários sinais portadores unitários usando a técnica óptica coerente de multiplexação ortogonal de divisão de frequência (CO-OFDM, *Coherent Optical Orthogonal Frequency-Division Multiplexing*) [19]. Desta forma, o espaçamento entre os canais dessas portadoras é flexível de formar a otimizar a ocupação do espectro óptico [4].

Pode se concluir que existem vários tipos de equipamentos DWDM, como amplificadores EDFA e Raman, transponders de diferentes taxas que podem ter transceptores que funcionam em modulações diferentes, ROADMs que configuram adição ou remoção de canais locais, e dentre outros equipamentos. Sendo assim, o gerenciamento e a manutenção da rede de transporte óptico é complexo devido a quantidade de possibilidades de configurações e ajustes nos equipamentos pois cada fabricante implementa produtos específicos com funcionalidades que agrupam um conjunto de funções dos elementos DWDM. Com o objetivo para especificar as funções chaves de gerenciamento remoto para o controle dos equipamentos da rede de transporte, ITU publicou a recomendação M.3348 [20] que especifica o sistema de gerenciamento de redes (NMS, *Network Management System*), que é responsável por monitorar os elementos da rede (NE, *Network Element*), armazenar os estados, exibir notificações e alarmes, além de gerenciar o desempenho de métricas e alarmes.

1.1. Problematização

Nas redes ópticas atuais cada fabricante de equipamentos DWDM oferece um NMS proprietário de código fechado que é capaz de somente gerenciar os seus equipamentos. Conseqüentemente, o planejamento e provisionamento de serviços em uma rede de transporte com

vários fabricantes é complexo e custoso por causa da quantidade de NMS que o usuário tem que configurar. A desagregação dos elementos da rede de transporte óptico surge como uma solução adequada para flexibilizar a configuração de toda a topologia da rede, desapropriando a tecnologia proprietária de comunicação de cada fabricante.

1.2. Objetivo Geral

Este trabalho tem como objetivo investigar e avaliar as funcionalidades e capacidades do modelo de desagregação da rede óptica OpenROADM [21].

1.2.1. Objetivos Específicos

- Revisar a literatura sobre os conceitos de desagregação da camada óptica e como são aplicados às redes ópticas definidas por software (SDN, *Software Defined Network*).
- Revisar a documentação e arquivos do modelo de equipamentos do OpenROADM.
- Verificar os atuais projetos de código aberto e quais os níveis de desagregação que oferecem.
- Configurar e demonstrar os principais módulos dos modelos YANGs do OpenROADM por meio da interface NETCONF dos simuladores ConfD na versão gratuita.
- Simular uma topologia de rede e verificar como o controlador SDN provisionaria um serviço entre dois pontos da rede.

1.3. Justificativa

O gerenciamento e controle de equipamentos DWDM de diferentes fabricantes mostrou-se complexo devido à falta de um sistema central de NMS. Dessa forma, a desagregação da rede de transporte óptica aplicada com conceitos de SDN vem se tornando uma alternativa para construir a topologia de redes, flexibilizar a configuração de diferentes equipamentos e separar a camada de controle da camada de dados. Para isso, é necessário que os equipamentos da rede de transporte sejam desagregados em modelos de dados padronizados sem perder suas funcionalidades ópticas. Portanto, avaliar se os modelos de equipamentos são compatíveis, flexíveis e desagregam as funções ópticas de diferentes tecnologias é de extrema importância para viabilizar a adesão de equipamentos desagregados.

1.4. Proposta e Organização do Trabalho

Este trabalho de conclusão de curso tem como objetivo investigar como o modelo de dados OpenROADM controla e desagrega os equipamentos DWDM mais relevantes da rede como

o transponder e ROADM. Portanto, os módulos do modelo mais essenciais serão apresentados no trabalho para demonstrar como as funcionalidades ópticas dos equipamentos DWDM são modeladas e controladas.

A organização do trabalho está dividida na seguinte ordem: o capítulo 2 explora os conceitos da rede definidas por software, o capítulo 3 expõe as principais arquiteturas de rede definidas por software propostas para a aplicação da rede de transporte ópticas, o capítulo 4 demonstra os principais módulos do modelo do equipamento YANG OpenROADM para os nós do tipo transponder e ROADM, o capítulo 5 contém os passos que o controlador SDN executa para provisionar um serviço na rede e por fim o capítulo 6 apresenta a conclusão e estudos propostos por este trabalho.

Capítulo 2

Redes Definidas por Software

Redes definidas por software (SDN, *Software Defined Network*) é uma arquitetura de rede que gerencia seus elementos de uma forma inteligentemente programável e dinâmica, permitindo reconfigurações de acordo com a demanda do usuário ou de aplicações da rede [22]. A arquitetura SDN foi inspirada inicialmente para ser desenvolvida em redes de pacotes de dados onde os principais elementos de rede são os roteadores [23]. Uma das definições mais convencionais do SDN consiste na separação do plano de dados do plano de controle. O plano de controle é responsável pela monitoração e configuração dos equipamentos da rede, aplicando as políticas de roteamento. Já a função do plano de dados é direcionar o tráfego, encaminhando os fluxos de dados de acordo com as regras estabelecidas pelo plano de controle [24].

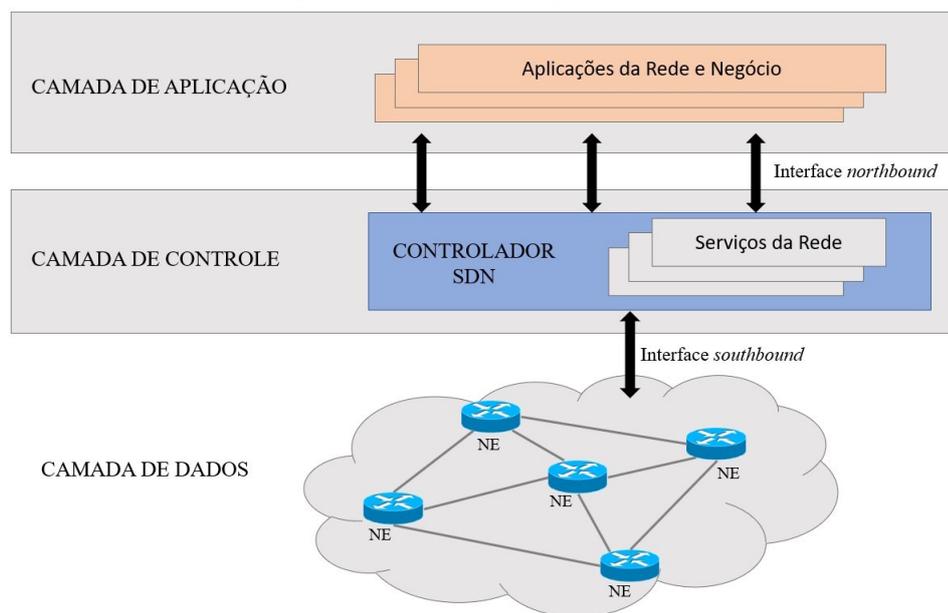
Por outro lado, a arquitetura de redes tradicional é caracterizada pela execução dos planos de dados e controle no mesmo equipamento de rede. Dessa maneira, a tabela de roteamento é construída no próprio dispositivo por programas de controle embarcados que determinam para qual interface encaminhar os pacotes que chegam, caracterizando um controle descentralizado [25]. Os elementos de rede dessa arquitetura têm seu código fonte fechado, não disponibilizado pelo fabricante, o que dificulta a evolução de novas ideias e aplicações da rede [24].

Em uma solução SDN, todos os equipamentos são inteligentemente programados e controlados usando aplicações de software, tornando assim, a topologia da rede centralizada em um único sistema de gerenciamento [26]. Dessa forma, os provedores de serviços de telecomunicações podem controlar qualquer entidade da rede não envolvendo as dependências específicas de cada fabricante, e conseqüentemente, contribuindo para uma queda significativa do custo [22]. Uma outra vantagem da solução SDN é que adota o padrão de código aberto dando a oportunidade de qualquer desenvolvedor contribuir para o desenvolvimento e manutenção das funcionalidades [23].

A arquitetura SDN centraliza o plano de controle por meio de um controlador SDN, conhecido também como Sistema Operacional da Rede (NOS, *Network Operating System*), que permite a visualização de todos os elementos da rede e centraliza sua programação. Além das camadas de controle e de dados, o modelo SDN também incorpora o plano de aplicação responsável por hospedar e executar os aplicativos da rede, definido políticas do negócio e de alto

nível [27]. Portanto, o controlador SDN ou NOS exerce um papel essencial na arquitetura SDN como controlar os serviços de rede, enviar comandos, receber notificações dos elementos de redes e fornecer dados para a camada de aplicação [23]. A Figura 1 ilustra as camadas da arquitetura SDN.

Figura 1 – Camadas da arquitetura SDN.



Fonte: Adaptada pelo autor, baseado em [24].

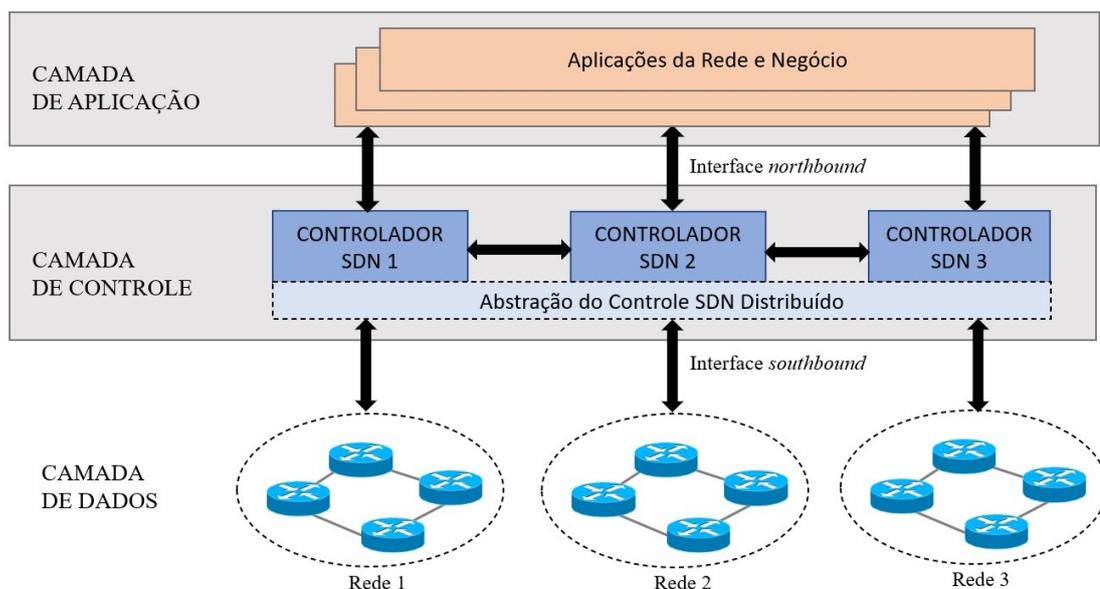
Como pode ser observado na Figura 2, o plano de controle comunica-se com os outros planos através das interfaces de comunicação sul (SBI, *SouthBound Interface*) e de comunicação norte (NBI, *NorthBound Interface*). A SBI é responsável por programar o plano de dados através de protocolos de comunicação enquanto a função da NBI é fornecer informações para os aplicativos da rede do plano de gerenciamento [23].

As aplicações da rede e de negócio são executadas na camada de aplicação mostrado na Figura 2 e se comunicam com o controlador SDN pela NBI. O controlador SDN ou NOS pode oferecer uma visão abstrata da rede para a camada de aplicação, de forma a reduzir a complexidade não essencial para a operação das aplicações de uma forma centralizada [22]. As principais funcionalidades implementadas são: 1) a configuração de rotas do fluxo de dados para determinar o menor custo, 2) o recálculo e reconfiguração das tabelas de roteamento dos elementos de rede em caso de falha dos caminhos, 3) o balanceamento de tráfego para evitar a sobrecarga e 4) a aplicação das políticas de rede como autenticação, bloqueio de pacotes e aplicação de níveis de acesso [23].

A arquitetura SDN facilita a evolução e a inovação das redes pois centraliza o controle dos elementos de redes em uma visão geral, permitindo implementações de serviços de rede

diretamente na camada de controle [24]. Por outro lado, escalabilidade, alta disponibilidade, tempo de resposta e segurança são fatores cruciais para cumprir o padrão de desempenho na arquitetura SDN exigidos pelos provedores de serviços de telecomunicação [26]. Nos aspectos de escalabilidade e desempenho, um único controlador pode não ser suficiente para responder todas as requisições de uma rede com grande número de elementos com um tempo de resposta aceitável. Para resolver esse problema, a arquitetura SDN permite uma abordagem distribuída de controladores indicado na Figura 2, onde cada controlador da camada de controle se intercomunicam para criar uma abstração de um controlador geral respondendo requisições das aplicações da rede e negócio [28]. No aspecto de alta disponibilidade, os controladores podem ser configurados como controlador mestre ou escravo. Se o controlador mestre para de funcionar por qualquer motivo, o controlador escravo assume o papel de mestre tomando todo o controle da arquitetura, logo, reduzindo o risco de indisponibilidade do sistema [23].

Figura 2 – Arquitetura SDN distribuída de controladores.



Fonte: Adaptada pelo autor, baseada em [23].

Historicamente o paradigma SDN foi implementado em algumas soluções para as redes de pacotes no passado, mas não houve aderência das tecnologias propostas no passado por diferentes motivos como a incompatibilidade de hardware e a falta de especificação efetiva [28].

A primeira arquitetura SDN de destaque desenvolvida nas redes de pacotes foi a Separação do Elemento de Controle e Envio (ForCES, *Forwarding and Control Element Separation*) que padronizava os comutadores simples de tráfego controlados por elementos de controle responsáveis por estabelecer rotas de tráfego usando o protocolo de redes *Border Gateway Protocol* (BGP) [24]. A falta de especificação efetiva e a exigência de novo hardware foram os motivos pelos quais a arquitetura ForCES não foi aderida [28].

Em 2007 a arquitetura de redes Ethane foi especificada de forma que implementou políticas de controle de acesso distribuído a partir de um controle de supervisão centralizado em ambientes corporativos. Nessa arquitetura, o supervisor é consultado pelos elementos de rede para decidir onde encaminhar o fluxo de dados a partir de regras globais, definidas pelos administradores de rede como criar níveis de acesso da informação e autenticação de usuários [24]. Dessa forma, a tabela de roteamento dos elementos de redes é construída por um controlador central. O principal motivo da não ascensão dos sistemas de redes Ethane, era incompatibilidade com o hardware dos elementos de redes já instalados em campo, portanto, a solução foi adotada somente para novas redes implantadas com essa tecnologia [23].

No ano de 2008 surgiram os orquestradores com o intuito de automatizar algumas funcionalidades na rede em diferentes elementos, utilizando recursos como: 1) Interface de Programação de Aplicativos (API, *Application Programming Interfaces*), 2) Interfaces de Linhas de Comandos (CLI, *Command Line Interface*) e 3) o Protocolo Simples de Gerência de Rede (SNMP, *Simple Network Management Protocol*). Tais orquestradores também podem ser automatizados para responder a eventos da rede, como por exemplo, provisionar novos serviços de rede para suprir a demanda de tráfego de horários de pico e recalcular uma nova rota de fluxo de dados para um serviço em falha [23].

A arquitetura de orquestração funciona na camada de aplicação apresentado na Figura 2, convertendo a instrução do comando desejado para a tecnologia proprietária do elemento de rede por meio de uma extensão de *software* fornecido pelo fabricante. No caso do SNMP, a função de controle é definida pela Base de Informação de Gerenciamento (MIB, *Management Information Base*). Tais MIBs podem ser públicas que são genéricas para qualquer elemento de rede ou privadas que são de uso específico definidos pelos fabricantes. Como não havia uma padronização efetiva das MIBs, problemas de interoperabilidade entre diferentes fabricantes inviabilizavam a adesão dos orquestradores nas redes [23].

Em 2008, o protocolo OpenFlow foi definido herdando os mesmos conceitos da arquitetura de redes Ethane com a adição de compatibilidade de hardware dos elementos de redes já fabricados, marcando efetivamente a origem do SDN [27].

A rede com roteadores OpenFlow é composta por um controlador que opera em um servidor separado dos roteadores e comutadores, o qual, tem uma visão de toda topologia da rede e é capaz de calcular quais são as melhores rotas para transportar os pacotes por diferentes terminações. Portanto, o controlador SDN configura o plano de dados nos respectivos roteadores, preenchendo as regras na tabela de roteamento por meio de um agente OpenFlow, responsável por

processar as requisições e configurar o plano de dados [25]. Deste modo, a solução OpenFlow define que os elementos de redes são programados remotamente por meio de um dispositivo externo com alta capacidade de processamento de aplicações de redes de código aberto [23].

A adesão do protocolo OpenFlow ocorreu por volta de 2011 quando os fabricantes começaram a implementar o protocolo nos seus produtos, permitindo aos pesquisadores inovarem e experimentarem novas ideias. Por causa disso, a Fundação Aberta de Redes (ONF, *Open Network Foundation*) foi criada para dar continuidade à evolução do protocolo OpenFlow e também para representar as soluções SDN globais [22]. Além disso, a ONF certifica elementos de rede que implementam o protocolo OpenFlow avaliando todos os critérios requeridos [23].

Atualmente, os fabricantes e os provedores de serviços de telecomunicações estão aderindo a controladores SDN, os quais, utilizam o protocolo de configuração de redes (NETCONF, *Network Configuration Protocol*) especificado na norma RFC 6241 [29] como SBI, pois permite o gerenciamento de configurações e estados nos elementos da rede [30]. Assim, o controlador SDN comunica-se com os elementos de redes que informam suas habilidades e capacidades de configuração representados por um modelo de dados usando a linguagem YANG (*Yet Another Next Generation Data Modeling Language*) definida na RFC 6020 [31]. A utilização de modelos de dados YANG popularizou-se devido a facilidade de entendimento da estrutura semântica agilizando o processo de desenvolvimento e integração. Para a representação do modelo YANG usando o protocolo NETCONF geralmente utiliza-se o formato XML (Extensible Markup Language) [23].

O controlador SDN descobre e aprende quais são as operações suportadas ao se conectar no elemento de rede pela primeira vez, e de certo modo, cada equipamento de fabricante diferente vai prover um modelo YANG distinto que será processado pelo controlador. Para manter a interoperabilidade entre os equipamentos, é necessário que o modelo YANG implementado por cada fabricante tenha um conjunto de características mínimas iguais e padronizadas [32].

O protocolo NETCONF é o sucessor do protocolo SNMP, operando por meio de procedimentos de chamadas remotas (RPC, *Remote Procedure Call*) ao elemento da rede, fornecendo os parâmetros de configuração e validando se a operação foi executada com sucesso ou falha [30]. Além disso, tal protocolo permite que o elemento de rede notifique o controlador por meio do envio de notificações, como também, mantém um histórico de translações de configurações executados no dispositivo, sendo possível retroceder para qualquer estado no passado. Para isso, cada funcionalidade é separada em estado de configuração e o estado

operacional, este representa o estado no qual o equipamento está atuando no momento enquanto aquele consiste no estado desejado e alvo [32].

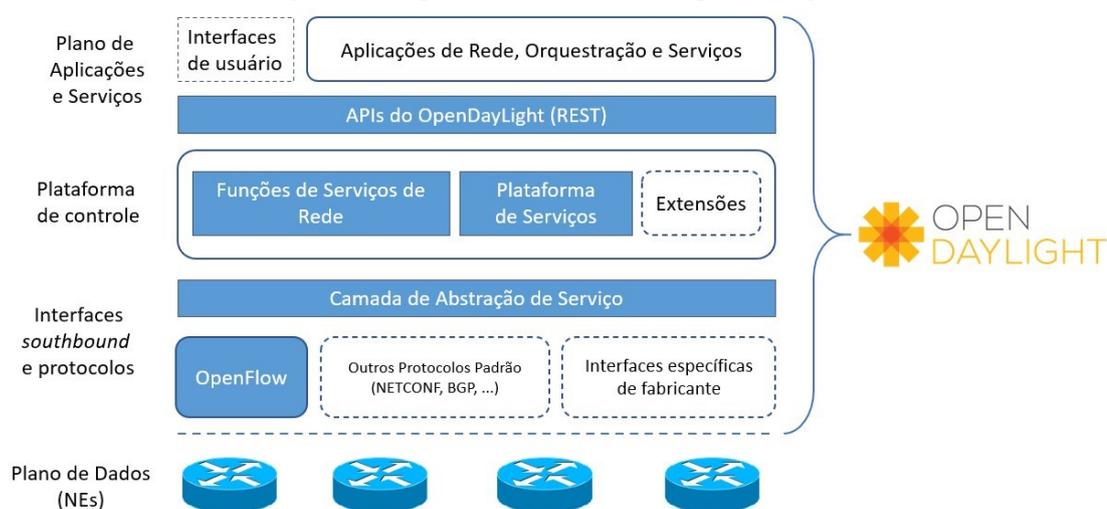
Existem vários tipos de controladores SDN disponíveis para diferentes aplicações. Neste trabalho será detalhado os controladores OpenDayLight e ONOS por causa de suas funcionalidades mais compatíveis para as redes de transporte ópticas de grandes provedores de serviços de telecomunicação.

2.1. OpenDayLight

O OpenDayLight é um controlador SDN de código aberto que suporta uma grande variedade de protocolos de comunicação como o OpenFlow e é escrito na linguagem de programação Java. Foi inicialmente desenvolvido pela Cisco e transferido para fundação Linux. Esse controlador atende os requisitos para testes em ambientes de pesquisa e até de grandes redes, além de oferecer uma interface web para visualizar e configurar os elementos de redes por meio dos serviços e das aplicações [23].

A Figura 3 representa a arquitetura de camadas do OpenDayLight. A camada de abstração de serviço faz a função de SBI, providenciando suporte dinâmico aos protocolos mais comuns como OpenFlow, NETCONF e BGP para a comunicação dos NEs. Além do mais, OpenDayLight no NBI responde requisições das aplicações de rede e serviços por meio do paradigma REST (*Representational State Transfer*) que é o acesso dos dados e das configurações por serviços da web. Por fim, a plataforma de controle é modular permitindo a adição e remoção de serviços de rede em forma de extensões sem interferir na operação do controlador e a arquitetura também gerencia modelos de dados YANG dos equipamentos conectados na rede [33].

Figura 3 – Arquitetura do controlador OpenDayLight.



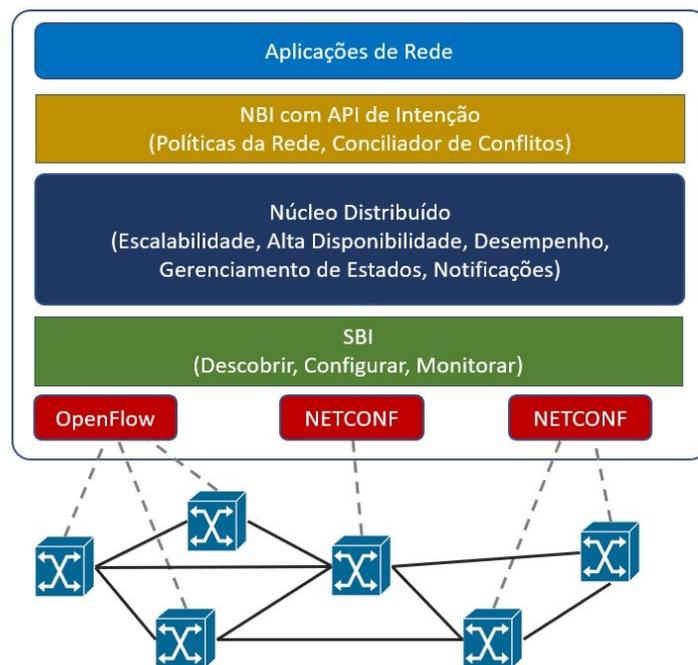
Fonte: Adaptada pelo autor, baseado em [33].

2.2. ONOS

Em 2014, o Sistema Operacional de Rede Aberta (ONOS, *Open Network Operating System*) foi o primeiro controlador de código aberto projetado especificamente para os provedores de serviços de telecomunicações com foco em escalabilidade e desempenho, funcionando com multiprocessamento de forma distribuída, atendendo os requisitos de baixo tempo de resposta, alto desempenho, alta disponibilidade e suportando redes grandes [23].

O ONOS também oferece na NBI uma abstração da topologia da rede provendo estados dos NEs e a utilização dos recursos em uma interface gráfica para os usuários. Na SBI, o ONOS suporta vários protocolos como NETCONF e OpenFlow, e também gerencia as capacidades dos elementos das redes através de modelos YANG. Tal controlador também oferece uma API para auto descoberta de dispositivos na rede, configuração de funcionalidades e consulta de notificações, de certa forma, facilitando o desenvolvimento das aplicações de rede [30].

Figura 4 – Arquitetura do controlador ONOS.



Fonte: Adaptada pelo autor, baseado em [34].

A camada de aplicação se comunica com o ONOS usando um paradigma de intenção de configuração, significando que a aplicação faz uma requisição para o NBI do ONOS com o pedido de configuração desejado em forma de uma política que será processada e traduzida para se tornar uma operação na mudança do ambiente da rede. Portanto, as aplicações da rede podem ser desenvolvidas usando a biblioteca de intenções fornecida pelo ONOS ou os desenvolvedores podem modificar a estrutura nativa de intenções do ONOS [30].

As arquiteturas convencionais e os controladores das Redes Definidas por Software foram apresentados neste capítulo, com o objetivo de apresentar uma revisão bibliográfica das tecnologias ForCES, Ethane, OpenFlow, OpenDayLight e ONOS. No próximo capítulo desse trabalho serão explorados os principais conceitos da aplicação do SDN para as redes abertas desagregadas de transporte ópticas.

Nesse capítulo foi apresentado os conceitos da arquitetura SDN, terminologias dos elementos da rede e das interfaces, arquiteturas das camadas e um breve histórico das soluções até o nascimento do protocolo OpenFlow. Além disso, foi mostrado o protocolo de comunicação NETCONF que utiliza modelos de dados escritos em linguagem YANG, bem como, os controladores SDN de código aberto ONOS e OpenDayLight.

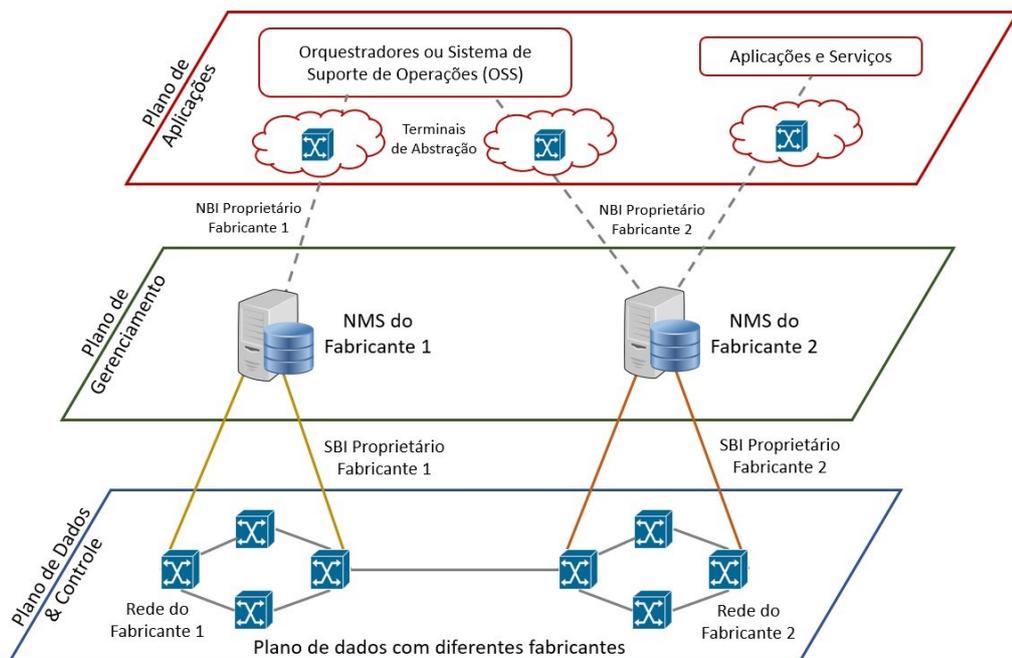
Capítulo 3

Redes de Transporte Desagregadas

Em redes de transporte ópticas convencionais, os equipamentos DWDM de cada fabricante são controlados e monitorados por um NMS proprietário exclusivo ilustrado na Figura 5, caracterizando uma rede de transporte totalmente agregada [30]. Portanto, os provedores de serviços de telecomunicações usam diferentes NMSs para controlar partes isoladas das redes, dificultando o provisionamento de serviços entre redes de diferentes fabricantes pois não existe uma visão de todos os domínios da rede.

A Figura 5 também mostra que alguns fabricantes oferecem NBI não padronizada, geralmente implementando MIBs do protocolo SNMP, para prover informações e aceitar comandos de configurações para aplicações como orquestradores proprietários e Sistemas de Suporte de Operações (OSS, *Operations Support System*) [35]. Portanto, o desenvolvimento de novas funcionalidades para melhorar o controle da rede e reduzir o custo operacional nesse tipo de arquitetura convencional é lento e segmentado, surgindo a necessidade de evoluir as redes de transporte com conceitos de SDN como a desagregação [36].

Figura 5 – Redes de transporte convencionais.



Fonte: Adaptada pelo autor, baseado em [30].

A desagregação da rede de transporte consiste em evoluir os NEs com protocolos abertos de comunicação e configuração de modo que possam ser genericamente configurados por um controlador. O processo de desagregação, portanto, envolve a escolha de um protocolo padrão de comunicação SBI para o plano de dados e NBI para o plano de aplicações [37]. Além disso, os equipamentos de diferentes fabricantes podem ser gradualmente desagregados em componentes de rede ou subsistemas que exercitam funções de controle específicos. Para isso, a evolução da desagregação da rede de transporte é caracterizada pelos níveis parcial e completa. A desagregação parcial compreende no controle das funções dos transponders nos terminais da rede, enquanto o nível de desagregação completa expande o controle em mais elementos mais específicos como amplificadores e ROADMs [35]. A desagregação total dos equipamentos, de fato, providencia flexibilidade para redes de transporte com diferentes fabricantes permitindo a utilização de qualquer produto da mesma classe classificados como equipamentos de caixa branca. De certa forma, tal tipos de equipamentos influenciam na competitividade entre os fabricantes promovendo a inovação e a redução do custo das redes de transporte [37].

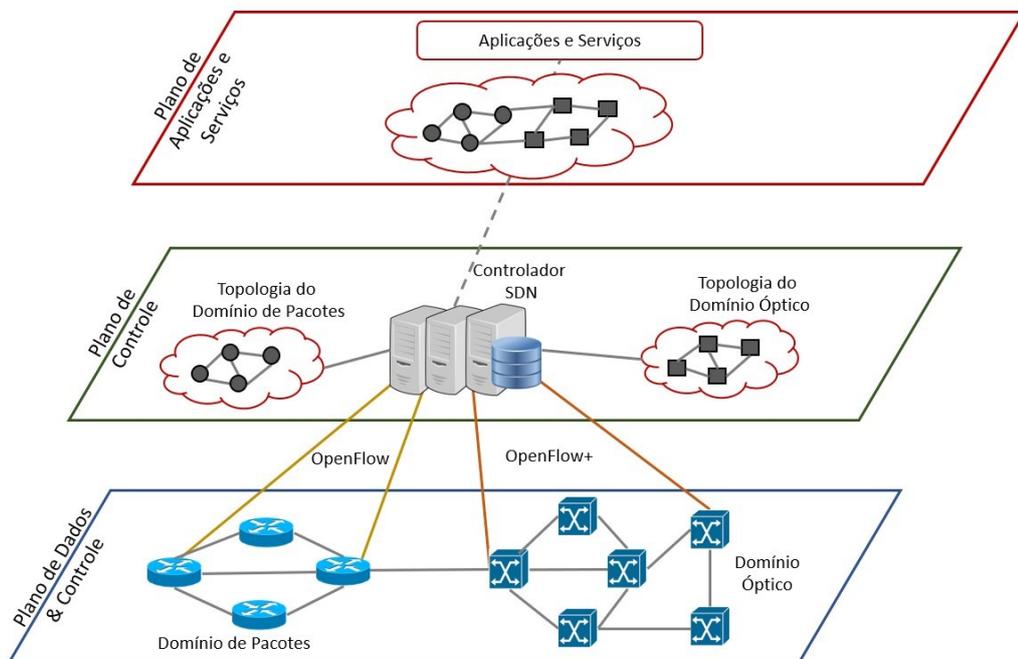
Historicamente, a desagregação surgiu como solução SDN para a evolução da rede de transporte definindo a arquitetura do SDN transporte (T-SDN, *Transport SDN*). O T-SDN é denominada quando o paradigma do SDN, as quais soluções estão implementadas principalmente no controle de redes de pacotes, é adaptada e aplicada para as redes de transporte. Atualmente existem diferentes tecnologias encontradas nas redes de transporte, começando na camada 0 do modelo OSI, correspondendo a camada física óptica pelos equipamentos DWDM e alguns equipamentos legados TDM. Logo mais, na camada 1 existe os protocolos de rede de transporte OTN, SDH e SONET que especificam níveis de multiplexação e hierarquias. Finalmente, as redes de transporte também incluem as camadas 2 e 3 do modelo OSI que envolvem os protocolos de rede de computadores como o Protocolo de Internet (IP, *Internet Protocol*), Ethernet, Comutação de Rótulos Multiprotocolo (MPLS, *Multiprotocol Label Switching*) [38]. O maior desafio da arquitetura T-SDN é como controlar e abstrair todas essas camadas e suas tecnologias específicas.

As soluções T-SDN também devem suportar várias camadas de abstração e domínios de rede diferentes, interoperar entre diferentes fabricantes de equipamentos de tecnologias diferentes, otimizar a utilização dos canais ópticos e providenciar um único sistema de monitoração e configuração de forma a reduzir custo operacional e manutenção da rede. Além disso, as redes de transporte têm funcionalidades de proteção e restauração de tráfego em caso de falhas nos enlaces ópticos para garantir a disponibilidade de serviços exigidos pelos provedores de serviços de telecomunicações. Para exemplificar, a proteção de tráfego nas redes de transporte

óptico é executada pelo hardware em caso se falhas forem detectadas nos enlaces monitorados, caracterizando a rápida recuperação do serviço. Enquanto a restauração de tráfego é executada por uma aplicação externa como PCE que calcula uma nova rota disponível na topologia e configura todos os NEs da rede, caracterizando uma lenta recuperação do serviço. Por isso, a arquitetura T-SDN é mais complexa que uma arquitetura SDN pois incorpora diversas funcionalidades exclusivas das redes de transporte [30].

Entre 2009 até 2014, as primeiras tentativas de uma solução T-SDN, conhecida como arquitetura T-SDN monolítica (SDON, *Software Defined Optical Networking*), consistiram em estender a arquitetura do protocolo OpenFlow em um único controlador SDN para somente um domínio de rede de transporte óptico [39]. O princípio de comunicação é o mesmo do protocolo OpenFlow já mencionado, um agente OpenFlow é executado nos NEs para responder as requisições do controlador SDN. Para atingir esse objetivo, o protocolo OpenFlow evoluiu para a versão 1.4 nomeado OpenFlow estendido (OF+, *Extended OpenFlow*) com funcionalidades de controle de portas ópticas como configuração do comprimento de onda, medição de potência de entrada e saída [40]. Apesar que os experimentos com soluções de OF+ como mostrado na Figura 6 no controle de funcionalidades avançadas como espaçamento flexível entre canais, gerenciamento de comprimentos de onda e de multiplexações ODU. A vantagem desse tipo de arquitetura é a concretização de um único protocolo de comunicação na SBI.

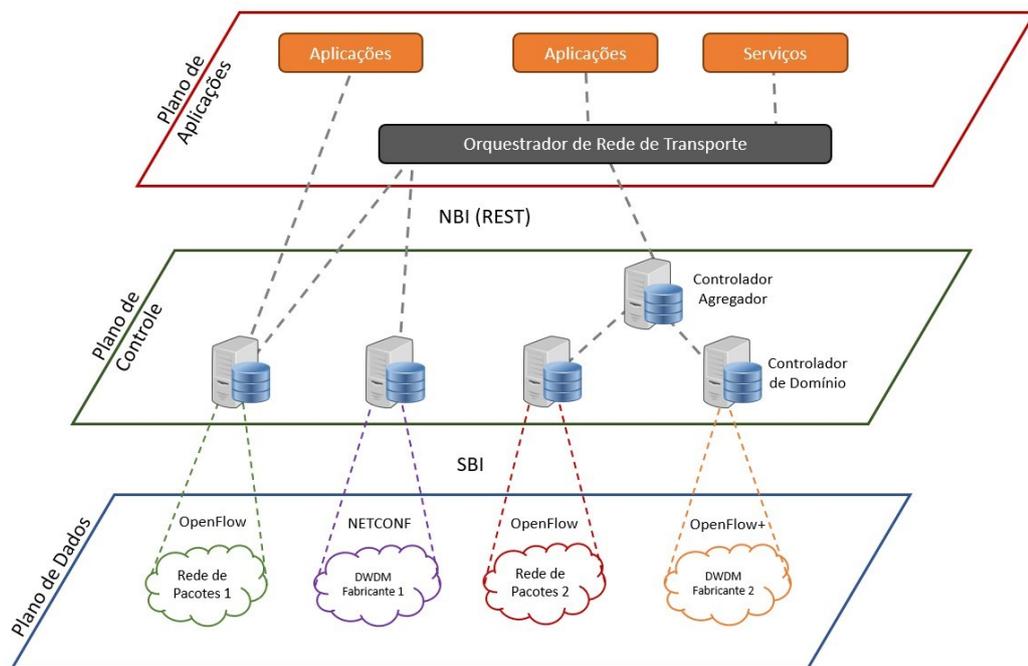
Figura 6 – Arquitetura SDN para redes de transporte usando OpenFlow+ e OpenFlow.



Fonte: Elaborada pelo autor, baseado em [30].

A posterior arquitetura proposta foi a arquitetura hierárquica de T-SDN (HT-SDN, *Hierarchical SDN*) que consiste na distribuição de um controlador SDN para cada domínio da rede de transporte, supervisionados por um controlador agregador ou orquestrador [41], representado na Figura 7. Cada controlador SDN de domínio é responsável por monitorar e configurar apenas os equipamentos de seu domínio, abstraindo um certo nível de complexidade para o controlador SDN geral, que por sua vez, é responsável por criar uma visão da topologia geral da rede e calcular as rotas de menor caminho entre os diferentes domínios para provisionar serviços [30].

Figura 7 – Arquitetura HT-SDN para rede de transportes de múltiplos domínios.



Fonte: Elaborada pelo autor, baseado em [30].

A arquitetura HT-SDN foi mais aderida pois providencia melhor escalabilidade e integração entre domínios de diferentes fabricantes. Atualmente, observa-se que os fabricantes implementaram seus próprios orquestradores para lidar com a camada de aplicação como a empresa Cisco com o Orquestrador de Serviços da Rede (NSO, *Network Service Orchestration*) e a Ciena com o Planeta Azul (*Blue Planet*). Os orquestradores de fabricantes não são reconhecidos pelos conceitos do SDN que exigem aplicações de código aberto [42]. Os controladores ONOS e OpenDayLight podem servir como orquestradores de código aberto, porém ainda não existe uma customização padronizada para atender os provedores de serviços de telecomunicação [30].

Para a adesão da arquitetura HT-SDN, surgiu o processo de desagregação da rede de transporte. Para a escolha do protocolo de comunicação padrão da SBI do controlador, o protocolo NETCONF surgiu como dominante devido à sua característica de adaptação à novas funcionalidades e no controle de redes com diferentes fabricantes [32]. Em contraste, o protocolo

OpenFlow+ não surgiu como opção para as redes de transporte pois uma nova funcionalidade exige um longo processo de desenvolvimento iniciando pela atualização da norma e posteriormente pela implementação e atualização dos equipamentos. Por causa da escolha do protocolo NETCONF como padrão preferencial de comunicação da SBI, surgiu a necessidade de definir modelos YANG padronizados para abstrair as funcionalidades dos equipamentos DWDM de diferentes fabricantes. Para isso, alguns modelos de dados surgiram como os padrões OpenROADM e OpenConfig liberados pelos grandes provedores de serviços de telecomunicações [43].

O OpenROADM [21] é um projeto iniciado pela empresa AT&T focado na desagregação dos equipamentos ROADMs para externalizar as funcionalidades de equipamentos DWDM através da tecnologia NETCONF/YANG. Os modelos OpenROADM podem desagregar diversos equipamentos DWDM como transceptores, transponders, ROADMs e amplificadores ópticos. Em 2019, foi publicada a versão 3.01 com várias novas funcionalidades como o suporte de configuração avançada de transponders e melhoria nas métricas de potência óptica. O projeto também inclui modelos de abstração da rede, de serviços e de equipamentos para facilitar o processo de desagregação. A maior vantagem de tal projeto é que modela detalhadamente algumas restrições e comportamentos da camada óptica, necessário para a otimização de recurso em uma rede de transporte [37].

O projeto OpenConfig [44] foi iniciado pela Google com objetivo de implementar modelos YANG transparentes a dependências de fabricantes. A modelagem dos elementos ópticos da rede de transporte é dividida nas categorias de terminação óptica, monitoração de canais, proteção de serviços, roteamento de comprimentos de onda e amplificação óptica. Qualquer categoria permite a configuração e a monitoramentos dos contadores de desempenho. Pode se concluir que o nível de desagregação dos elementos nos modelos OpenConfig é parcial enquanto os modelos OpenROADM contemplam um nível mais completo [43].

A ONF conduziu a padronização da NBI no processo de desagregação pelo projeto de transporte API (TAPI, *Transport API*) [45]. O princípio de funcionamento do TAPI implementa funções para abstrair e controlar a rede como prover o consumo do recurso dos enlaces da topologia, provisionar de serviços, calcular rotas entre os NEs e fornecer notificações da rede como falhas [46]. O projeto TAPI oferece ferramentas automatizadas para acelerar o processo de desenvolvimento e documentação, sendo sua primeira versão publicada em 2016, e posteriormente evoluída em 2018 com novas funcionalidades avançadas de monitoração, métricas de desempenho,

proteção e restauração de serviços. Por fim, o TAPI também incorpora modelos YANG padronizados para a configuração de equipamentos [45].

A partir das definições de arquitetura, interfaces NBI e SBI, padronização dos protocolos de comunicação, surgiram alguns projetos aplicando o processo de desagregação em redes de transporte já instaladas. Em 2018, o projeto Metro-Haul [35] utilizou controladores ONOS distribuídos para cada estação de metrô operando com modelos OpenConfig para os transponders e OpenROADM para plugáveis e ROADMs. As principais contribuições desse projeto foi demonstrar como a linguagem de dados YANG é capaz de abstrair detalhes não essenciais para a operação da camada de dados, simplificando de certa forma, a implementação e manutenção das aplicações de rede. Para isso, o projeto utiliza uma ferramenta automática do ONOS denominada como ONOS Yang Tools. Por fim, um dos objetivos alcançados desse projeto foi mostrar o controle de uma rede de transporte óptica desagregada de alta capacidade através de programas de código aberto.

Nesse trabalho, o projeto Redes de Transporte Aberta e Desagregada (ODTN, *Open and Disaggregated Transport Network*) será detalhado na próxima subseção.

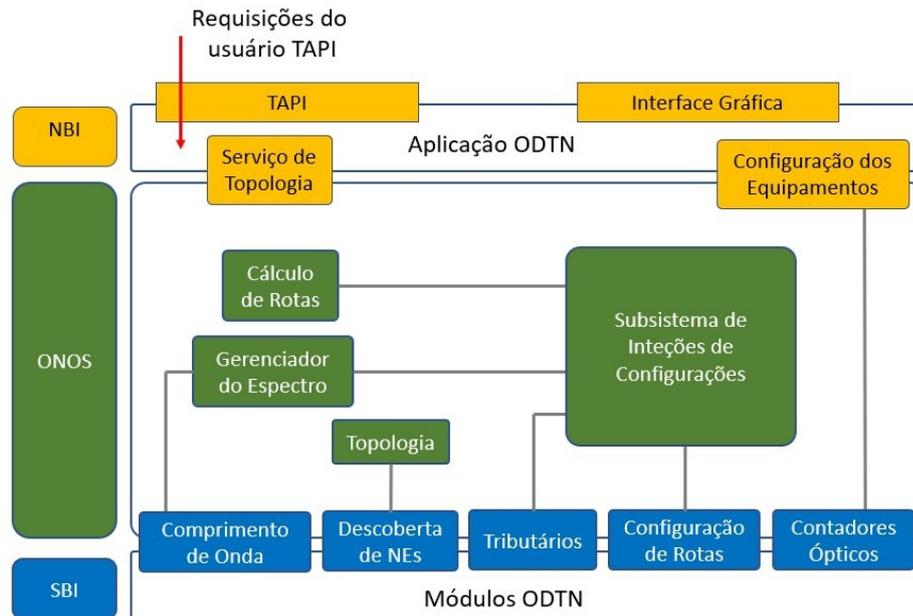
3.1. Rede de Transporte Aberta e Desagregada (ODTN)

Em 2018, a ONF criou o projeto ODTN [47] para servir como uma solução de redes de transporte desagregadas usando modelos de dados e aplicações não proprietárias. A Figura 8 representa a arquitetura em diagrama de blocos do projeto ODTN que basicamente estende o controlador ONOS nas interfaces de comunicação. O usuário pode fazer requisições e consulta de dados na NBI pelo TAPI versão 2 ou pela interface gráfica do controlador, enquanto na SBI, o projeto ODTN utiliza o protocolo NETCONF operando com modelos YANG padronizados, como OpenROADM, OpenConfig e TAPI, e modelos proprietários fornecidos pelos fabricantes. Dessa maneira, a arquitetura se torna mais flexível para diversas demandas do mercado, viabilizando sua introdução nas redes de transporte óptico já implantadas [48].

A Figura 8 também apresenta algumas funcionalidades nativas do controlador ONOS que são o cálculo de rotas, o gerenciador do espectro, topologia e o subsistema de intenções de configurações. Quando uma requisição de provisionamento de um serviço é recebida na NBI, o controlador aciona o subsistema de intenções para calcular a rota de menor custo da rede de transporte verificando quais são os dispositivos com recurso livre. Logo após, o controlador gera duas intenções de configurações: uma para a configuração da camada óptica alocando um canal óptico em todos os equipamentos da rota e outra para o provisionamento de conexões entre as

portas clientes e as portas de transporte nos transponders, incluindo a alocação de tributários de baixa ordem multiplexados em tributários de alta ordem. Além disso, as medidas ópticas são externalizadas na NBI do controlador, indicado na Figura 8 como o bloco de configuração de equipamentos.

Figura 8 – Diagrama de blocos resumido do projeto OTDN.



Fonte: Elaborada pelo autor, baseado em [36].

O projeto OTDN está na etapa de experimentação e provas de conceito para integrar no mercado. Em 2019, uma demonstração do projeto em fase de desagregação parcial foi apresentada pelos provedores de serviços de telecomunicação Telefonica e NTT Communications em parceria com a ONF [49]. Além disso, foi mostrado a descoberta de topologia e o provisionamento de serviços utilizando dois transponders desagregados de fabricantes diferentes sendo comandados por requisições do usuário pela a NBI implementando o TAPI. Por fim, foi possível evidenciar a configuração dinâmica dos transponders após suas capacidades identificadas pelo o ONOS durante a descoberta da topologia utilizando modelos OpenConfig. A primeira fase do projeto, portanto, priorizou o desenvolvimento das interfaces NBI e SBI estendidas pelo o ONOS.

A segunda fase do projeto OTDN consiste na definição de modelos para o controle desagregado de equipamentos ROADMs e no suporte de funcionalidades avançadas de configuração dos transponders como o formato de modulação e parâmetros de compensação. Algumas demonstrações no controle desagregado de ROADMs foram já apresentadas no passado de forma que contribuíram para a concretização da segunda fase. Por exemplo, em 2019 foi demonstrado o provisionamento de canais 100G por meio do controlador ONOS aprimorado para

uma rede de transporte com equipamentos totalmente e parcialmente desagregados de diferentes fabricantes [50]. O experimento adaptou os modelos de dados definidos no projeto OpenROADM de forma que as restrições físicas da camada óptica foram consideradas como a degradação e interferência, simulando de fato, uma rede de transporte óptica.

A terceira e última fase do projeto OTDN é desagregar totalmente as funções do ROADM, porém essa etapa ainda não foi iniciada por causa da incerteza no nível de detalhamento das funções de desagregação.

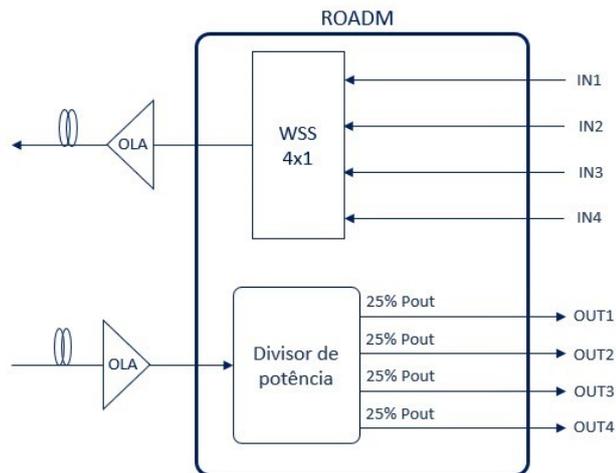
3.2. ROADM

Essa subseção tem como objetivo esclarecer brevemente alguns conceitos sobre os equipamentos ROADMs utilizando o módulo WSS. Como já foi mencionado, a função do ROADM é prover um mecanismo de roteamento de comprimentos de ondas, ou canais, no domínio óptico de forma reprogramável. Portanto, o sistema de controle NMS ou controlador SDN pode configurar tal equipamento remotamente por meio das interfaces de comunicação disponíveis. Desse modo, os provedores de serviços de telecomunicações têm uma flexibilidade maior para executar operações remotas em estações de telecomunicações de difícil acesso.

Os equipamentos ROADMs são projetados com um módulo WSS de um certo número de graus ou direções. O WSS é um elemento DWDM com uma porta de entrada e N portas de saída, onde N define o grau do módulo. Tal módulo é capaz de rotear qualquer comprimento de onda da porta de entrada para uma de suas portas de saída no domínio óptico ou vice-versa [51]. Por exemplo, o dispositivo WSS 1x4 indica que sua construção tem uma porta de entrada e quatro portas de saída, podendo rotear qualquer canal inserido em uma das portas de saída.

A Figura 9 apresenta um diagrama de blocos de um equipamento ROADM composto com um dispositivo WSS 4x1 e um componente de divisor de potência. O sinal óptico da linha de transmissão é recebido pelo componente de divisor de potência que direciona um quarto da potência do sinal para cada saída, OUT1, OUT2, OUT3 e OUT4. As entradas IN1, IN2, IN3 e IN4 são destinadas para receberem sinais de outros ROADMs ou para o banco de adição e remoção de canais, que será explicado posteriormente. Em outros termos, o módulo WSS une todos os comprimentos de onda recebidos das entradas e transmite na malha de transmissão.

Figura 9 – Diagrama de blocos de um equipamento ROADM.



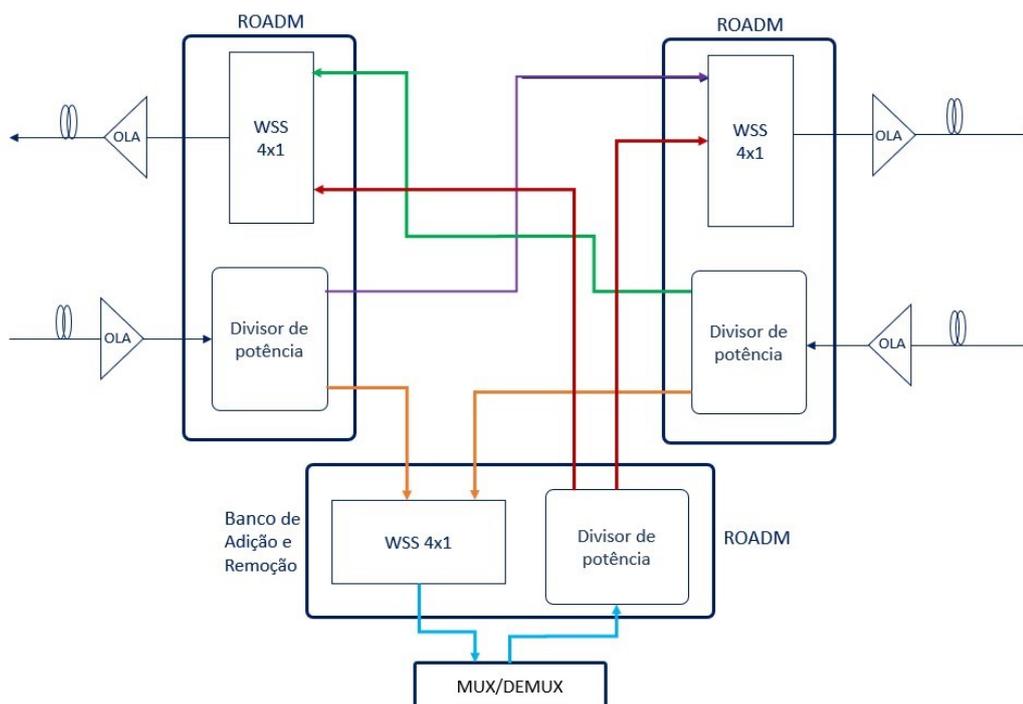
Fonte: Elaborada pelo autor, baseado em [51].

O módulo WSS tem uma perda de inserção do sinal óptico tipicamente grande, em alguns casos de 16 dB, e por isso utiliza-se amplificadores ópticos na entrada e saída de cada direção para compensar tais perdas [51]. O banco de adição e remoção de canais tem a função de receber os sinais das placas ROADMs que fazem as direções para remover os canais da malha de transmissão, ao mesmo tempo que adiciona outros canais. Esse bloco consiste em uma placa ROADM e um MUX/DEMUX pelo menos para receber os sinais de outros elementos DWDM, demonstrado na Figura 10.

A Figura 10 ainda ilustra o diagrama de blocos de nó ROADM de 2 graus ou direções com o banco de adição e remoção de canais. Cada placa ROADM pode suportar até 3 direções com um módulo WSS 4x1 contando com a última porta conectada no banco de adição e remoção de canais.

Utilizando-se essa arquitetura de direções com equipamentos ROADMs, é possível montar topologias de redes que atendam melhor as necessidades dos provedores de telecomunicações no quesito de proteger ou restaurar sinais em casos de rompimentos de fibras.

Figura 10 – Nó ROADM de 2 graus ou direções com banco de adição e remoção.



Fonte: Elaborada pelo autor, baseado em [51].

Os ROADMs de última geração podem ser classificados em duas categorias: C/D (*colorless/directionless*) e C/D/C (*colorless/directionless/contentionless*). A propriedade *colorless* indica que o ROADM é capaz de se processar qualquer comprimento de onda recebido ou transmitido. Já a propriedade *directionless* permite que o ROADM pode direcionar tráfego para qualquer outro nó da rede independente da direção. Em outros termos, todas as direções têm as mesmas propriedades e limitações. Por último, a propriedade *contentionless* informa que o equipamento ROADM é livre de bloqueio de canais originados de duas direções diferentes [51].

Nesse capítulo foi apresentado como os conceitos SDN foram aplicados nas redes ópticas de transporte originando as arquiteturas monolítica, utilizando o protocolo OpenFlow estendido, e hierárquica distribuída, aderindo o protocolo NETCONF. Com isso, foi apontado os modelos YANG padrões OpenConfig e OpenROADM. Depois, o objetivo e as fases do projeto ODTN foram expostos. O capítulo foi finalizado com o detalhamento do equipamento ROADM e seus possíveis configurações em uma rede de transporte.

Capítulo 4

Modelo OpenROADM

O principal objetivo dessa seção é detalhar como o modelo OpenROADM desagrega os equipamentos DWDM em entidades com atributos específicos. Para isso, o modelo do equipamento foi simulado em máquinas virtuais e dados fictícios foram inseridos para criar as entidades do modelo. O foco desse trabalho está no detalhamento das entidades que exercem a função dos equipamentos ROADMs.

4.1. Preparação dos Simuladores

O software ConfD Basic [52] foi usado para simular os elementos de rede operando como um servidor NETCONF para os modelos de dados OpenROADM. O ConfD é um sistema de gerenciamento embarcado projetado para sistemas Linux e sua principal vantagem é que providencia várias interfaces de comunicação como NETCONF, RESTCONF, SNMP, CLI e Web. Tais interfaces de comunicação são automaticamente geradas pelos modelos YANG, facilitando o desenvolvimento de funcionalidades dos equipamentos. Neste experimento, foram usados apenas as interfaces NETCONF e CLI.

O ConfD foi instalado em máquinas virtuais com pouco recurso computacional operando com a distribuição do Linux Ubuntu 18 [53] administrados pelo gerenciador de máquinas virtuais VirtualBox [54]. Cada elemento de rede óptico simulado é representado por uma única máquina virtual operando o ConfD com um banco de dados distinto. Após isso, o modelo OpenROADM [55] foi compilado no ConfD, permitindo a introdução de dados de configuração pelas interfaces de comunicação, de certa forma, simulando um equipamento NETCONF/YANG.

A recomendação do OpenROADM é composta pelos modelos de equipamento, rede e serviço que providenciam um nível aprofundado de desagregação e de abstração das diferentes camadas. A camada mais baixa, representada pelo modelo de equipamento, agrega os dispositivos DWDM em quatro tipos de nós de desagregação: ROADM, transponder, amplificadores de linha e transceptor. As informações de somente de leitura para o controlador SDN como dados de inventário das placas de um nó OpenROADM, bem como alguns estados e dados de configuração, são providenciadas pelos fabricantes na camada de equipamento. Por outro lado, as camadas de rede e serviço são primordialmente usadas pelo controlador SDN e pelas aplicações para oferecer

maneiras de construir a topologia da rede e provisionar circuitos. Tais camadas são independentes de tecnologia proprietária e de informação específica dos fabricantes, contribuindo para o ambiente de código aberto conceituado pela tecnologia SDN.

O modelo YANG do equipamento é representado pelo arquivo *org-openroadm-device.yang* do repositório OpenROADM [55] contendo todas as características necessárias para modelar um nó óptico desagregado. Tal arquivo tem diversas referências para outros modelos YANG definidos na mesma pasta e também na pasta *Common* do repositório. Para importar o modelo do equipamento no ConfD, é necessário primeiramente compilar todos os arquivos essenciais para a operação do nó ROADM. Para isso, o ConfD gera arquivos binários no formato FXS com o comando *confdc* para o processo de compilação. O Quadro 1 informa todos os arquivos YANG utilizados na simulação, cada um com seu arquivo FXS correspondente.

Quadro 1 – Lista dos arquivos YANG utilizados na simulação.

```
# ls | grep yang
org-openroadm-common-alarm-pm-types.yang
org-openroadm-common-amplifier-types.yang
org-openroadm-common-equipment-types.yang
org-openroadm-common-link-types.yang
org-openroadm-common-node-types.yang
org-openroadm-common-optical-channel-types.yang
org-openroadm-common-state-types.yang
org-openroadm-common-types.yang
org-openroadm-device-types.yang
org-openroadm-device.yang
org-openroadm-equipment-states-types.yang
org-openroadm-ethernet-interfaces.yang
org-openroadm-interfaces.yang
org-openroadm-maintenance-loopback.yang
org-openroadm-maintenance-testsignal.yang
org-openroadm-media-channel-interfaces.yang
org-openroadm-network-media-channel-interfaces.yang
org-openroadm-optical-channel-interfaces.yang
org-openroadm-optical-operational-interfaces.yang
org-openroadm-otn-common-types.yang
org-openroadm-physical-types.yang
org-openroadm-pm-types.yang
org-openroadm-pm.yang
org-openroadm-port-types.yang
org-openroadm-resource-types.yang
org-openroadm-resource.yang
org-openroadm-swdl.yang
org-openroadm-switching-pool-types.yang
org-openroadm-user-mgmt.yang
```

Fonte: Elaborada pelo autor.

O próximo passo é iniciar o ConfD com os arquivos FXS gerados. Para tal finalidade, é preciso executar o ConfD informando o caminho da pasta dos modelos OpenROADM compilados, mostrado no Quadro 2. Se não houver nenhum erro na inicialização por falta de algum

modelo não compilado, o ConfD inicia-se com sucesso e cria seu banco de dados denominado CDB. Qualquer dado de configuração ou operacional é armazenado nesse banco de dados. Por fim, o comando `confd_cli` acessa a interface CLI do ConfD permitindo ao usuário a manipulação de dados do simulador.

Quadro 2 – Iniciação do ConfD e da sua interface CLI.

```
# confd -v --addloadpath $CONFD_DIR/etc/confd --addloadpath
~/OpenROADM_MSA_Public/model/Custom/
# confd_cli
```

Fonte: Elaborada pelo autor.

4.2. Modelo de Informações Gerais: Módulo *info*

Os primeiros dados introduzidos no ConfD pertencem ao módulo *info*, responsável por providenciar informações gerais do nó ROADM. O Quadro 3 exibe parte do modelo do módulo *info* com alguns atributos (*leaf*) relevantes. Para exemplificar, o identificador único do nó é dado pelo atributo *node-id* do tipo *node-id-type*, definido no arquivo *org-openroadm-common-node-types.yang* e consiste em uma cadeia de 7 a 63 caracteres começando por uma letra. Os modelos OpenROADM oferecem um campo de descrição (*description*) para ajudar ao leitor a entender qual a função de cada atributo. De certa forma, a representação de dados na linguagem YANG facilita a compreensão humana, e conseqüentemente, contribuindo para a sua aceitação no mercado.

Quadro 3 – Parte do modelo YANG módulo *info*.

```
container info {
  leaf node-id {
    type org-openroadm-common-node-types:node-id-type;
    default "openroadm";
    description
      "Globally unique identifier for a device.";
  }
  leaf node-type {
    type org-openroadm-device-types:node-types;
    config true;
    mandatory true;
    description
      "Identifier for node-type e.g Roadm, xponder.
      Once the node-type is configured, it should not be modified.";
  }
  leaf vendor {
    type string;
    //config false;
    mandatory true;
    description
      "Vendor of the equipment";
  }
  leaf model {
    type string;
    //config false;
  }
}
```

```

    mandatory true;
    description
      "Physical resource model information.";
  }
  leaf serial-id {
    type string;
    //config false;
    mandatory true;
    description
      "Product Code for this physical resource";
  }
  [...]
}

```

Fonte: Elaborada pelo autor, baseado em [55].

O ConfD classifica os dados em dados de configurações que podem ser alterados pelo usuário e em dados operacionais que são fornecidos pelo equipamento. Por isso, o modelo OpenROADM indica quais são os atributos que são dados operacionais pela propriedade *config false*, fazendo o ConfD bloquear a modificação desses atributos pelo usuário. Os atributos *vendor*, *model*, *serial-id* do Quadro 3 tem tal propriedade, logo, exigindo que o fabricante implemente uma aplicação externa para prover esses dados para o ConfD. Para os objetivos dessa simulação, todos os atributos foram convertidos para dados de configuração, e assim, permitindo a configuração de qualquer atributo pelo usuário usando as interfaces de comunicação como o CLI.

O modelo YANG pode ser apresentado em forma de árvore como mostrado no Quadro 4 para o módulo *info*. Nesse tipo de visualização, todo atributo é classificado como dado de configuração permitindo a escrita *rw* (*read-write*) ou dado operacional para somente leitura *ro* (*read-only*). Analisando o Quadro 4, as informações de fabricante (*vendor*), modelo (*model*), número de serial (*serial-id*), número máximo de graus (*max-degrees*) e número máximo de SRGs (*max-srgs*) são providenciadas pelo próprio equipamento enquanto os outros atributos como o tipo do nó (*node-type*) e as coordenadas geográficas (*geoLocation*) são de responsabilidade do usuário ou controlador SDN. Esse tipo de representação do modelo YANG também informa qual o tipo de dados para cada atributo. Por exemplo, o campo do fabricante (*vendor*) é do tipo de cadeia de caracteres (*string*) e o campo do número do nó ROADM (*node-number*) recebe um número inteiro de até 32 caracteres (*uint32*). Nesse trabalho, a representação em árvore YANG será preferencialmente usada para apresentar os modelos utilizados.

Quadro 4 – Árvore YANG reduzida de atributos do módulo *info*.

```

+--rw info
|   +--rw node-id?                org-openroadm-common-node-types:node-id-type
|   +--rw node-number?           uint32
|   +--rw node-type              org-openroadm-device-types:node-types
|   +--ro vendor                 string
|   +--ro model                 string

```

	+-ro serial-id		string
	+-rw geoLocation		
		+-rw latitude?	decimal64
		+-rw longitude?	decimal64
	+-ro max-degrees?		uint16
	+-ro max-srgs?		uint16

Fonte: Elaborada pelo autor, baseado em [55].

O módulo *info* foi modelado para cada simulador desse experimento no ConfD por meio da interface CLI. Para consultar essas informações, o controlador SDN envia mensagens RPC no formato XML para o ConfD, este servindo função de um servidor NETCONF. Logo após o recebimento da mensagem, o ConfD a processa e retorna a mensagem com o resultado solicitado para o controlador SDN, este servindo função de um cliente NETCONF. Para exemplificar, o Quadro 5 representa uma mensagem RPC no formato XML para executar uma operação *get*, solicitando ao ConfD para retornar todos os dados operacionais e de configuração do módulo *info* do modelo do equipamento.

Quadro 5 – Mensagem RPC do NETCONF cliente para consulta do módulo *info*.

```
<get>
  <filter type="subtree">
    <org-openroadm-device xmlns="http://org/openroadm/device">
      <info/>
    </org-openroadm-device>
  </filter>
</get>
```

Fonte: Elaborada pelo autor.

Para enviar a mensagem RPC do Quadro 5, é necessário executar um cliente NETCONF que pode ser instalado na mesma máquina virtual do ConfD. Para isso, existe uma ferramenta chamada *netconf-console* de código aberto para exercer tal função. Por fim, a mensagem RPC de resposta enviada pelo ConfD é mostrada no Quadro 6. O corpo da mensagem é composto por diversas informações, como por exemplo, esse nó é do tipo ROADM (*node-type*) que suporta no máximo 3 direções (*max-degrees*) e está localizado no Rio de Janeiro, Brasil (*geoLocation*) identificado como RIO-ROADM-1 (*node-id*).

Quadro 6 – Resposta RPC do ConfD com as todas as informações do módulo *info*.

```
# netconf-console --rpc=get_device_info.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <org-openroadm-device xmlns="http://org/openroadm/device">
      <info>
        <node-id>RIO-ROADM-1</node-id>
        <node-number>1</node-number>
        <node-type>rdm</node-type>
```

```

<cli>RIO-ROADM-1</cli>
<vendor>UFU</vendor>
<model>ROADM_3DIR</model>
<serial-id>0001</serial-id>
<geoLocation>
  <latitude>-22.9935447</latitude>
  <longitude>-43.4374205</longitude>
</geoLocation>
<max-degrees>3</max-degrees>
<max-srgs>1</max-srgs>
</info>
</org-openroadm-device>
</data>
</rpc-reply>

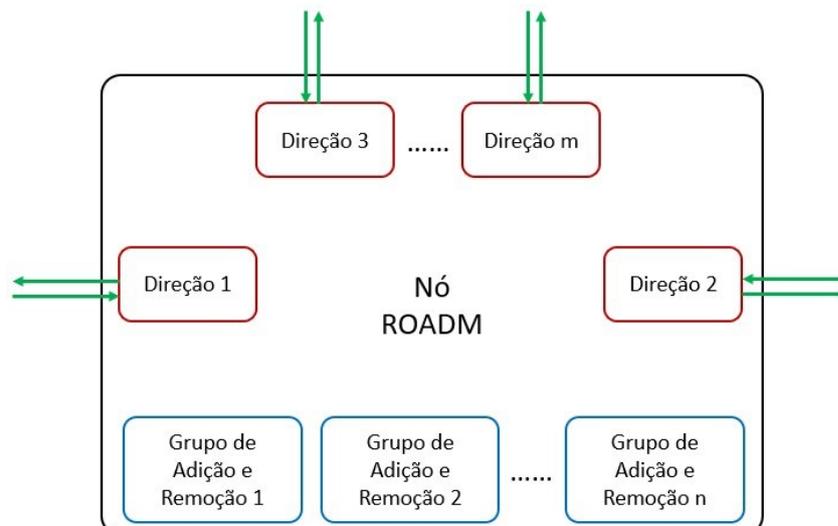
```

Fonte: Elaborada pelo autor.

4.3. Representação do Equipamento tipo ROADM

O modelo de equipamentos define um nó ROADM como um conjunto de placas DWDM que podem suportar um número máximo de direções e um número máximo de grupo de adição e remoção, também denominado como grupo compartilhado de risco (SRG, *Shared Risk Group*). A Figura 11 representa o diagrama de blocos de tal nó onde a quantidade máxima de direções m e SRGs n é fornecida pelos parâmetros *max-degrees* e *max-srgs* do módulo *info* do modelo. Cada bloco de direção ou SRG do diagrama é representado por uma lista de entidades do tipo *circuit-pack* do modelo, constituindo-se, portanto, por um conjunto de placas DWDM físicas que em conjunto desempenham a função de desagregação desejada.

Figura 11 – Diagrama de blocos de um nó ROADM.

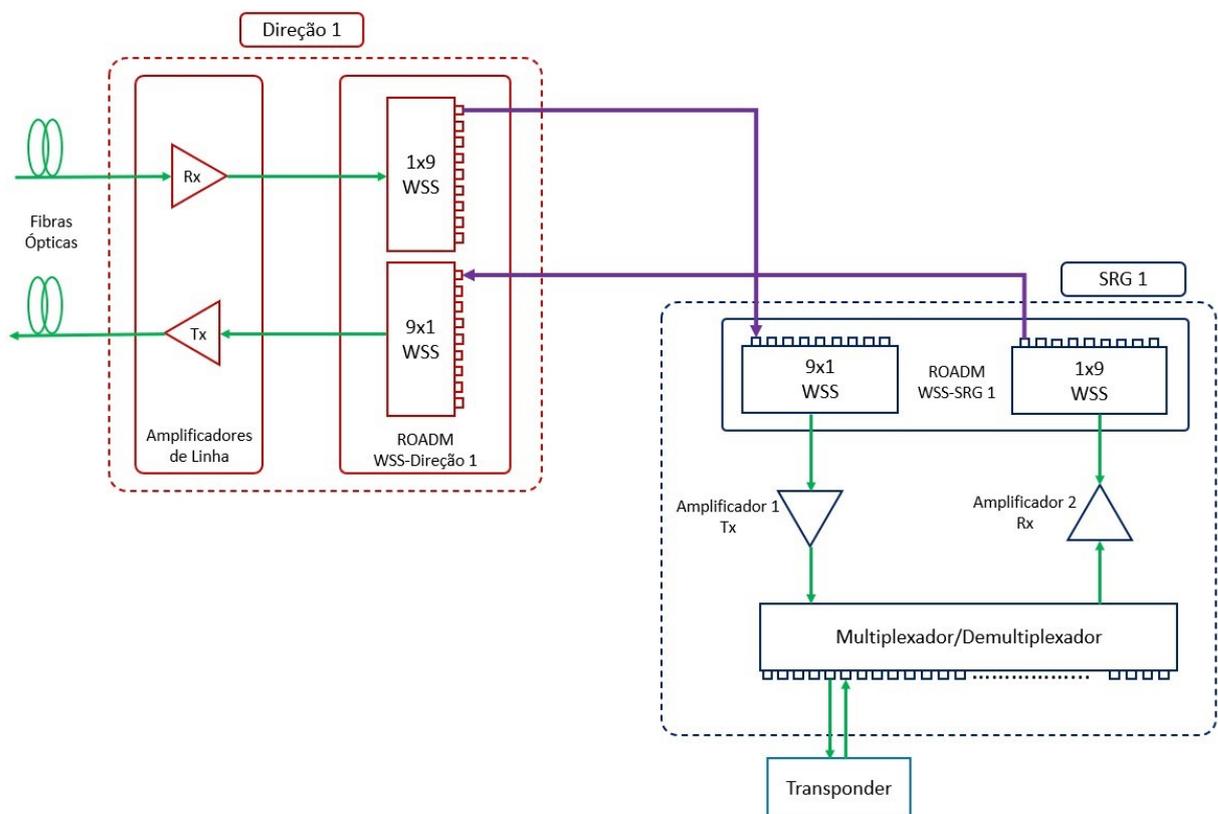


Fonte: Elaborada pelo autor, baseado em [56].

A documentação do OpenROADM [57] fornece um exemplo de como os blocos de direção e SRG devem ser desagregados em elementos DWDM. A Figura 12 ilustra os componentes

do bloco de direção composto por um ROADM direcional, contendo um amplificador na recepção Rx e outro na transmissão do enlace óptico designado como Tx, ambos interligados no módulo de WSS. Logo mais, a Figura 12 também apresenta a composição do bloco SRG que é constituído um ROADM e um MUX/DEMUX de vários canais. Os amplificadores ópticos nesse bloco são denominados como Amplificador 1 Tx e Amplificador 2 Rx, ambos necessários para compensar a perda do sinal óptico pelo módulo WSS. A principal função do bloco SRG é adicionar e remover canais ópticos do nó ROADM para os transponders locais a fim de permitir a interconexão entre diferentes estações. Portanto, um nó ROADM é uma coleção de vários blocos de direção e SRG interligados por fibras ópticas, internamente ou externamente.

Figura 12 – Composição de elementos DWDM dos blocos de direção e SRG em um nó ROADM.

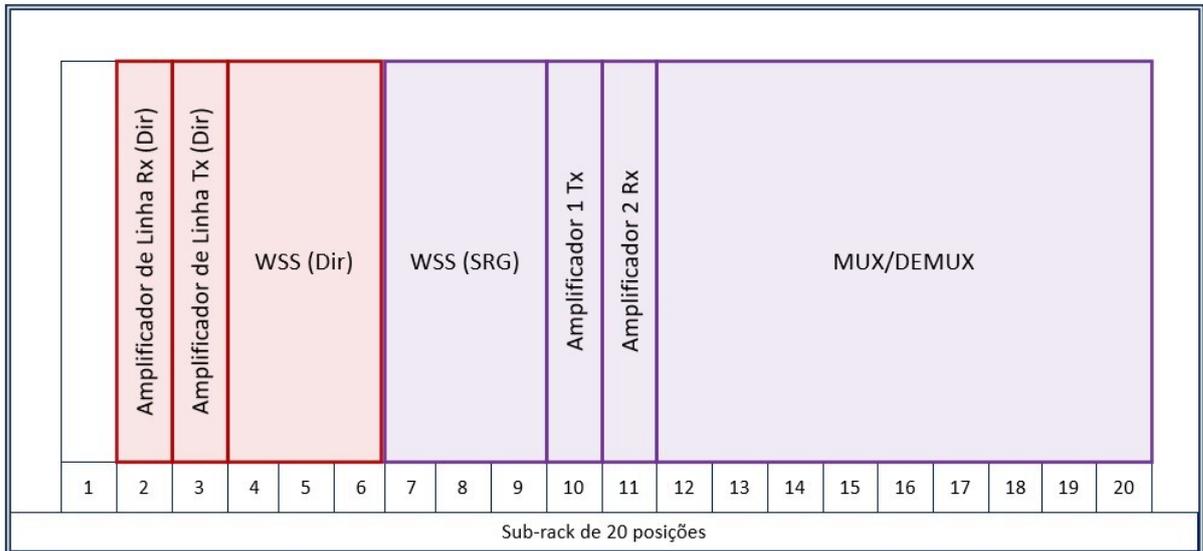


Fonte: Elaborada pelo autor, baseado em [57].

É importante mencionar que cada fabricante desenvolve um produto ROADM diferente com tecnologia proprietária. Porém, o modelo OpenROADM permite que qualquer produto seja desagregado por um conjunto de elementos DWDM em uma mesma placa física ou não, de certa forma, formando um nó ROADM genérico. Portanto, é responsabilidade do fabricante definir como seus equipamentos são desagregados fisicamente. A Figura 13 ilustra a configuração física usada nas simulações, consistindo-se de um sub-rack de 20 posições. Cada placa equipada no sub-rack ocupada um certo número de posições, por exemplo, a placa WSS de

3 posições e o MUX/DEMUX de 8 posições. Antes de modelar as placas dos blocos de direção e SRG, é necessário primeiro definir os sub-racks do nó ROADM.

Figura 13 – Placas físicas instaladas em um sub-rack de 20 posições.



Fonte: Elaborada pelo autor, baseado em [57].

4.4. Modelo do Sub-Rack: Módulo *shelves*

Nas estações de telecomunicações, os sub-racks são instalados em racks de geralmente 19 polegadas padronizados por diversas recomendações como o IEC 60297 [58], definido as dimensões físicas de racks, sub-racks e afins. No modelo do equipamento, não existe uma entidade exclusiva para representar racks enquanto os sub-racks são representados pelo módulo *shelves*. O Quadro 7 apresenta a representação de árvore dos atributos mais relevantes do modelo YANG definido para tal entidade. Todos os atributos contribuem para gerar uma visão desagregada dos sub-racks para o controlador SDN.

Quadro 7 – Árvore YANG reduzida de atributos do módulo *shelves*.

```

+--rw shelves* [shelf-name]
|   |--rw shelf-name          string
|   |--rw shelf-type         string
|   |--rw rack?              string
|   |--rw shelf-position?    string
|   |--rw administrative-state org-openroadm-equipment-states-types:admin-states
|   |--ro vendor              string
|   |--ro model               string
|   |--ro serial-id           string
|   |--ro type?               string
|   |--ro product-code?      string
|   |--ro manufacture-date?   yang:date-and-time
|   |--ro hardware-version?   string
|   |--ro operational-state    org-openroadm-common-state-types:state
|   |--rw equipment-state?    org-openroadm-equipment-states-types:states

```

```

| +-ro is-physical          boolean
| +-ro is-passive          boolean
| +-ro faceplate-label     string
| +-rw user-description?   string
| +-ro slots* [slot-name]

```

Fonte: Elaborada pelo autor.

Na simulação, dados fictícios de inventário de um sub-rack foram introduzidos no ConfD por meio da interface CLI. O atributo *slots* representa as entidades inseridas nas posições do sub-rack que será discutido posteriormente com mais detalhes. A mensagem RPC que o controlador SDN deve enviar para consultar o módulo *shelves* é apresentada no Quadro 8.

Quadro 8 – Mensagem RPC do NETCONF cliente para consulta do módulo *shelves*.

```

<get>
  <filter type="subtree">
    <org-openroadm-device xmlns="http://org/openroadm/device">
      <shelves/>
    </org-openroadm-device>
  </filter>
</get>

```

Fonte: Elaborada pelo autor.

A mensagem RPC de retorno do ConfD é mostrada no Quadro 9. É possível determinar que o sub-rack tem nome igual a 1 (*shelf-name*) e está instalado no rack de nome RACK_CANTAGALO (*rack*) na posição de rack 1A (*shelf-position*). Além disso, os atributos de inventário são exibidos como número de serial (*serial-id*), modelo (*model*), fabricante (*vendor*) e assim por diante. Note que todos esses dados foram fornecidos pelo o autor e são fictícios.

Quadro 9 – Resposta RPC do ConfD com as todas as informações do módulo *shelves*.

```

# netconf-console --rpc=query_shelves.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <org-openroadm-device xmlns="http://org/openroadm/device">
      <shelves>
        <shelf-name>1</shelf-name>
        <shelf-type>SHELF_ROADM_1</shelf-type>
        <rack>RACK_CANTAGALO</rack>
        <shelf-position>1A</shelf-position>
        <administrative-state>inService</administrative-state>
        <vendor>UFU</vendor>
        <model>subrack_20_posicoes</model>
        <serial-id>00001</serial-id>
        <type>shelf</type>
        <product-code>COD_12345</product-code>
        <manufacture-date>2020-10-21T12:30:33-00:00</manufacture-date>
        <hardware-version>1.0.0</hardware-version>
        <operational-state>inService</operational-state>
        <equipment-state>not-reserved-inuse</equipment-state>
        <is-physical>true</is-physical>
      </shelves>
    </org-openroadm-device>
  </data>
</rpc-reply>

```

```

    <is-passive>true</is-passive>
    <user-description>exemplo_de_um_sub-rack</user-description>
  </shelves>
</org-openroadm-device>
</data>
</rpc-reply>

```

Fonte: Elaborada pelo autor.

Após a criação da entidade *shelves* nos simuladores, foi possível continuar com a modelagem das placas físicas dos blocos de direção e SRG apresentados na Figura 13. Cada placa equipada no sub-rack é representada pelo módulo *circuit-pack* do modelo de equipamento. Observando a Figura 13, o bloco de direção é representado por três entidades *circuit-packs* que estão inseridos nas posições 2, 3 e 4 enquanto 4 entidades *circuit-packs* representam o bloco SRG. Em outros termos, a entidade *circuit-pack* desagrega funções ópticas de uma ou mais placas físicas, e inclusive pode representar transceptores.

4.5. Modelo da Placa: Módulo *circuit-pack*

O Quadro 10 apresenta o modelo YANG em forma de árvore das entidades mais relevantes do módulo *circuit-pack*. Além das informações de inventário e de estados, o modelo define que o tipo da entidade (*circuit-pack-category type*) e exige qual sub-rack (*shelf*) está instalado. Outra propriedade importante é a lista de portas do modelo. Cada porta do *circuit-pack* pode ter um nome (*port-name*), tipo (*port-type*), qualificação (*port-qual*) e etc. Além do mais, se a direção da porta (*port-direction*) for unidirecional, é recomendado que uma porta parceira (*port-partner-port*) seja referenciada para sistemas ópticos bidirecionais.

Quadro 10 – Árvore YANG reduzida de atributos do módulo *circuit-pack*.

```

+--rw circuit-packs* [circuit-pack-name]
|  +--rw circuit-pack-type          string
|  +--rw circuit-pack-name          string
|  +--rw administrative-state        ..equipment-states-types:admin-states
|  +--ro vendor                      string
|  +--ro model                       string
|  +--ro serial-id                   string
|  +--ro type?                       string
|  +--ro product-code?               string
|  +--ro operational-state            ..common-state-types:state
|  +--ro circuit-pack-category
|  |  +--ro type                      equipment-type-enum
|  +--rw equipment-state?            ..equipment-states-types:states
|  +--rw circuit-pack-mode?          string
|  +--rw shelf                       -> /org-openroadm-device/shelves/shelf-name
|  +--ro is-physical                 boolean
|  +--ro is-passive                  boolean
|  +--rw ports* [port-name]
|  |  +--rw port-name                string
|  |  +--rw port-type?               string

```

	+++rw port-qual?	..device-types:port-qual
	+++ro port-wavelength-type?	..port-types:port-wavelength-types
	+++ro port-direction	..common-alarm-pm-types:direction
	+++ro partner-port	
	+++ro circuit-pack-name?	circuit-pack-name
	+++ro port-name?	-> port-name

Fonte: Elaborada pelo autor.

Existem três tipos de placas instaladas no sub-rack apresentado na Figura 13: WSS, amplificador e MUX/DEMUX. Todas as placas foram criadas no ConfD por meio da interface CLI obedecendo suas posições no sub-rack. Por fim, será apresentado a resposta RPC de cada tipo de placa.

4.5.1. Modelo *circuit-pack*: Amplificador

O Quadro 11 exibe a configuração do amplificador de linha Rx do bloco de direção, servindo como exemplo de um *circuit-pack* do tipo amplificador. Primeiramente, o nome do *circuit-pack* (*circuit-pack-name*) é Dir1-RxAMP significando que esse é o amplificador Rx da direção 1. Além disso, tal elemento está instalado no sub-rack de nome 1 (*shelf*) e se encontra na posição 2 (*slot*) assim como está exibido na Figura 13. Identifica-se na resposta RPC também que existem duas portas nomeadas (*port-name*) como Dir1-RxAMP-OUT e Dir1-RxAMP-IN, representando a porta de saída e entrada do amplificador respectivamente. Cada porta possui uma direção (*port-direction*) unidirecional distinta e não se classificam como portas parceiras, pois esse amplificador é capaz de atuar em uma única direção.

Quadro 11 – Resposta RPC do ConfD com informações do *circuit-pack* Dir1-RxAMP.

```
# netconf-console --rpc=query_dir1_rxamp.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <org-openroadm-device xmlns="http://org/openroadm/device">
      <circuit-packs>
        <circuit-pack-name>Dir1-RxAMP</circuit-pack-name>
        <circuit-pack-type>amplificador</circuit-pack-type>
        <administrative-state>inService</administrative-state>
        <vendor>UFU</vendor>
        <model>amplificador_23dB</model>
        <serial-id>00001</serial-id>
        <operational-state>inService</operational-state>
        <circuit-pack-category>
          <type>circuitPack</type>
        </circuit-pack-category>
        <shelf>1</shelf>
        <slot>2</slot>
        <is-pluggable-optics>false</is-pluggable-optics>
        <is-physical>true</is-physical>
        <is-passive>false</is-passive>
        <ports>
```

```

    <port-name>Dir1-RxAMP-IN</port-name>
    <port-qual>ila-external</port-qual>
    <port-wavelength-type>multi-wavelength</port-wavelength-type>
    <port-direction>rx</port-direction>
    <is-physical>true</is-physical>
    <administrative-state>inService</administrative-state>
    <operational-state>inService</operational-state>
    <logical-connection-point>Dir1-TTP-Rx</logical-connection-point>
  </ports>
  <ports>
    <port-name>Dir1-RxAMP-OUT</port-name>
    <port-qual>ila-internal</port-qual>
    <port-wavelength-type>multi-wavelength</port-wavelength-type>
    <port-direction>tx</port-direction>
    <is-physical>true</is-physical>
    <administrative-state>inService</administrative-state>
    <operational-state>inService</operational-state>
  </ports>
</circuit-packs>
</org-openroadm-device>
</data>
</rpc-reply>

```

Fonte: Elaborada pelo autor.

4.5.2. Modelo *circuit-pack*: WSS

O próximo tipo de *circuit-pack* é o WSS ilustrado no Quadro 12, exemplificando o WSS do bloco de direção da Figura 13. É possível observar também que o nome do *circuit-pack* (*circuit-pack-name*) é Dir1-WSS e está instalado no sub-rack de nome 1 (*shelf*) e se encontra na posição 4 (*slot*).

A resposta RPC do Quadro 12 contém quatro portas nomeadas (*port-name*) como Dir1-WSS-OUT1, Dir1-WSS-IN1, Dir1-WSS-Tx e Dir1-WSS-Rx. O WSS simulado possui mais outras oito portas IN e também OUT apresentados na Figura 12, mas foram ocultadas da mensagem pois suas configurações são semelhantes às portas IN1 e OUT1. Para estabelecer um sistema óptico bidirecional, as portas OUT1 e IN1 operam como portas parceiras por meio do atributo *partner-port*, ou seja, uma porta faz referência da outra.

Quadro 12 – Resposta RPC do ConfD com informações do *circuit-pack* Dir1-WSS.

```

# netconf-console --rpc=query_dir1_wss.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <org-openroadm-device xmlns="http://org/openroadm/device">
      <circuit-packs>
        <circuit-pack-name>Dir1-WSS</circuit-pack-name>
        <circuit-pack-type>WSS</circuit-pack-type>
        <circuit-pack-product-code>00001</circuit-pack-product-code>
        <administrative-state>inService</administrative-state>
        <vendor>UFU</vendor>
      </circuit-packs>
    </org-openroadm-device>
  </data>
</rpc-reply>

```

```

<model>WSS_8_direcoes</model>
<serial-id>00001</serial-id>
<operational-state>inService</operational-state>
<circuit-pack-category>
  <type>circuitPack</type>
</circuit-pack-category>
<shelf>1</shelf>
<slot>4</slot>
<is-pluggable-optics>false</is-pluggable-optics>
<is-physical>true</is-physical>
<is-passive>false</is-passive>
<ports>
  <port-name>Dir1-WSS-IN1</port-name>
  <port-qual>roadm-internal</port-qual>
  <port-wavelength-type>multi-wavelength</port-wavelength-type>
  <port-direction>rx</port-direction>
  <is-physical>true</is-physical>
  <operational-state>inService</operational-state>
  <logical-connection-point>Dir1-CTP-Rx</logical-connection-point>
  <partner-port>
    <circuit-pack-name>Dir1-WSS</circuit-pack-name>
    <port-name>Dir1-WSS-OUT1</port-name>
  </partner-port>
</ports>
[...]
```

```

<ports>
  <port-name>Dir1-WSS-OUT1</port-name>
  <port-qual>roadm-internal</port-qual>
  <port-wavelength-type>multi-wavelength</port-wavelength-type>
  <port-direction>tx</port-direction>
  <is-physical>true</is-physical>
  <operational-state>inService</operational-state>
  <logical-connection-point>Dir1-CTP-Tx</logical-connection-point>
  <partner-port>
    <circuit-pack-name>Dir1-WSS</circuit-pack-name>
    <port-name>Dir1-WSS-IN1</port-name>
  </partner-port>
</ports>
[...]
```

```

<ports>
  <port-name>Dir1-WSS-Rx</port-name>
  <port-qual>roadm-external</port-qual>
  <port-wavelength-type>multi-wavelength</port-wavelength-type>
  <port-direction>rx</port-direction>
  <is-physical>true</is-physical>
  <operational-state>inService</operational-state>
</ports>
<ports>
  <port-name>Dir1-WSS-Tx</port-name>
  <port-qual>roadm-external</port-qual>
  <port-wavelength-type>multi-wavelength</port-wavelength-type>
  <port-direction>tx</port-direction>
  <is-physical>true</is-physical>
  <operational-state>inService</operational-state>
</ports>
</circuit-packs>
</org-openroadm-device>
</data>
</rpc-reply>

```

As portas Tx e Rx são qualificadas (*port-qual*) como *roadm-external* indicando que são as interfaces de entrada e saída para outros nós ROADMs. Por outro lado, as portas que são conectadas em outras portas internas ao nó ROADM são qualificadas como *roadm-internal*, por exemplo, as portas IN1 e OUT1.

O sinal óptico que é transmitido e recebido por meio das portas Tx e Rx contém vários canais distribuídos ao longo do espectro, e por isso, o tipo de comprimento de onda da porta (*port-wavelength-type*) é configurado para *multi-wavelength*. Por outro lado, as portas que operam com um único comprimento de onda são do tipo *wavelength*, indicado no Quadro 12 pelas portas IN1 e OUT1.

4.5.3. Modelo *circuit-pack*: MUX/DEMUX

A placa MUX/DEMUX é o último tipo de *circuit-pack* da Figura 13. O equipamento foi modelado para suportar 96 canais DWDM na grade fixa de 50 GHz, começando no canal C15 (191.5 THz) até no C62 (196.2 THz). O Quadro 13 apresenta a resposta RPC do ConfD para esse tipo de *circuit-pack* e foi nomeado (*circuit-pack-name*) como SRG1-MUX-DEMUX. Além disso, o equipamento está instalado no sub-rack de nome 1 (*shelf*) e se encontra na posição 12 (*slot*).

É mostrado na resposta apenas quatro portas nomeadas (*port-name*) como SRG1-MUX-DEMUX-IN1, SRG1-MUX-DEMUX-OUT1, SRG1-MUX-DEMUX-Tx e SRG1-MUX-DEMUX-Rx. Entretanto esse equipamento é equipado mais outras 190 portas para os outros 95 canais suportados da grade, mas não são exibidos no Quadro 13. Similarmente ao *circuit-pack* do WSS, as portas IN1 e OUT1 são portas parceiras (*partner-port*) para sistemas ópticos bidirecionais. As recomendações especificam que as portas do MUX/DEMUX deveriam ser qualificadas (*port-qual*) como *roadm-internal* somente se forem internas ao bloco do SRG, portanto, somente as portas Tx e Rx.

Quadro 13 – Resposta RPC do ConfD com informações do *circuit-pack* Dir1-MUX-DEMUX.

```
# netconf-console --rpc=query_dir1_mux_demux.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:iETF:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <org-openroadm-device xmlns="http://org/openroadm/device">
      <circuit-packs>
        <circuit-pack-name>SRG1-MUX-DEMUX</circuit-pack-name>
        <circuit-pack-type>MUX-DEMUX</circuit-pack-type>
        <administrative-state>inService</administrative-state>
        <vendor>UFU</vendor>
        <model>MUX_DEMUX_96_CANAIS</model>
        <serial-id>00001</serial-id>
```

```

<operational-state>inService</operational-state>
<circuit-pack-category>
  <type>circuitPack</type>
</circuit-pack-category>
<shelf>1</shelf>
<slot>12</slot>
<is-pluggable-optics>false</is-pluggable-optics>
<is-physical>true</is-physical>
<is-passive>true</is-passive>
<ports>
  <port-name>SRG1-MUX-DEMUX-IN1</port-name>
  <port-direction>rx</port-direction>
  <is-physical>true</is-physical>
  <operational-state>inService</operational-state>
  <partner-port>
    <circuit-pack-name>SRG1-MUX-DEMUX</circuit-pack-name>
    <port-name>SRG1-MUX-DEMUX-OUT1</port-name>
  </partner-port>
</ports>
<ports>
  <port-name>SRG1-MUX-DEMUX-OUT1</port-name>
  <port-direction>tx</port-direction>
  <is-physical>true</is-physical>
  <operational-state>inService</operational-state>
  <partner-port>
    <circuit-pack-name>SRG1-MUX-DEMUX</circuit-pack-name>
    <port-name>SRG1-MUX-DEMUX-IN1</port-name>
  </partner-port>
</ports>
[... ]
<ports>
  <port-name>SRG1-MUX-DEMUX-Rx</port-name>
  <port-qual>roadm-internal</port-qual>
  <port-direction>rx</port-direction>
  <is-physical>true</is-physical>
  <operational-state>inService</operational-state>
</ports>
<ports>
  <port-name>SRG1-MUX-DEMUX-Tx</port-name>
  <port-qual>roadm-internal</port-qual>
  <port-direction>tx</port-direction>
  <is-physical>true</is-physical>
  <operational-state>inService</operational-state>
</ports>
</circuit-packs>
</org-openroadm-device>
</data>
</rpc-reply>

```

Fonte: Elaborada pelo autor.

4.5.4. Modelo da Posição do Sub-Rack: Módulo *slot-info*

O módulo de posições (*slot-info*) complementa o módulo do sub-rack para informar quais são as posições do sub-rack que estão vazias ou com placas instaladas. Esse tipo de informação permite que o usuário tenha uma visão realística de todos os equipamentos da estação,

bem como, onde estão posicionadas. Portanto, esse módulo pode ser inserido nos simuladores após a criação dos *circuit-packs* no banco de dados do ConfD.

O Quadro 14 exibe a resposta RPC reduzida com dados de algumas posições, estas identificadas com o nome do slot (*slot-name*), placa instalada no local (*provisioned-circuit-pack*) e o estado da posição (*slot-status*). Por exemplo, o amplificador Rx da direção 1 cujo o nome do *circuit-pack* é Dir1-RxAMP está equipado na posição 1 do sub-rack com o estado de instalado provisionado (*installed-prov-match*).

É importante mencionar que o modelo do equipamento permite que o estado da posição assuma outros valores como vazio não provisionado (*empty-not-prov*), vazio provisionado (*empty-prov*) e instalado provisionado incompatível (*installed-prov-mismatch*). Tais estados são usados para contribuir com o sucesso da instalação e comissionamento de placas nas estações, bem como, a pré-configuração e simulação de serviços mesmo com os equipamentos físicos ainda não prontos para uso.

Quadro 14 – Resposta RPC do ConfD do módulo *shelves* com informações de posições.

```
# netconf-console --rpc=query_shelves.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <org-openroadm-device xmlns="http://org/openroadm/device">
      <shelves>
        <shelf-name>1</shelf-name>
        [...]
        <slots>
          <slot-name>1</slot-name>
          <provisioned-circuit-pack>Dir1-RxAMP</provisioned-circuit-pack>
          <slot-status>installed-prov-match</slot-status>
        </slots>
        <slots>
          <slot-name>2</slot-name>
          <provisioned-circuit-pack>Dir1-TxAMP</provisioned-circuit-pack>
          <slot-status>installed-prov-match</slot-status>
        </slots>
        <slots>
          <slot-name>3</slot-name>
          <provisioned-circuit-pack>Dir1-WSS</provisioned-circuit-pack>
          <slot-status>installed-prov-match</slot-status>
        </slots>
        <slots>
          <slot-name>4</slot-name>
          <slot-status>installed-prov-match</slot-status>
        </slots>
        [...]
        <slots>
          <slot-name>12</slot-name>
          <provisioned-circuit-pack>SRG1-MUX-DEMUX</provisioned-circuit-pack>
          <slot-status>installed-not-prov</slot-status>
        </slots>
      </shelves>
    </org-openroadm-device>
  </data>
</rpc-reply>
```

```

    </shelves>
  </org-openroadm-device>
</data>
</rpc-reply>

```

Fonte: Elaborada pelo autor.

4.6. Modelo da Direção: Módulo *degree*

O modelo da direção é dado pelo Quadro 15 com os seguintes atributos relevantes: número da direção (*degree-number*), número máximo de comprimento de ondas (*max-wavelengths*), lista de *circuit-packs*, portas de conexão (*connection-ports*) e perfil de capacidade de canais (*mc-capability-profile*). O número da direção é um número inteiro positivo menor que a capacidade máxima suportada do nó ROADM modelado. Além disso, a documentação do equipamento [59] sugere que as portas de conexões desse módulo devem ser apenas aquelas são interligadas com as fibras de transmissão.

Quadro 15 – Árvore YANG reduzida de atributos do módulo *degree*.

```

+--rw degree* [degree-number]
  |   +--rw degree-number          uint16
  |   +--ro max-wavelengths       uint16
  |   +--rw circuit-packs* [index]
  |   |   +--rw index              uint32
  |   |   +--rw circuit-pack-name  -> circuit-pack-name
  |   +--rw connection-ports* [index]
  |   |   +--rw index              uint32
  |   |   +--rw circuit-pack-name  -> circuit-pack-name
  |   |   +--rw port-name          -> port-name
  |   +--ro mc-capability-profile-name* -> mc-capability-profile-name

```

Fonte: Elaborada pelo autor.

Os dados da direção 1 foram inseridos no ConfD e a resposta RPC desse módulo é exibida no Quadro 16. É possível observar que a direção 1 é definida pelo atributo *degree-number* e suporta até 96 comprimentos de ondas. A lista de *circuit-packs* é composta por três entidades previamente criadas que são Dir1-WSS, Dir1-TxAMP e Dir1-Rx-AMP, apresentados na Figura 12. Acrescenta-se também as portas de conexões definidas na resposta RPC são as portas ligadas nas fibras de transmissão de linha, Dir1-RxAMP-IN e Dir1-TxAMP-OUT.

Quadro 16 – Resposta RPC do ConfD com as todas as informações do módulo *degree*.

```

# netconf-console --rpc=query_degree.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <org-openroadm-device xmlns="http://org/openroadm/device">
      <degree>
        <degree-number>1</degree-number>
        <max-wavelengths>96</max-wavelengths>

```

```

    <circuit-packs>
      <index>1</index>
      <circuit-pack-name>Dir1-WSS</circuit-pack-name>
    </circuit-packs>
    <circuit-packs>
      <index>2</index>
      <circuit-pack-name>Dir1-TxAMP</circuit-pack-name>
    </circuit-packs>
    <circuit-packs>
      <index>3</index>
      <circuit-pack-name>Dir1-RxAMP</circuit-pack-name>
    </circuit-packs>
    <connection-ports>
      <index>1</index>
      <circuit-pack-name>Dir1-RxAMP</circuit-pack-name>
      <port-name>Dir1-RxAMP-IN</port-name>
    </connection-ports>
    <connection-ports>
      <index>2</index>
      <circuit-pack-name>Dir1-TxAMP</circuit-pack-name>
      <port-name>Dir1-TxAMP-OUT</port-name>
    </connection-ports>
    <mc-capability-profile-name>fixed-grid</mc-capability-profile-name>
  </degree>
</org-openroadm-device>
</data>
</rpc-reply>

```

Fonte: Elaborada pelo autor.

O último atributo *mc-capability-profile-name* define o tipo de perfil nomeado *fixed-grid*. O Quadro 17 apresenta dois perfis cadastrados no ConfD: um para espaçamento fixo de 50 GHz de nome *fixed-grid* e outro de espaçamento flexível nomeado como *flex-grid*. Os atributos apresentados na resposta RPC fornecem ao controlador SDN uma visão da alocação do espectro dos blocos de direção e SRG, permitindo a otimização da utilização da rede. A tecnologia de espaçamento escolhida para os simulares será fixa de 50 GHz, permitindo alocação de canais entre C15 e C62.

Quadro 17 – Resposta RPC do ConfD com as todas as informações do módulo *mc-capability-profile*.

```

# netconf-console --rpc=query_mc_profiles.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <org-openroadm-device xmlns="http://org/openroadm/device">
      <mc-capability-profile>
        <profile-name>fixed-grid</profile-name>
        <center-freq-granularity>50.0</center-freq-granularity>
        <slot-width-granularity>50.0</slot-width-granularity>
        <min-slots>1</min-slots>
        <max-slots>1</max-slots>
      </mc-capability-profile>
      <mc-capability-profile>
        <profile-name>flex-grid</profile-name>
        <center-freq-granularity>6.25</center-freq-granularity>
      </mc-capability-profile>
    </org-openroadm-device>
  </data>
</rpc-reply>

```

```

<slot-width-granularity>12.5</slot-width-granularity>
<min-slots>3</min-slots>
<max-slots>384</max-slots>
</mc-capability-profile>
</org-openroadm-device>
</data>
</rpc-reply>

```

Fonte: Elaborada pelo autor.

4.7. Modelo do SRG: Módulo *shared-risk-group*

O Quadro 18Quadro 15 apresenta os atributos mais relevantes do módulo do SRG, sendo caracterizado por pelo número do SRG (*srg-number*) que deve ser menor que o máximo suportado pelo nó ROADM. Os atributos do número máximo de adição e remoção (*max-add-drop-ports*) e o número atual provisionado (*current-provisioned-add-drop-ports*) informam sua capacidade máxima e atual de portas do bloco SRG. Como o modelo de direção, é possível configurar a lista de *circuit-packs* pertencentes a esse bloco, assim como, seu perfil de capacidade de canais (*mc-capability-profile*).

Quadro 18 – Árvore YANG reduzida de atributos do módulo *shared-risk-group*.

```

+--rw shared-risk-group* [srg-number]
|   +--ro max-add-drop-ports          uint16
|   +--ro current-provisioned-add-drop-ports  uint16
|   +--rw srg-number                  uint16
|   +--ro wavelength-duplication      ..wavelength-duplication-type
|   +--rw circuit-packs* [index]
|   |   +--rw index                  uint32
|   |   +--rw circuit-pack-name      -> circuit-pack-name
|   +--ro mc-capability-profile-name* -> mc-capability-profile-name

```

Fonte: Elaborada pelo autor.

O SRG de número (*srg-number*) inserido no ConfD é apresentado no Quadro 19. Os *circuit-packs* que foram tal entidade são SRG1-WSS, SRG1-AMP1, SRG1-AMP2 e SRG1-MUX-DEMUX, os quais presentes na Figura 12. Na resposta RPC, nenhuma das 96 portas suportadas (*max-add-drop-ports*) está sendo utilizada no bloco SRG pois o valor do atributo *current-provisioned-add-drop-ports* é igual a zero. Além disso, o perfil de capacidade de canais escolhido é de 50 GHz (*fixed-grid*) já discutido no Quadro 17. O atributo de duplicação de comprimento de onda (*wavelength-duplication*) pode assumir os valores de *one-per-srg* informando que o SRG suporta a tecnologia C/D/C ou *one-per-degree*, operando no modo C/D. Foi configurado o valor *one-per-srg* nos simuladores desse trabalho.

Quadro 19 – Resposta RPC do ConfD com as todas as informações do módulo *shared-risk-group*.

```

# netconf-console --rpc=query_srg.xml
<?xml version="1.0" encoding="UTF-8"?>

```

```

<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <org-openroadm-device xmlns="http://org/openroadm/device">
      <shared-risk-group>
        <srg-number>1</srg-number>
        <max-add-drop-ports>96</max-add-drop-ports>
        <current-provisioned-add-drop-ports>0</current-provisioned-add-drop-ports>
        <wavelength-duplication>one-per-srg</wavelength-duplication>
        <circuit-packs>
          <index>1</index>
          <circuit-pack-name>SRG1-WSS</circuit-pack-name>
        </circuit-packs>
        <circuit-packs>
          <index>2</index>
          <circuit-pack-name>SRG1-AMP1</circuit-pack-name>
        </circuit-packs>
        <circuit-packs>
          <index>3</index>
          <circuit-pack-name>SRG1-AMP2</circuit-pack-name>
        </circuit-packs>
        <circuit-packs>
          <index>4</index>
          <circuit-pack-name>SRG1-MUX-DEMUX</circuit-pack-name>
        </circuit-packs>
        <mc-capability-profile-name>fixed-grid</mc-capability-profile-name>
      </shared-risk-group>
    </org-openroadm-device>
  </data>
</rpc-reply>

```

Fonte: Elaborada pelo autor.

4.8. Modelo das Ligações Internas: Módulo *internal-link*

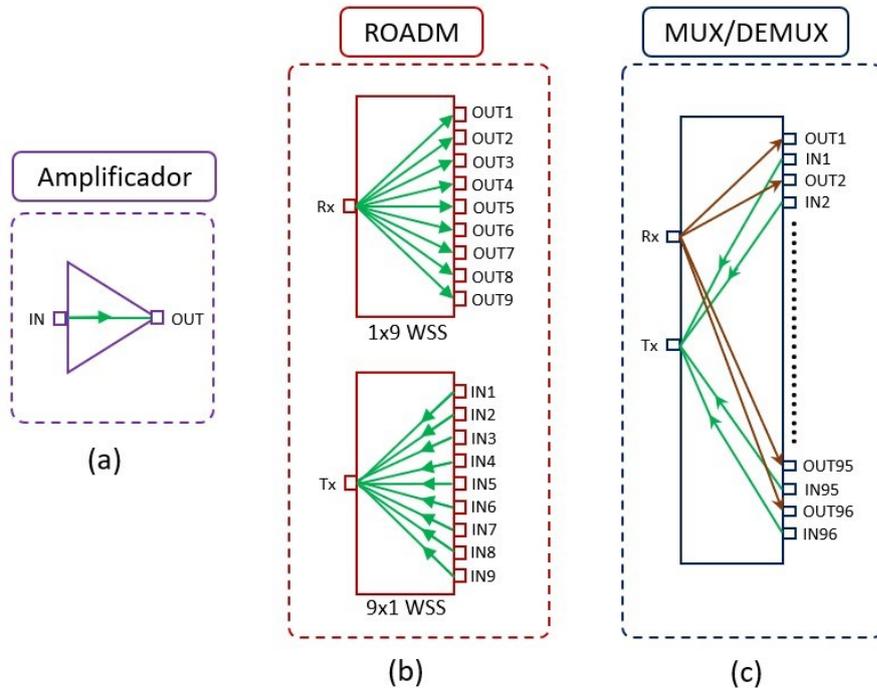
A documentação do modelo de equipamento [57] sugere que o módulo de ligações internas (*internal-link*) de cada *circuit-pack* seja preenchido conforme a trajetória do tráfego dentro do equipamento simulado. As ligações físicas de cada tipo de placa é ilustrada na Figura 14 (a) para o amplificador, (b) ROADM e (c) MUX/DEMUX. Existe somente uma ligação interna para modelo do amplificador que interliga a porta IN para a porta OUT. Note que a tecnologia proprietária usada para construir o amplificador óptico não é uma informação desagregada, sendo necessário apenas representar o caminho do tráfego.

O ROADM simulado, exibido na Figura 14 (b), possui ligações internas da porta Rx para todas as portas OUT, assim como na outra direção, de todas as portas IN para a porta Tx. É importante destacar que a ligação interna deve ser registrada de acordo com a direção do tráfego. Por exemplo, uma ligação interna do ROADM da porta OUT1 para a porta Rx está incorreta pois a direção do tráfego acontece da porta Rx para a porta OUT1.

A placa MUX/DEMUX é o tipo de *circuit-pack* que contém mais ligações internas devido ao grande número de portas IN e OUT como ilustrado na Figura 14 (c). Cada conjunto de

porta parceira, por exemplo IN1 e OUT1, suportam apenas um comprimento de onda em um sistema óptico bidirecional. Do mesmo modo do ROADM, as ligações internas são registradas da direção Rx para as portas OUT e da direção das portas IN para a porta Tx.

Figura 14 – Ligações internas de cada tipo de *circuit-pack* (a) amplificador, (b) ROADM e (c) MUX/DEMUX.



Fonte: Elaborada pelo autor, baseado em [57].

O modelo da ligação interna é dado pelo módulo *internal-link* e é composto pelas referências da porta de origem para a porta de destino. Para isso é necessário providenciar no ConfD os nomes dos *circuit-packs* e as portas envolvidas. O Quadro 20 ilustra partes da resposta RPC do módulo *internal-link* destacando-se as ligações internas de nomes (*internal-link-name*) Dir1-RxAMP-IN-OUT, Dir1-WSS-IN1-Tx e SRG1-MUX-DEMUX-IN1-Tx. Respectivamente tais ligações correspondem ao *circuit-packs* do amplificador, WSS e MUX/DEMUX. Para exemplificar, a ligação Dir1-WSS-Rx-OUT1 reflete ao *circuit-pack* de nome Dir1-WSS para a porta de origem Dir1-WSS-Rx (Rx) para a porta de destino Dir1-WSS-OUT1 (OUT1) como é mostrado na Figura 14 (b).

Quadro 20 – Resposta RPC do ConfD com informações do módulo *internal-link*.

```
# netconf-console --rpc=query_internal_links.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <org-openroadm-device xmlns="http://org/openroadm/device">
      <internal-link>
        <internal-link-name>Dir1-RxAMP-IN-OUT</internal-link-name>
        <source>
          <circuit-pack-name>Dir1-RxAMP</circuit-pack-name>
```

```

    <port-name>Dir1-RxAMP-IN</port-name>
  </source>
  <destination>
    <circuit-pack-name>Dir1-RxAMP</circuit-pack-name>
    <port-name>Dir1-RxAMP-OUT</port-name>
  </destination>
</internal-link>
[... ]
<internal-link>
  <internal-link-name>Dir1-WSS-IN1-Tx</internal-link-name>
  <source>
    <circuit-pack-name>Dir1-WSS</circuit-pack-name>
    <port-name>Dir1-WSS-IN1</port-name>
  </source>
  <destination>
    <circuit-pack-name>Dir1-WSS</circuit-pack-name>
    <port-name>Dir1-WSS-Tx</port-name>
  </destination>
</internal-link>
[... ]
<internal-link>
  <internal-link-name>Dir1-WSS-Rx-OUT1</internal-link-name>
  <source>
    <circuit-pack-name>Dir1-WSS</circuit-pack-name>
    <port-name>Dir1-WSS-Rx</port-name>
  </source>
  <destination>
    <circuit-pack-name>Dir1-WSS</circuit-pack-name>
    <port-name>Dir1-WSS-OUT1</port-name>
  </destination>
</internal-link>
[... ]
<internal-link>
  <internal-link-name>SRG1-MUX-DEMUX-IN1-Tx</internal-link-name>
  <source>
    <circuit-pack-name>SRG1-MUX-DEMUX</circuit-pack-name>
    <port-name>SRG1-MUX-DEMUX-IN1</port-name>
  </source>
  <destination>
    <circuit-pack-name>SRG1-MUX-DEMUX</circuit-pack-name>
    <port-name>SRG1-MUX-DEMUX-Tx</port-name>
  </destination>
</internal-link>
[... ]
</org-openroadm-device>
</data>
</rpc-reply>

```

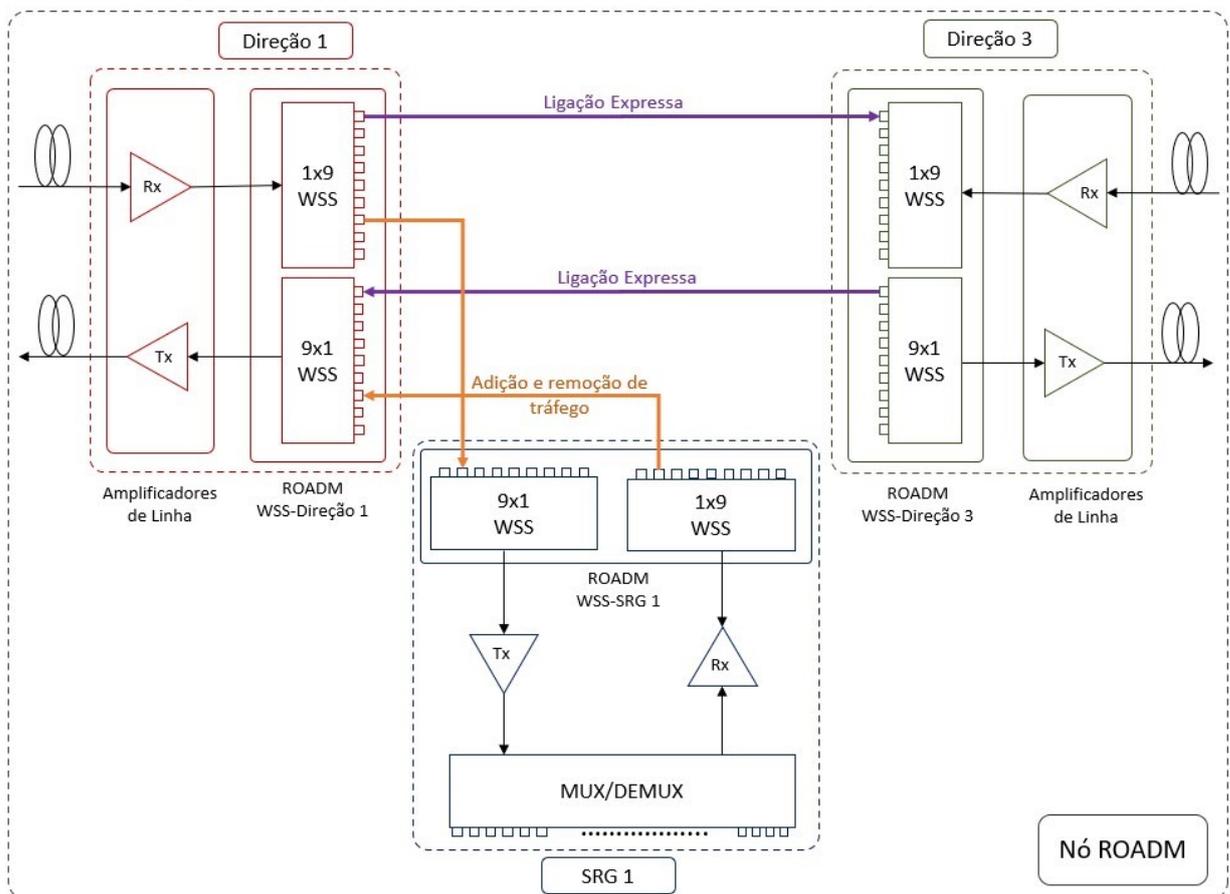
Fonte: Elaborada pelo autor.

4.9. Modelo das Ligações Físicas: Módulo *physical-link*

Dado que um nó ROADM é um conjunto de blocos de direções e SRGs, o modelo das ligações físicas (*physical-link*) representa uma entidade que interliga diferentes blocos de direção e SRG, bem como qualquer ligação entre os *circuit-packs*. A Figura 15 ilustra um nó ROADM com um SRG índice 1 e com duas direções de números 1 e 3, os quais são interligados por

diferentes enlaces ópticos caracterizando as ligações físicas. Recomenda-se a padronização de nomes de três tipos de ligações físicas de acordo com a documentação do modelo [56] que são as ligações expressas (*express links*), ligações de remoção (*drop links*), e por fim, as ligações de adição (*add links*). As outras ligações físicas não seguem nenhuma padronização como a fibra que conecta o amplificador Rx ao ROADM de um bloco de direção.

Figura 15 – Ligações físicas entre os blocos de direções e SRG de um nó ROADM.



Fonte: Elaborada pelo autor, baseado em [57].

A ligação expressa é o enlace de fibras entre duas direções distintas e deve ser nomeado como *ExpLink_{xy}*, onde *x* é o número da direção de origem e *y* de destino. Por exemplo, a ligação expressa de origem da direção 1 para a direção 3 da Figura 15 é designada como *ExpLink₁₃*, ao contrário do outro sentido do tráfego que é intitulado de *ExpLink₃₁*.

As ligações de adição são as fibras ópticas que interligam os blocos de direção para os blocos de SRG enquanto as ligações de remoção fazem o sentido oposto, ou seja, dos blocos SRG para os blocos de direção. Os nomes de tais ligações físicas são definidas por *ALink_{xy}* e *Dlink_{xy}* para as ligações de adição e remoção respectivamente, onde *x* e *y* são valores substituídos pelo número do bloco de direção ou SRG de acordo com o sentido desejado.

Devido os cenários de casos de uso das placas ROADM, convencionalmente recomenda-se [59] reservar as primeiras portas de cada ROADM direcional para interligar com outros ROADM direcionais no mesmo nó ROADM. Enquanto as portas restantes devem ser direcionadas para os blocos SRGs. Em outros termos, as ligações expressas são reservadas nas primeiras portas dos blocos de direção ao mesmo tempo que as ligações de adição e remoção são alocadas para as portas finais.

O Quadro 21 apresenta o módulo das ligações físicas (*physical-link*) com os atributos mais relevantes: nome da ligação física (*physical-link-name*), equipamento físico (*is-physical*) e portas dos *circuit-packs* de origem (*source*) e destino (*destination*). De acordo com a recomendação [57], os enlaces físicos devem ter o atributo de equipamento físico igual a verdadeiro ao contrário das ligações lógicas que são controladas pelo modelo da rede com valor igual a falso.

Quadro 21 – Árvore YANG reduzida de atributos do módulo *physical-link*.

```

+--rw physical-link* [physical-link-name]
  |   +--rw physical-link-name      string
  |   +--rw is-physical?            boolean
  |   +--rw source
  |   |   +--rw circuit-pack-name    -> circuit-pack-name
  |   |   +--rw port-name           -> port-name
  |   +--rw destination
  |   |   +--rw circuit-pack-name    -> circuit-pack-name
  |   |   +--rw port-name           -> port-name

```

Fonte: Elaborada pelo autor.

Foram inseridos no ConfD as ligações físicas apresentadas na Figura 15 que consistem em uma ligação expressa, uma ligação de adição e outra de remoção e as ligações físicas que interligam cada *circuit-pack* dentro de cada bloco. O Quadro 22 ilustra algumas ligações físicas nomeadas nesta ordem: Alink11, Dlink11, ExpLink13, ExpLink31 e Dir1-TxLink. A primeira ligação Alink11 indica que é uma ligação de adição da porta SRG1-WSS-OUT2 para a porta Dir1-WSS-IN7, sendo Dlink 11 para o sentido contrário como mostrado na Figura 15. As ligações expressas ExpLink13 e ExpLink31 representam as fibras ópticas interligando os blocos de direção 1 e 3. Por fim, a fibra óptica que interliga a porta Dir1-WSS-Tx do ROADM Dir1-WSS à porta Dir1-TxAMP-IN do amplificador Dir1-TxAMP é representada pela ligação física de nome Dir1-TxLink.

Quadro 22 – Resposta RPC do ConfD com as informações do módulo *physical-link*.

```

# netconf-console --rpc=query_physical_links.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>

```

```

<org-openroadm-device xmlns="http://org/openroadm/device">
  <physical-link>
    <physical-link-name>Alink11</physical-link-name>
    <is-physical>true</is-physical>
    <source>
      <circuit-pack-name>SRG1-WSS</circuit-pack-name>
      <port-name>SRG1-WSS-OUT2</port-name>
    </source>
    <destination>
      <circuit-pack-name>Dir1-WSS</circuit-pack-name>
      <port-name>Dir1-WSS-IN7</port-name>
    </destination>
  </physical-link>
  <physical-link>
    <physical-link-name>DLink11</physical-link-name>
    <is-physical>true</is-physical>
    <source>
      <circuit-pack-name>Dir1-WSS</circuit-pack-name>
      <port-name>Dir1-WSS-OUT7</port-name>
    </source>
    <destination>
      <circuit-pack-name>SRG1-WSS</circuit-pack-name>
      <port-name>SRG1-WSS-IN2</port-name>
    </destination>
  </physical-link>
  [...]
  <physical-link>
    <physical-link-name>ExpLink13</physical-link-name>
    <is-physical>true</is-physical>
    <source>
      <circuit-pack-name>Dir1-WSS</circuit-pack-name>
      <port-name>Dir1-WSS-OUT1</port-name>
    </source>
    <destination>
      <circuit-pack-name>Dir3-WSS</circuit-pack-name>
      <port-name>Dir3-WSS-IN1</port-name>
    </destination>
  </physical-link>
  <physical-link>
    <physical-link-name>ExpLink31</physical-link-name>
    <is-physical>true</is-physical>
    <source>
      <circuit-pack-name>Dir3-WSS</circuit-pack-name>
      <port-name>Dir3-WSS-OUT1</port-name>
    </source>
    <destination>
      <circuit-pack-name>Dir1-WSS</circuit-pack-name>
      <port-name>Dir1-WSS-IN1</port-name>
    </destination>
  </physical-link>
  [...]
  <physical-link>
    <physical-link-name>Dir1-TxLink</physical-link-name>
    <is-physical>true</is-physical>
    <source>
      <circuit-pack-name>Dir1-WSS</circuit-pack-name>
      <port-name>Dir1-WSS-Tx</port-name>
    </source>
    <destination>

```

```

<circuit-pack-name>Dir1-TxAMP</circuit-pack-name>
<port-name>Dir1-TxAMP-IN</port-name>
</destination>
</physical-link>
</org-openroadm-device>
</data>
</rpc-reply>

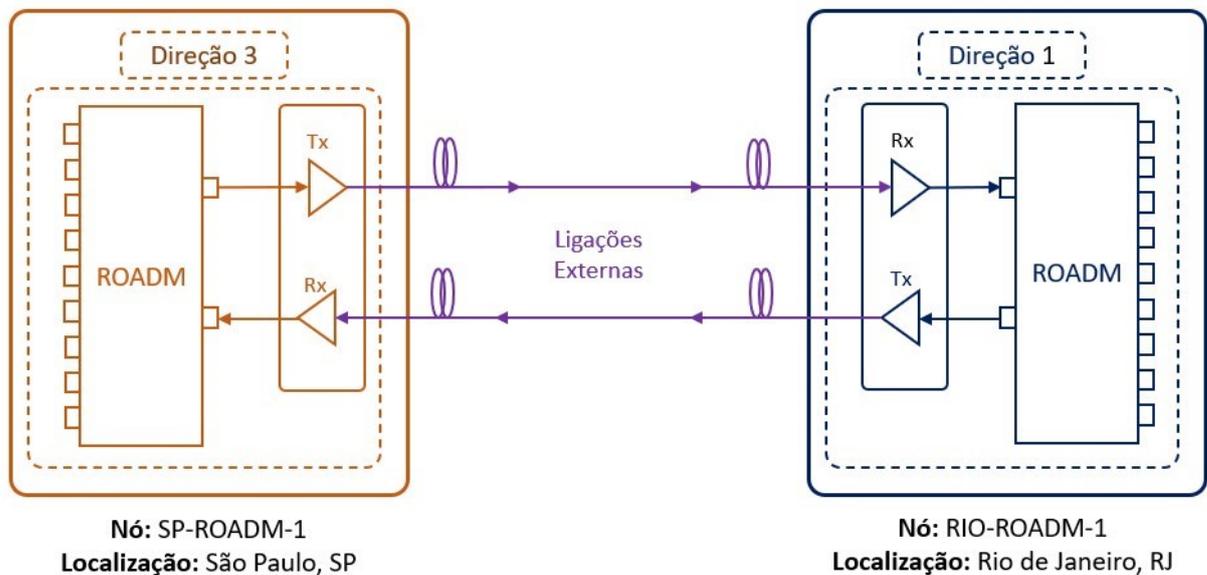
```

Fonte: Elaborada pelo autor.

4.10. Modelo das Ligações Externas: Módulo *external-link*

O último tipo de ligações do modelo OpenROADM são as ligações externas (*external-link*) e são caracterizados pelas conexões entre nós ROADMs e entre nós ROADMs e transponders. A Figura 16 apresenta dois nós ROADMs identificados como SP-ROADM-1 e RIO-ROADM-1, localizados em São Paulo e Rio de Janeiro respectivamente. Existem duas ligações externas, uma em cada direção formando um sistema bidirecional, para interligar os blocos de direções de cada nó ROADM. O controlador SDN pode automaticamente descobrir a topologia instalada da rede entre as estações de telecomunicações através do modelo de ligações externas.

Figura 16 – Ligações externas entre dois nós ROADMs de localizações diferentes.



Fonte: Elaborada pelo autor.

As ligações externas apresentadas na Figura 16 foram criadas no ConfD do nó RIO-ROADM-1. O Quadro 23 apresenta a resposta RPC que exibe todas as informações do módulo de ligações externas. Existem duas ligações externas de nome (*external-link-name*) ExtLink-RIO-ROADM-1-Dir1-SP-ROADM-1-Dir3 e ExtLink-SP-ROADM-1-Dir3-RIO-ROADM-1-Dir1. Por exemplo, a primeira ligação tem como nó de origem (*source node-id*) RIO-ROADM-1 com a porta (*source port-name*) Dir1-TxAMP-OUT da direção 1 e tem como nó de destino (*destination node-*

id) SP-ROADM-1 com a porta (*destination port-name*) Dir3-RxAMP-IN da direção 3. A segunda ligação é representada pela linha de transmissão do nó de São Paulo para o nó de Rio de Janeiro.

Quadro 23 – Resposta RPC do ConfD com as informações do módulo *external-link*.

```
# netconf-console --rpc=query_external_links.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <org-openroadm-device xmlns="http://org/openroadm/device">
      <external-link>
        <external-link-name>ExtLink-RIO-ROADM-1-Dir1-SP-ROADM-1-Dir3</external-link-name>
        <source>
          <node-id>RIO-ROADM-1</node-id>
          <circuit-pack-name>Dir1-TxAMP</circuit-pack-name>
          <port-name>Dir1-TxAMP-OUT</port-name>
        </source>
        <destination>
          <node-id>SP-ROADM-1</node-id>
          <circuit-pack-name>Dir3-RxAMP</circuit-pack-name>
          <port-name>Dir3-RxAMP-IN</port-name>
        </destination>
      </external-link>
      <external-link>
        <external-link-name>ExtLink-SP-ROADM-1-Dir3-RIO-ROADM-1-Dir1</external-link-name>
        <source>
          <node-id>SP-ROADM-1</node-id>
          <circuit-pack-name>Dir3-TxAMP</circuit-pack-name>
          <port-name>Dir3-TxAMP-OUT</port-name>
        </source>
        <destination>
          <node-id>RIO-ROADM-1</node-id>
          <circuit-pack-name>Dir1-RxAMP</circuit-pack-name>
          <port-name>Dir1-RxAMP-IN</port-name>
        </destination>
      </external-link>
    </org-openroadm-device>
  </data>
</rpc-reply>
```

Fonte: Elaborada pelo autor.

4.11. Representação do Equipamento tipo Transponder

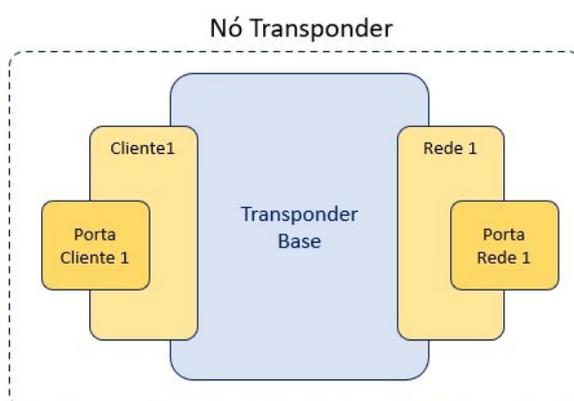
O modelo de equipamentos define um nó transponder como uma entidade que contém um conjunto de portas clientes, as quais recebem os sinais de baixa ordem, e uma lista de portas redes que transmitem um sinal OTN multiplexando os sinais clientes. Para cada porta, o transponder pode ter um transceptor externo ou módulo óptico interno. De qualquer forma, a entidade *circuit-pack* do modelo de equipamentos deve ser utilizada para representar cada porta do nó transponder com suas funcionalidades ópticas definidas. Caso um transceptor externo seja

instalado em alguma porta, este também deve ser representado como uma entidade *circuit-pack* com a propriedade de validação de transceptor (*is-pluggable-optics*) com valor igual a verdadeiro.

O modelo do transponder é capaz de desagregar diferentes arquiteturas e tecnologias das placas dos fabricantes como o muxponder e o OTN switch. O muxponder possui uma matriz fixa de crossconexão entre as portas clientes e redes. Por exemplo, um muxponder 100G é capaz de multiplexar até 10 portas clientes de taxa de 10G na sua porta rede que é configurada no protocolo OTU4. O OTN switch é um produto que possui uma matriz de crossconexão dinâmica permitindo que qualquer porta seja configurada como cliente ou rede, considerando as limitações da tecnologia. Nesse trabalho, será demonstrado a modelagem de um transponder de taxa de 100G com uma porta cliente e uma porta rede fixas.

A Figura 17 ilustra a representação de blocos do transponder modelado no ConfD. Inicialmente, a placa do transponder é desagregada pelo *circuit-pack* Transponder Base que agrega mais outros dois elementos *circuit-packs* do tipo transceptor nomeados Cliente 1 e Rede 1. Para fazer esse tipo de herança, é necessário utilizar o atributo de hereditariedade de *circuit-pack* denominado como *parent-circuit-pack*. Ainda na Figura 17, cada placa de circuito contém uma porta apenas, fornecendo todas as características e limitações ópticas associadas ao transponder. O modelo do equipamento, além disso, define a propriedade de caixa de formato (*cp-slots*) para decompor a configuração física de entrada dos transceptores de uma determinada porta. Por exemplo, o transceptor do tipo SPF e outro SPF+ tem o mesmo formato de caixa, assim como os transceptores do tipo QSFP e QSFP28.

Figura 17 – Representação em blocos de *circuit-packs* e portas do nó transponder modelado.



Fonte: Elaborada pelo autor. Baseada em [59].

O Quadro 24 exibe a resposta RPC reduzida do ConfD das informações de *circuit-pack* inseridas no simulador do transponder, onde as informações de inventário foram ocultadas da mensagem. É possível constatar que existem dois *circuit-packs* do tipo transceptor (*is-pluggable-optics*) nomeados (*circuit-pack-name*) como CP_Client e CP_Network e outro *circuit-*

pack do tipo padrão nomeado de *Transponder_Base*. O atributo de hereditariedade (*parent-circuit-pack*) está presente nos dois *circuit-packs* do tipo transceptor indicando o *Transponder_Base* como *circuit-pack* de base, origem. Além disso, cada elemento do tipo transceptor declara uma única porta, uma de qualidade (*port-qual*) de transponder cliente (*xpdr-client*) de nome (*port-name*) *CP_Client_1* e outra de transponder rede (*xpdr-network*) nomeada *CP_Network_1*. Portanto, este transponder contempla a estruturação apresentada na Figura 17 sendo composto por uma porta cliente e outra rede na caixa de formato CFP.

Quadro 24 – Resposta RPC do ConfD com as informações do módulo *circuit-pack* do transponder.

```
# netconf-console --rpc=query_circuits_packs.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <org-openroadm-device xmlns="http://org/openroadm/device">
      <circuit-packs>
        <circuit-pack-name>CP_Client</circuit-pack-name>
        <circuit-pack-type>CFP</circuit-pack-type>
        <is-pluggable-optics>true</is-pluggable-optics>
        <parent-circuit-pack>
          <circuit-pack-name>Transponder_Base</circuit-pack-name>
        </parent-circuit-pack>
        <ports>
          <port-name>CP_Client_1</port-name>
          <port-type>CFP-Grey</port-type>
          <port-qual>xpdr-client</port-qual>
          <port-wavelength-type>wavelength</port-wavelength-type>
          <port-direction>bidirectional</port-direction>
        </ports>
      </circuit-packs>
      <circuit-packs>
        <circuit-pack-name>CP_Network</circuit-pack-name>
        <circuit-pack-type>CFP</circuit-pack-type>
        <is-pluggable-optics>true</is-pluggable-optics>
        <parent-circuit-pack>
          <circuit-pack-name>Transponder_Base</circuit-pack-name>
        </parent-circuit-pack>
        <ports>
          <port-name>CP_Network_1</port-name>
          <port-type>CFP-DCO</port-type>
          <port-qual>xpdr-network</port-qual>
          <port-wavelength-type>wavelength</port-wavelength-type>
          <port-direction>bidirectional</port-direction>
        </circuit-packs>
      <circuit-packs>
        <circuit-pack-name>Transponder_Base</circuit-pack-name>
        <circuit-pack-type>Transponder</circuit-pack-type>
        <is-pluggable-optics>false</is-pluggable-optics>
      </circuit-packs>
    </org-openroadm-device>
  </data>
</rpc-reply>
```

As características ópticas e limitações da tecnologia do nó transponder são modeladas pelos módulos de perfil de hierarquia de multiplexação ODU (*otn-odu-mux-hierarchy-profile*), perfil de multiplexação (*muxp-profile*), perfil de capacidade OTN (*otn-capability-profile*) e perfil de capacidade de canais ópticos (*mc-capability-profile*). Tais perfis foram instanciados no ConfD para desagregar as capacidades do transponder simulado e a mensagem RPC desses módulos está exposta no Quadro 25. Existem ainda outros módulos de restrições de portas para desagregar funções de tipos especiais de transponders como o OTN switch e o muxponder que não serão detalhados nesse trabalho.

Quadro 25 – Resposta RPC adaptada do ConfD com as informações de capacidades do transponder.

```
# netconf-console --rpc=query_odu_mux_profile.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <org-openroadm-device xmlns="http://org/openroadm/device">
      <otn-odu-mux-hierarchy-profile>
        <profile-name>TRANSPONDER-ODU-MUX-HIERARCHY</profile-name>
        <mux-capability>
          <stage-number>1</stage-number>
          <ho-odu-type>ODU4</ho-odu-type>
          <ho-odu-payload-type>21</ho-odu-payload-type>
          <supported-lo-odu-type>ODU4</supported-lo-odu-type>
          <lo-odu-proactive-DMp>false</lo-odu-proactive-DMp>
          <lo-odu-tcm-capable>false</lo-odu-tcm-capable>
        </mux-capability>
      </otn-odu-mux-hierarchy-profile>
      <muxp-profile>
        <profile-name>TRANSPONDER-TRIBUTARY-ODU4-OTU4</profile-name>
        <odtu-type>ODTU4.ts</odtu-type>
        <network-odu-rate>ODU4</network-odu-rate>
        <network-ho-odu-trib-port-number>1</network-ho-odu-trib-port-number>
        <network-ho-odu-trib-slots>1</network-ho-odu-trib-slots>
        [...]
        <network-ho-odu-trib-slots>80</network-ho-odu-trib-slots>
      </muxp-profile>
      <otn-capability-profile>
        <profile-name>TRANSPONDER-OTN-CAPABILITIES</profile-name>
        <if-protection-capability>odu-one-plus-one</if-protection-capability>
        <proactive-DMp>false</proactive-DMp>
        <tcm-capable>false</tcm-capable>
        <opu-payload-type-mapping>21</opu-payload-type-mapping>
      </otn-capability-profile>
      <mc-capability-profile>
        <profile-name>100G-CFP-DCO</profile-name>
        <center-freq-granularity>50.0</center-freq-granularity>
        <min-edge-freq>191.25</min-edge-freq>
      </mc-capability-profile>
    </org-openroadm-device>
  </data>
</rpc-reply>
```

```

<max-edge-freq>196.25</max-edge-freq>
<slot-width-granularity>50.0</slot-width-granularity>
<min-slots>1</min-slots>
<max-slots>1</max-slots>
</mc-capability-profile>

</org-openroadm-device>
</data>
</rpc-reply>

```

Fonte: Elaborada pelo autor.

O perfil de hierarquia de multiplexação ODU apresentado no Quadro 25 parte A tem a responsabilidade de informar quantos estágio de multiplexação (*stage-number*) possíveis, os tipos suportados de entidades ODU de maior (*ho-odu-type*) e menor (*supported-lo-odu-type*) ordens, e finalmente, informações de funcionalidade de monitoração OTN (*lo-odu-tcm-capable*) e medição de atraso do sinal (*lo-odu-proactive-DMp*). Por meio da interpretação da resposta RPC, o transponder simulado pode multiplexar apenas um nível do protocolo OTN na ordem alta e baixa de ODU4, equivalente as taxas clientes de 100 Gpbs. Além disso, tal transponder não oferece suporte nas aplicações de monitoração e medição de atraso.

O Quadro 25 parte B ilustra os dados fictícios do perfil de multiplexação inseridos no ConfD. A função desse perfil é informar como é a estrutura de multiplexação do quadro OPU, indicando o conteúdo de cada posição do tributário. Com essa informação, é possível identificar quais são e onde estão os tributários de baixa ordem dentro de um quadro OTN. A mensagem RPC informa que o transponder simulado tem taxa de rede (*network-odu-rate*) igual a ODU4 e que a estrutura de multiplexação é do tipo (*odtu-type*) ODTU4.ts contendo somente uma porta de tributário (*network-ho-odu-trib-port-number*) distribuídos em 80 posições de tributários (*network-ho-odu-trib-slots*). Em outros termos, tal equipamento não suporta a multiplexação de taxas de mais baixas ordem como ODU3 e ODU2, somente ODU4.

O perfil de capacidade OTN é exposto no Quadro 25 parte C e representa um conjunto de funcionalidades suportadas das interfaces como o tipo de proteção de tráfego (*if-protection-capability*), tipo do mapeamento dos dados transportados (*opu-payload-type-mapping*), suporte de monitoração OTN específica (*tcm-capable*) e dentre outros atributos. Na resposta RPC, é possível interpretar que as interfaces do equipamento não suportam a monitoração OTN específica e o tipo de mapeamento de dados é 21, valor definido na recomendação G.709 [12]. Outro fator importante é que o transponder simulado suporta o tipo de proteção ODU 1+1 (*odu-one-plus-one*) significando que pode proteger o tráfego em suas portas redes por dois caminhos possíveis.

Por último, o Quadro 25 parte D exibe o perfil de capacidade de canais ópticos, já apresentado para o nó ROADM, indicando a tecnologia de espaçamento e grade de canais

suportada pelo transponder. A mensagem RPC indica que tal perfil é capaz de configurar canais DWDM dentro da faixa de canais C12 (*min-edge-freq*) à C62 (*max-edge-freq*). O espaçamento entre canais (*slot-width-granularity*) é fixo de 50 GHz, não permitindo a tecnologia flexível de canais pois apenas suporta uma posição de frequência (*min-slots*, *max-slots*) dentro do canal. A utilidade desse perfil é caracterizar a tecnologia de espaçamento na porta rede do transponder simulado.

Todos os perfis ópticos apresentados são usados, portanto, para desagregar as características e limitações da tecnologia e são associados para cada porta do transponder. Diferentes tipos de perfis são necessários para modelar alguma restrição de configuração ou modo de operação do transponder. De certo modo, isso permite que os fabricantes consigam abstrair qualquer funcionalidade ou arquitetura de operação de seus transponders. Logo, os modelos OpenROADM são capazes de desagregar os equipamentos ópticos em níveis detalhados e realísticos.

O Quadro 26 apresenta a mensagem RPC adaptada do ConfD para o retorno das capacidades configuradas das portas de tipo transceptor do transponder simulado. A primeira porta de nome CP_Client_1, servindo como uma porta cliente, implementa as camadas OTU4-ODU4 e tem os seguintes perfis configurados: hierarquia de multiplexação ODU (*otn-odu-mux-hierarchy-profile-name*), capacidade OTN (*otn-capability-profile-name*) e multiplexação (*muxp-profile*). O perfil de multiplexação é agrupado no grupo de restrições de multiplexação e mapeamento de cliente (*mpdr-client-restriction*) com as informações da porta rede de destino. No final da mensagem, a porta rede de nome CP_Network_1 que suporta as camadas OCH-OTU4-ODU4 tem os perfis aplicados: capacidade OTN e hierarquia de multiplexação ODU.

Quadro 26 – Resposta RPC adaptada do ConfD com as informações de capacidade das portas.

```
# netconf-console --rpc=query_port_capabilities.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <org-openroadm-device xmlns="http://org/openroadm/device">
      <ports>
        <port-name>CP_Client_1</port-name>
        <port-capabilities xmlns="http://org/openroadm/port-capability">
          <supported-interface-capability>
            <if-cap-type>if-OTU4-ODU4</if-cap-type>
            <otn-capability>
              <otn-capability-profile-name>TRANSPONDER-OTN-CAPABILITIES</otn-
capability-profile-name>
              <mpdr-client-restriction>
                <network-ho-odu-circuit-pack-name>CP_Network</network-ho-odu-
circuit-pack-name>
                <network-ho-odu-port-name>CP_Network_1</network-ho-odu-port-name>
```

```

name>
    <muxp-profile-name>TRANSPONDER-TRIBUTARY-ODU4-OTU4</muxp-profile-
name>
    </mpdr-client-restriction>
    <otn-odu-mux-hierarchy-profile-name>TRANSPONDER-ODU-MUX-
HIERARCHY</otn-odu-mux-hierarchy-profile-name>
    </otn-capability>
    </supported-interface-capability>
    </port-capabilities>
</ports>
<ports>
    <port-name>CP_Network_1</port-name>
    <port-capabilities xmlns="http://org/openroadm/port-capability">
    <supported-interface-capability>
    <if-cap-type>if-OCH-OTU4-ODU4</if-cap-type>
    <otn-capability>
    <otn-capability-profile-name>TRANSPONDER-OTN-CAPABILITIES</otn-
capability-profile-name>
    <otn-odu-mux-hierarchy-profile-name>TRANSPONDER-ODU-MUX-
HIERARCHY</otn-odu-mux-hierarchy-profile-name>
    </otn-capability>
    </supported-interface-capability>
    </port-capabilities>
    </ports>
</data>
</rpc-reply>

```

Fonte: Elaborada pelo autor.

Foi mencionado, portanto, que o modelo do equipamento permite a configuração das capacidades nas portas. Além disso, as portas de transponders são geralmente projetadas para funcionarem com transceptores externos de certos tipos. De certo modo, o modelo permite que as caixas de formato sejam abstraídas para indicarem quais são os possíveis tipos de transceptores compatíveis. Por exemplo, pode existir uma porta que tenha uma caixa de formato do tipo SFP e SFP+. O transceptor SFP geralmente funciona com taxas abaixo de 2.5 Gbps enquanto o SFP+ opera em taxas de 10 Gbps. Portanto, os dados da caixa de formato é uma lista que informa as capacidades e limitações de todos transceptores compatíveis de uma porta.

A propriedade de caixa de formato (*cp-slots*) dos *circuit-packs* do tipo transceptor são ilustrados no Quadro 27. A primeira caixa de formato pertence ao *circuit-pack* de nome (*provisioned-circuit-pack*) CP_Client servindo de porta cliente enquanto a segunda opera como porta rede de nome CP_Network. De acordo com a documentação [59], a caixa de formato de transceptor é declarada no modelo pelo atributo (*slot-type*) com o valor configurado para *pluggable-optics-holder*. Após essa configuração, é possível indicar quais são os tipos de transceptores aceitos pelo atributo *supported-pluggable-id-type*. Na mensagem RPC, o valor CFP é configurado para ambas as caixas de formato. Por fim, cada tipo de transceptor suportado recebe um subconjunto das capacidades suportadas pela porta configurada.

```
# netconf-console --rpc=query_cp_slots.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <org-openroadm-device xmlns="http://org/openroadm/device">
      <cp-slots>
        <slot-name>CFP-100g</slot-name>
        <provisioned-circuit-pack>CP_Client</provisioned-circuit-pack>
        <slot-type>pluggable-optics-holder</slot-type>
        <pluggable-optics-holder-capability>
          <supported-circuit-pack>
            <supported-pluggable-id-type>CFP</supported-pluggable-id-type>
          </supported-circuit-pack>
          <ports>
            <port-name>1</port-name>
            <port-capabilities>
              [Perfis de capacidade configurada da porta cliente.]
            </port-capabilities>
          </ports>
        </supported-circuit-pack>
      </pluggable-optics-holder-capability>
    </cp-slots>
    <cp-slots>
      <slot-name>CFP-100g</slot-name>
      <provisioned-circuit-pack>CP_Network</provisioned-circuit-pack>
      <slot-type>pluggable-optics-holder</slot-type>
      <pluggable-optics-holder-capability>
        <supported-circuit-pack>
          <supported-pluggable-id-type>CFP</supported-pluggable-id-type>
        </supported-circuit-pack>
        <ports>
          <port-name>1</port-name>
          <port-capabilities>
            [Perfis de capacidade configurada da porta rede.]
          </port-capabilities>
        </ports>
      </supported-circuit-pack>
    </pluggable-optics-holder-capability>
  </cp-slots>
</org-openroadm-device>
</data>
</rpc-reply>
```

Fonte: Elaborada pelo autor.

A documentação do OpenROADM [59] aponta que o transponder deve informar ao controlador SDN a conectividade entre as portas cliente e rede através do módulo de matriz de crossconexão (*odu-switching-pools*). A partir de tal módulo, é possível modelar as restrições entre as portas clientes e redes como limitações de configuração de protocolos devido à falta de banda da matriz de comutação e qual a porta rede de destino para um conjunto de portas clientes. Portanto, a configuração da matriz de crossconexão para o transponder simulado de duas portas é simples em comparação com outros tipos de transponder como OTN switch e muxponder.

O Quadro 28 ilustra a mensagem RPC das configurações da matriz de crossconexão criada no ConfD para o transponder simulado. Por meio da interpretação dos dados, existe apenas uma matriz de número (*switching-pool-number*) igual a 1 do tipo (*switching-pool-type*) bloqueante. Isso significa que essa matriz bloqueia qualquer conexão entre as portas do transponder exceto as que estão presente na lista não bloqueante (*non-blocking-list*). A lista da resposta RPC é composta por 1 bloco (*nbl-number*) possuindo duas portas de nome CP_Client_1 e CP_Network_1, as quais não compartilham nenhuma banda da matriz de conexão por causa do valor 0 nos atributos *interconnect-bandwidth* e *interconnect-bandwidth-unit*. Portanto, isso indica que somente a porta CP_Client_1 pode ser conectada com a porta CP_Network_1.

Quadro 28 – Resposta RPC do ConfD com as informações da matriz de crossconexão.

```
# netconf-console --rpc=query_switching_pools.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:iETF:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <org-openroadm-device xmlns="http://org/openroadm/device">
      <odu-switching-pools>
        <switching-pool-number>1</switching-pool-number>
        <switching-pool-type>blocking</switching-pool-type>
        <non-blocking-list>
          <nbl-number>1</nbl-number>
          <interconnect-bandwidth-unit>0</interconnect-bandwidth-unit>
          <interconnect-bandwidth>0</interconnect-bandwidth>
          <port-list>
            <circuit-pack-name>CP_Client</circuit-pack-name>
            <port-name>CP_Client_1</port-name>
          </port-list>
          <port-list>
            <circuit-pack-name>CP_Network</circuit-pack-name>
            <port-name>CP_Network_1</port-name>
          </port-list>
        </non-blocking-list>
      </odu-switching-pools>
    </org-openroadm-device>
  </data>
</rpc-reply>
```

Fonte: Elaborada pelo autor.

O último módulo relevante para finalizar a construção do simulador do transponder é implementar o módulo do transponder (*xponder*). O Quadro 29 apresenta a mensagem RPC com as informações inseridas no ConfD para esse módulo. O nó transponder é composto por apenas um único transponder (*xpdr-number*) do tipo transponder (*xpdr-type*). Outros tipos definidos no modelo são muxponder, regenerador e OTN switch. Ainda na resposta RPC, é possível identificar que existem duas portas (*xpdr-port*) nesse nó transponder de nomes CP_Client_1 e CP_Network_1. Além disso, existem duas caixas de formato (*xpdr-pluggable-optics-holder*) das

duas portas desse transponder correspondendo ao tranceptor do tipo CFP definido no módulo *cp-slots* de nome CFP-100g. Desse modo, o nó transponder está pronto para uso e simulação.

Quadro 29 – Resposta RPC do ConfD com as informações do módulo transponder.

```
# netconf-console --rpc=query_transponder.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <org-openroadm-device xmlns="http://org/openroadm/device">
      <xponder>
        <xpdr-number>1</xpdr-number>
        <xpdr-type>tpdr</xpdr-type>
        <xpdr-port>
          <index>1</index>
          <circuit-pack-name>CP_Client</circuit-pack-name>
          <port-name>CP_Client_1</port-name>
        </xpdr-port>
        <xpdr-port>
          <index>2</index>
          <circuit-pack-name>CP_Network</circuit-pack-name>
          <port-name>CP_Network_1</port-name>
        </xpdr-port>
        <xpdr-pluggable-optics-holder>
          <index>1</index>
          <circuit-pack-name>CP_Client</circuit-pack-name>
          <cp-slot-name>CFP-100g</cp-slot-name>
        </xpdr-pluggable-optics-holder>
        <xpdr-pluggable-optics-holder>
          <index>2</index>
          <circuit-pack-name>CP_Network</circuit-pack-name>
          <cp-slot-name>CFP-100g</cp-slot-name>
        </xpdr-pluggable-optics-holder>
      </xponder>
    </org-openroadm-device>
  </data>
</rpc-reply>
```

Fonte: Elaborada pelo autor.

Nesse capítulo foi apresentado quais são os principais módulos do modelo de equipamentos do OpenROADM para a formação de um nó ROADM e um nó transponder. Com isso, foi possível compreender qual a função específica de cada módulo e como são suas interações entre si para desagregar os equipamentos DWDM da rede de transporte. Recorda-se que a inserção de dados fictícios no ConfD foi executada pelo autor por meio da interface CLI em diferentes máquinas virtuais.

Capítulo 5

Simulação e Resultados

O principal objetivo da simulação é demonstrar como o controlador SDN deveria provisionar serviços usando os elementos de rede ópticas desagregados do modelo do equipamento de acordo com a recomendação publicada do OpenROADM MSA [59]. Portanto, será apresentado como os equipamentos DWDM desagregados são controlados e configurados. Além disso, a simulação incorpora o nível de desagregação definido na segunda fase do projeto OTDN, o qual explora as funcionalidades avançadas dos equipamentos DWDM: ROADMs e transponders.

5.1. Simulação

A simulação do controle das placas para provisionar um serviço foi estimulada usando as máquinas virtuais com ConfD importando o modelo do equipamento OpenROADM. Cada máquina virtual tem um conjunto de dados fictícios distintos caracterizando um nó diferente da rede. A simulação consiste em demonstrar as configurações que o controlador SDN deve criar em cada um dos nós para provisionar um serviço em uma direção apenas. Para começar, a próxima subseção apresentará a rede utilizada pra essa simulação.

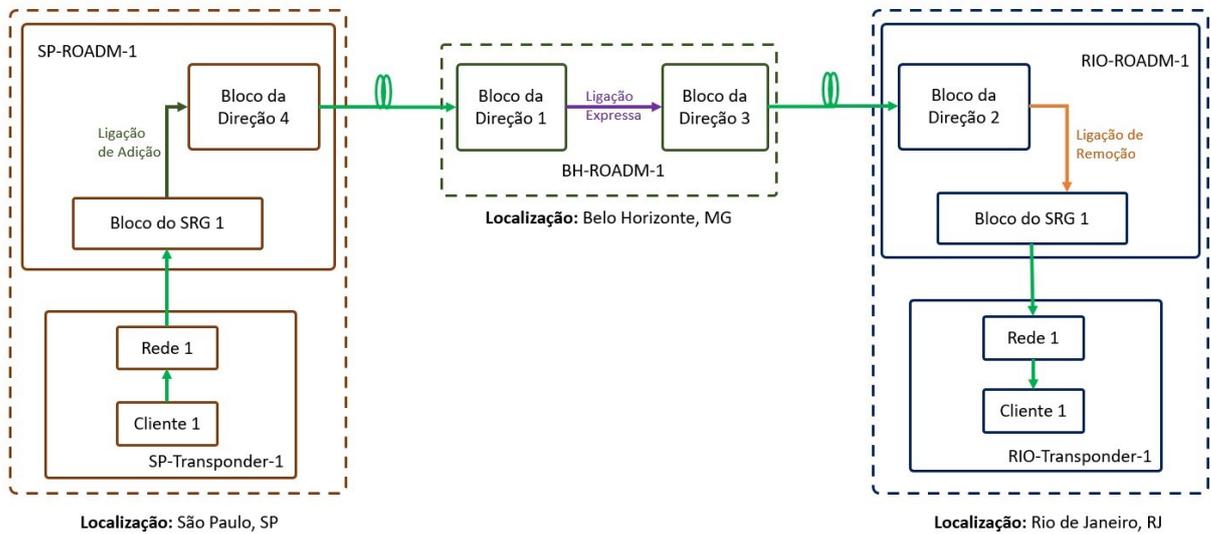
5.1.1. Ambiente da Simulação

A Figura 18 ilustra o ambiente de simulação contendo três estações de telecomunicações em localizações diferentes. Cada estação é composta por equipamentos DWDM que foram desagregados em nós. A estação de São Paulo consiste de um nó ROADM nomeado como SP-ROADM-1 e um nó transponder SP-Transponder-1. O segundo nó ROADM RIO-ROADM-1 e o nó transponder RIO-Transponder-1 formam a estação do Rio de Janeiro. A última estação é a estação de Belo Horizonte que contém apenas um nó ROADM nomeado como BH-ROADM-1. As linhas de transmissão ópticas conectam as estações de Rio de Janeiro e São Paulo na estação de Belo Horizonte.

A simulação do provisionamento do serviço tem como origem a porta Cliente 1 do nó SP-Transponder-1 e destino a porta Cliente 1 do nó RIO-Transponder-1. Existe apenas uma trajetória possível pra estabelecer o tráfego pra tal serviço, passando pela estação de Belo Horizonte. Cada máquina virtual executa um ConfD diferente que representa um nó da topologia

de rede. Na simulação não será apresentado nenhum dado de inventário ou de configuração física das estações pois alguns exemplos já foram expostos conforme apresentado nas seções 4.2 e 4.4.

Figura 18 – Ambiente de simulação.



Fonte: Elaborada pelo autor, baseado em [59].

De acordo com a recomendação [59], o controlador SDN deve seguir uma sequência de passos em cada nó da topologia para provisionar um serviço entre os nós transponders. Segue a lista de passos aplicada na topologia da simulação.

1. **SP-Transponder-1:** Criar as interfaces nas portas e configurar o canal para transmissão.
2. **SP-ROADM-1:** Instanciar a ligação de adição do bloco SRG para o bloco da direção.
3. **BH-ROADM-1:** Configurar a ligação expressa entre os blocos de direções.
4. **RIO-ROADM-1:** Gerar uma ligação de remoção do bloco da direção para o bloco SRG.
5. **RIO-Transponder-1:** Criar as interfaces nas portas e recuperar o sinal cliente do canal de transmissão.

Cada subseção subsequente irá expor a sequência de configurações para cada nó da rede que o controlador SDN executa durante o provisionamento de um serviço.

5.1.2. Nó Transponder de Transmissão

O nó transponder de transmissão é representado pelo nó SP-Transponder-1, um transponder com uma porta cliente compatível com protocolo OTU4 (taxas de 100 Gbps) e com uma porta rede que suporta a configuração para qualquer canal da banda C. O controlador SDN configura as interfaces da porta cliente e da porta rede para transmitir o sinal cliente multiplexado no protocolo OTN em um canal disponível. Para a simulação, o sinal será transmitido no canal C27 que consiste no valor de frequência de 192.7 THz ou 1555.75 nm aproximadamente. Esse

canal foi escolhido aleatoriamente dentro dos canais suportados pelo transponder de acordo com seu perfil de capacidade de canais ópticos apresentado na seção 4.11.

A primeira interface configurada é a interface OCh na porta rede do transponder. O Quadro 30 apresenta tal interface criada pelo autor para simular a ação do controlador SDN. Primeiramente, o nome da interface (*name*) definido é XPDR1-NETWORK1-TXRX-192.7, seguindo um padrão sugerido na documentação [59]. Pelo nome é possível deduzir que a interface pertence ao transponder de número 1 (XPDR1) da porta rede 1 (NETWORK1) bidirecional (TXRX) que está configurada no canal de frequência 192.7 THz. Porém, tais informações também estão declaradas em seus respectivos atributos. Além disso, o tipo da interface (*type*) é o OCh (*opticalChannel*) de taxa de linha (*rate*) de 100Gpbs no formato de modulação (*modulation-format*) DP-QPSK.

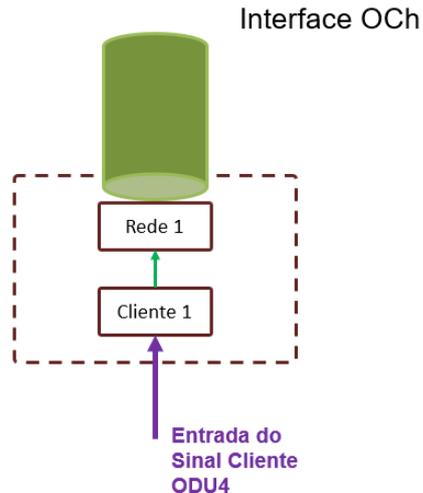
Quadro 30 – Resposta RPC adaptada do ConfD para a interface OCh da porta rede.

```
# netconf-console --rpc=query_och_transponder.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <org-openroadm-device xmlns="http://org/openroadm/device">
      <interface>
        <name>XPDR1-NETWORK1-TXRX-192.7</name>
        <type>opticalChannel</type>
        <administrative-state>inService</administrative-state>
        <supporting-circuit-pack-name>CP_Network</supporting-circuit-pack-name>
        <supporting-port>CP_Network_1</supporting-port>
        <och xmlns="http://org/openroadm/optical-channel-interfaces">
          <rate>R100G</rate>
          <frequency>192.7</frequency>
          <modulation-format>dp-qpsk</modulation-format>
        </och>
      </interface>
    </org-openroadm-device>
  </data>
</rpc-reply>
```

Fonte: Elaborada pelo autor.

A Figura 19 ilustra a representação em forma de tubo da interface OCh instanciada na porta rede 1, assim como, a entrada do sinal cliente ODU4 na porta cliente 1 do nó transponder SP-Transponder-1.

Figura 19 – Interface OCh da porta rede do nó transponder.



Fonte: Elaborada pelo autor.

O controlador SDN deve modelar uma interface OTU para servir a camada OTU do protocolo OTN. Portanto, tal interface foi criada no nó transponder e está apresentada no Quadro 31 como nome OTU4-XPDR1-NETWORK1-TXRX. As informações de *circuit-pack* e portas são as mesmas da interface OCh enquanto o tipo da interface é OTU (*otnOtu*). O atributo de lista de interfaces de suporte (*supporting-interface-list*) tem o valor do nome da interface OCh previamente criada, indicando que a camada OCh está acima da camada OTU.

O modelo OpenROADM oferece suporte das funcionalidades de monitoração e operação da camada OTU através do atributo de interface *otu*, exibido na mensagem RPC do Quadro 31. Por isso, é possível identificar que a taxa de linha (*rate*) da interface é OTU4 com suporte ao código corretor de erros (*fec*) é do tipo SDFEC. Além disso, a funcionalidade de monitoração dos identificadores de ponto de acesso de origem (SAPI, *source access point identifiers*) e destino (DAPI, *destination access point identifiers*) são representados pelos atributos *tx-sapi*, *tx-dapi*, *expected-sapi*, *expected-dapi*, *tim-act-enabled* e *tim-detect-mode*. O principal objetivo de tal funcionalidade OTN é auxiliar se as conexões físicas entre os nós ROADM e transponders de São Paulo estão corretamente instaladas. Os dois últimos atributos da mensagem RPC, *degm-intervals* e *degthr-percentage*, configuram o transponder para subir um alarme de degradação do sinal caso os limites sejam ultrapassados.

Quadro 31 – Resposta RPC adaptada do ConfD para a interface OTU da porta rede.

```
# netconf-console --rpc=query_otu_transponder.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <org-openroadm-device xmlns="http://org/openroadm/device">
      <interface>
        <name>OTU4-XPDR1-NETWORK1-TXRX</name>
```

```

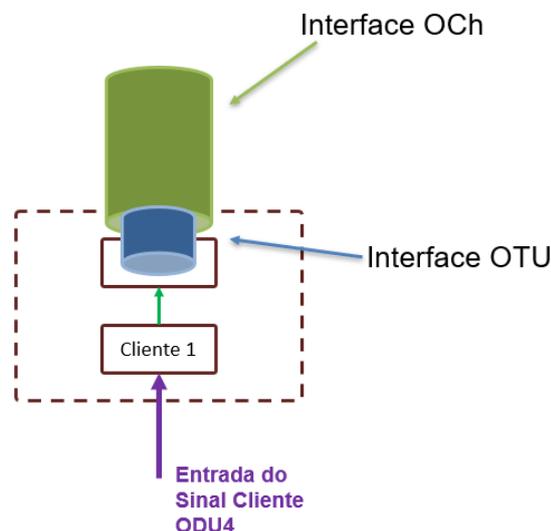
<type>otnOtu</type>
<administrative-state>inService</administrative-state>
<supporting-circuit-pack-name>CP_Network</supporting-circuit-pack-name>
<supporting-port>CP_Network_1</supporting-port>
<supporting-interface-list>XPDR1-NETWORK1-TXRX-192.7</supporting-interface-
list>
<otu xmlns="http://org/openroadm/otn-otu-interfaces">
  <rate>OTU4</rate>
  <fec>sdfec</fec>
  <tx-sapi>de-SP-Transp-1</tx-sapi>
  <tx-dapi>para-SP-ROADM-1</tx-dapi>
  <expected-sapi>de-SP-ROADM-1</expected-sapi>
  <expected-dapi>para-SP-Trans-1</expected-dapi>
  <tim-act-enabled>>false</tim-act-enabled>
  <tim-detect-mode>SAPI-and-DAPI</tim-detect-mode>
  <degm-intervals>2</degm-intervals>
  <degthr-percentage>100</degthr-percentage>
</otu>
</interface>
</org-openroadm-device>
</data>
</rpc-reply>

```

Fonte: Elaborada pelo autor.

A Figura 20 ilustra a representação em forma de tubo da interface OTU instanciada dentro da interface OCh previamente criada na porta rede 1, de certa forma, mostrando a correlação entre as interfaces ópticas OTU e OCh.

Figura 20 – Interface OTU da porta rede do nó transponder.



Fonte: Elaborada pelo autor.

A próxima interface instanciada é da camada ODU para servir a camada superior OTU. O Quadro 32 ilustra a interface criada no ConfD de nome ODU4-XPDR1-NETWORK1-TXRX e tipo ODU (*otnOdu*). Assim como as outras interfaces OTU e OCh, tal interface é aplicada na mesma porta rede. Porém o atributo da lista de interfaces de suporte (*supporting-interface-list*) tem

o valor do nome da interface OTU, conseqüentemente, associando a camada ODU inclusa na camada OTU.

As funções específicas do protocolo ODU são suportadas pelo atributo *odu* onde é possível ver na mensagem RPC que a interface criada tem a taxa de linha (*rate*) ODU4, o modo da operação (*monitoring-mode*) não terminada e a função ODU (*odu-function*) do tipo ODU-CTP. Em outros termos, esse nó transponder recebe um sinal cliente ODU4 que é transportado na sua porta rede de forma transparente, sem a intrusão e monitoração do protocolo. Tal modo de configuração não permite que o quadro OPU, incorporado na camada ODU, seja inspecionado pelo transponder. Por isso, as funcionalidades ofertadas pelo modelo OpenROADM para as camadas ODU e OPU não são utilizadas na simulação.

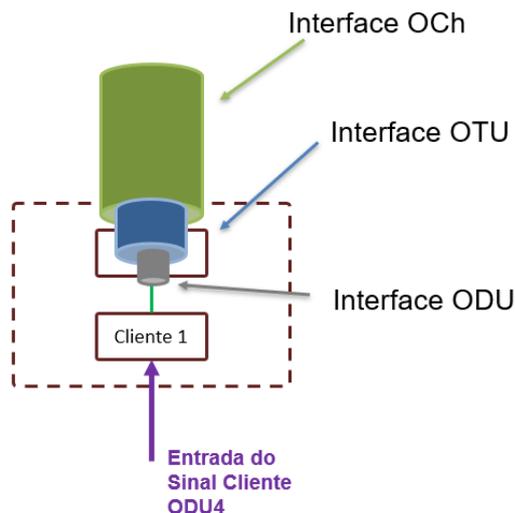
Quadro 32 – Resposta RPC adaptada do ConfD para a interface ODU da porta rede.

```
# netconf-console --rpc=query_odu_transponder.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <org-openroadm-device xmlns="http://org/openroadm/device">
      <interface>
        <name>ODU4-XPDR1-NETWORK1-TXRX</name>
        <type>otnOdu</type>
        <administrative-state>inService</administrative-state>
        <supporting-circuit-pack-name>CP_Network</supporting-circuit-pack-name>
        <supporting-port>CP_Network_1</supporting-port>
        <supporting-interface-list>OTU4-XPDR1-NETWORK1-TXRX</supporting-interface-
list>
        <odu xmlns="http://org/openroadm/otn-odu-interfaces">
          <rate>ODU4</rate>
          <odu-function>ODU-CTP</odu-function>
          <monitoring-mode>not-terminated</monitoring-mode>
        </odu>
      </interface>
    </org-openroadm-device>
  </data>
</rpc-reply>
```

Fonte: Elaborada pelo autor.

A Figura 21 ilustra a representação em forma de tubo da interface ODU instanciada dentro da interface OTU previamente criada na porta rede 1. Assim, é mostrado a correlação entre as interfaces ópticas ODU, OTU e OCh.

Figura 21 – Interface ODU da porta rede do nó transponder.



Fonte: Elaborada pelo autor.

Após a criação das interfaces da porta rede, agora o controlador SDN deve configurar a interface da porta cliente. Nessa simulação, assume-se que a porta cliente recebe um sinal cliente OTN do tipo ODU4. A camada OTU4 que é terminada na porta cliente não será contabilizada nas interfaces do simulador, somente foi criada a interface ODU4. O Quadro 33 ilustra a resposta RPC com informações da interface ODU de nome ODU4-XPDR1-CLIENT1-TXRX de taxa de linha ODU4. Além disso, os atributos do nome do *circuit-pack* (*supporting-circuit-pack-name*) e da porta (*supporting-port*) fazem referência à porta cliente que está diretamente conectada na porta rede de acordo com a matriz de conexão do nó transponder. Similarmente a configuração da interface rede, não existe monitoração ou intrusão no sinal ODU4 cliente indica no atributo *odu*.

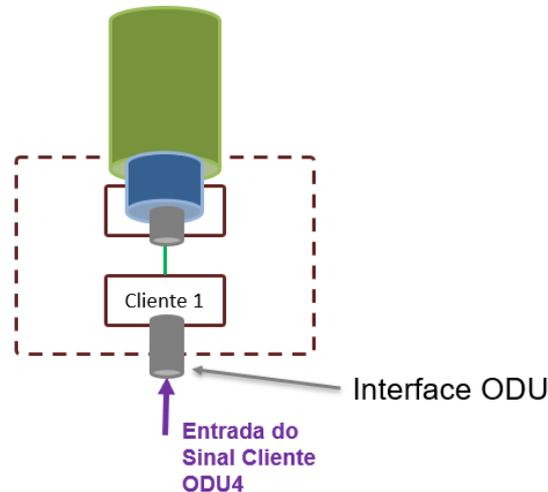
Quadro 33 – Resposta RPC adaptada do ConfD para a interface ODU da porta cliente.

```
# netconf-console --rpc=query_odu_client_transponder.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <data>
      <org-openroadm-device xmlns="http://org/openroadm/device">
        <interface>
          <name>ODU4-XPDR1-CLIENT1-TXRX</name>
          <type>otnOdu</type>
          <administrative-state>inService</administrative-state>
          <supporting-circuit-pack-name>CP_Client</supporting-circuit-pack-name>
          <supporting-port>CP_Client_1</supporting-port>
          <odu xmlns="http://org/openroadm/otn-odu-interfaces">
            <rate>ODU4</rate>
            <odu-function>ODU-CTP</odu-function>
            <monitoring-mode>not-terminated</monitoring-mode>
          </odu>
        </interface>
      </org-openroadm-device>
    </data>
  </rpc-reply>
```

Fonte: Elaborada pelo autor.

A Figura 22 ilustra a representação em forma de tubo da interface criada na porta cliente 1 do nó transponder, que recebe o sinal cliente ODU4.

Figura 22 – Interface ODU da porta cliente do nó transponder.



Fonte: Elaborada pelo autor.

Após a criação de todas as interfaces das camadas OTN no nó SP-Transponder-1, o controlador SDN precisa criar uma crossconexão entre as interfaces de origem e destino para estabelecer o tráfego efetivo da porta cliente para a porta rede. O modelo OpenROADM reserva o módulo *odu-connection* para tal finalidade e está ilustrado no Quadro 34. É possível observar que o nome da conexão segue um padrão com os nomes das interfaces de origem (*src-if*) ODU4-XPDR1-CLIENT1-TXRX e de destino (*dst-if*) ODU4-XPDR1-NETWORK1-TXRX. De certa forma, essa informação confirma o circuito estabelecido neste nó SP-Transponder-1 no provisionamento do serviço fim-a-fim.

Quadro 34 – Resposta RPC do ConfD da crossconexão de interfaces do nó SP-Transponder-1.

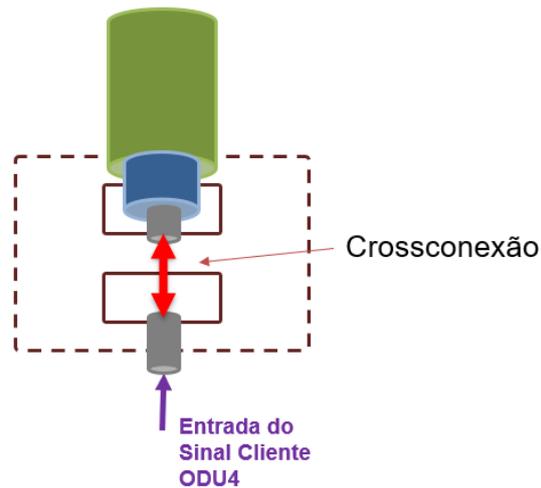
```
# netconf-console --rpc=query_odu_connections.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <org-openroadm-device xmlns="http://org/openroadm/device">
      <odu-connection>
        <connection-name>ODU4-XPDR1-CLIENT1-TXRX-to-ODU4-XPDR1-NETWORK1-
TXRX</connection-name>
        <source>
          <src-if>ODU4-XPDR1-CLIENT1-TXRX</src-if>
        </source>
        <destination>
          <dst-if>ODU4-XPDR1-NETWORK1-TXRX</dst-if>
        </destination>
      </odu-connection>
    </org-openroadm-device>
  </data>
</rpc-reply>
```

```
</data>  
</rpc-reply>
```

Fonte: Elaborada pelo autor.

A Figura 23 ilustra a crossconexão entre os tubos ODU4 do nó transponder, permitindo portanto, a passagem de tráfego da porta cliente para a porta rede.

Figura 23 – Crossconexão entre as interfaces ODUs do nó transponder.



Fonte: Elaborada pelo autor.

5.1.3. Nó ROADM da Ligação de Adição

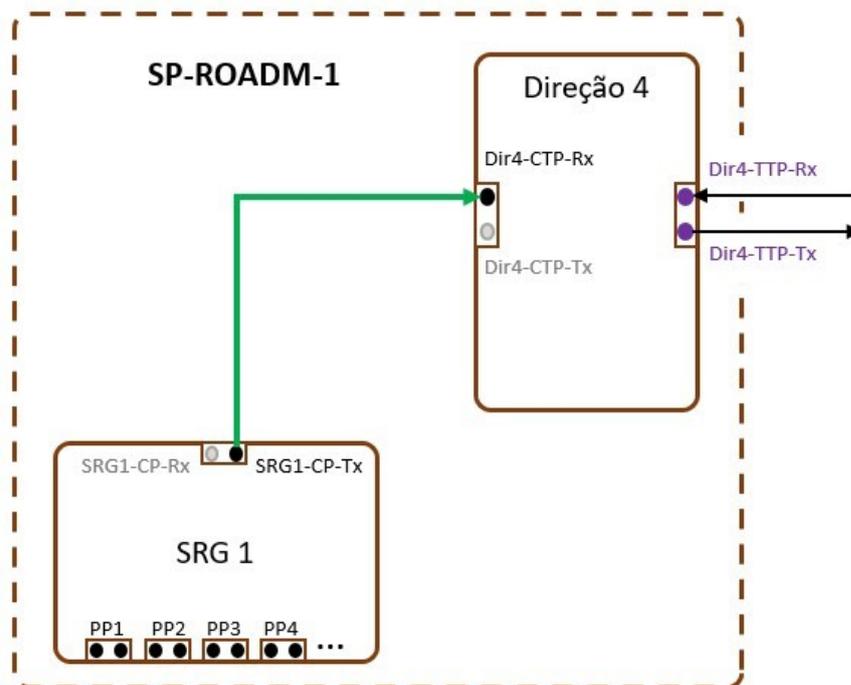
O próximo nó alvo para o controlador SDN configurar as interfaces é o nó SP-ROADM-1 localizado em São Paulo. A função deste nó é basicamente receber o sinal transmitido pelo transponder do canal C27 e adicioná-lo na malha de transmissão da rede, este agregando todos outros canais transmitidos por outros transponders. Reforçando que a ligação de adição de um nó ROADM representa a fibra óptica, ou um enlace físico, que conecta o bloco SRG ao bloco de direção. Antes de criar a ligação de adição, é necessário definir as ligações lógicas nos *circuit-packs* do nó ROADM.

O modelo da rede abstrai as ligações físicas em ligações lógicas como o ponto de terminação de conexão (CTP, *Connection Termination Point*), ponto de conexão (CP, *Connection Point*), ponto de terminação de trilha (TTP, *Trail Termination Point*) e ponto de pares de portas (PP, *Port Pairs*). Desse modo, o controlador SDN não precisa processar as configurações físicas de todos os nós da rede pois o modelo da rede providencia as ligações lógicas, que por sua vez, oferecem informações suficientes para criar serviços de uma forma agnóstica ao modelo de equipamento específico de cada fabricante.

A Figura 24 ilustra as terminações lógicas do nó SP-ROADM-1 localizado na estação de São Paulo. Nesse nó, existe apenas uma ligação de adição que conectam o bloco SRG 1 ao bloco da direção 4. De acordo com a recomendação [56], as terminações lógicas de uma ligação de adição são nomeadas como CTP para o bloco de direção e CP para o bloco SRG. Por exemplo, as portas de fronteiras Rx e Tx do bloco de direção 4 são intituladas de Dir4-CTP-Rx e Dir4-CTP-Tx respectivamente. Por outro lado, o bloco SRG 1 recebe as terminações lógicas SRG1-CP-Rx e SRG1-CP-Tx em suas portas de fronteiras.

As portas de fronteiras do bloco SRG que recebem os canais dos transponders são logicamente classificadas como PP, ou seja, existem 96 portas PP no nó SP-ROADM-1, uma porta PP para cada canal da banda C suportada. Por último, as portas lógicas TTP indicam ligações externas para outros nós ROADMs. As portas de fronteiras da direção 4 mostrado na Figura 24 recebem os nomes de Dir4-TTP-Rx e Dir4-TTP-Tx.

Figura 24 – Representação das terminações lógicas das ligações de adição do nó SP-ROADM-1.



Fonte: Elaborada pelo autor, baseado em [56].

As terminações lógicas podem ser definidas de acordo com a Figura 12 com a análise das conexões físicas. Desse modo, a terminação CP do bloco SRG 1 são as portas IN e OUT da placa ROADM, assim como as portas IN e OUT das placas ROADMs dos blocos de direção formam as terminações CTP. Além disso, as portas PP são pertencentes aos pares de portas parceiras da placa MUX/DEMUX do bloco SRG 1. Para a simulação, a porta PP27 foi utilizada pois representa o par de portas do canal C27. Ainda, a porta lógica TTP-Rx do bloco de direção 4 é

representada pela porta IN do amplificador Rx, em contraste com a porta OUT do amplificador Tx categorizando a porta lógica TTP-Tx.

As informações de terminações lógicas foram incluídas no ConfD de todos os nós ROADM através do atributo *logical-termination-point* do módulo do *circuit-pack*. O Quadro 35 ilustra a resposta RPC com algumas informações das terminações lógicas do nó SP-ROADM-1. O primeiro *circuit-pack* de nome (*circuit-pack-name*) Dir4-WSS apresenta duas portas de nome (*port-name*) Dir4-WSS-IN1 e Dir4-WSS-OUT1. A porta de direção (*port-direction*) de valor *rx* tem o ponto de terminação lógica (*logical-connection-point*) igual à Dir4-CTP-Rx, enquanto sua porta parceira de direção *tx* foi definida como Dir4-CTP-Tx. Analogicamente, o mesmo comportamento acontece para as portas IN e OUT do bloco do SRG 1, porém com as terminações lógicas são equivalentes à SRG1-CP-Rx e SRG1-CP-Tx.

Quadro 35 – Resposta RPC do ConfD com dados de terminações lógicas nas portas de *circuit-packs*.

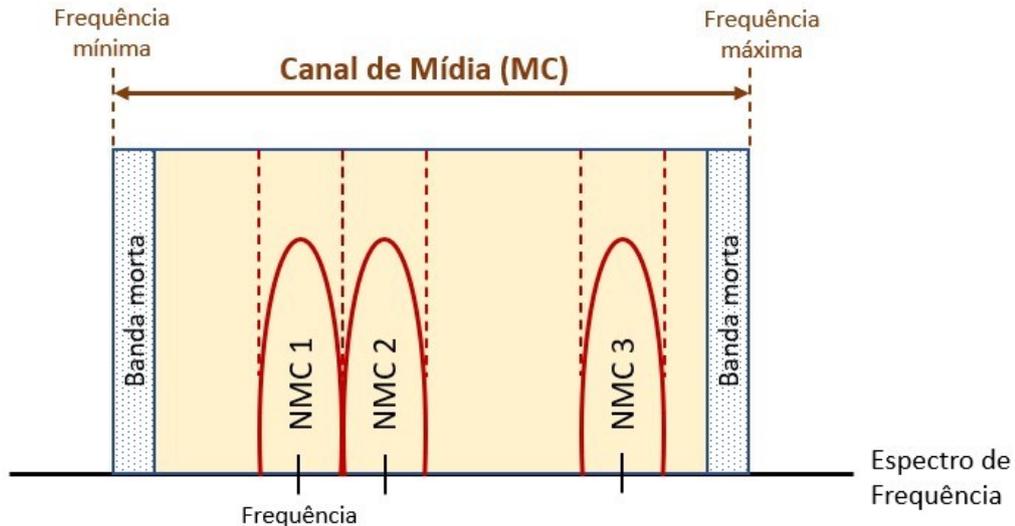
```
# netconf-console --rpc=query_circuit_packs_logical.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <org-openroadm-device xmlns="http://org/openroadm/device">
      <circuit-packs>
        <circuit-pack-name>Dir4-WSS</circuit-pack-name>
        <ports>
          <port-name>Dir4-WSS-IN1</port-name>
          <port-direction>rx</port-direction>
          <logical-connection-point>Dir4-CTP-Rx</logical-connection-point>
        </ports>
        <ports>
          <port-name>Dir4-WSS-OUT1</port-name>
          <port-direction>tx</port-direction>
          <logical-connection-point>Dir4-CTP-Tx</logical-connection-point>
        </ports>
        [...]
      </circuit-packs>
      <circuit-packs>
        <circuit-pack-name>SRG1-WSS</circuit-pack-name>
        <ports>
          <port-name>SRG1-WSS-IN1</port-name>
          <port-direction>rx</port-direction>
          <logical-connection-point>SRG1-CP-Rx</logical-connection-point>
        </ports>
        <ports>
          <port-name>SRG1-WSS-OUT1</port-name>
          <port-direction>tx</port-direction>
          <logical-connection-point>SRG1-CP-Tx</logical-connection-point>
        </ports>
        [...]
      </circuit-packs>
    </org-openroadm-device>
  </data>
</rpc-reply>
```

As terminações lógicas PP e TTP seguem o mesmo padrão apresentado para os outros tipos de ligações lógicas. No ConfD foi inserido para a porta Dir4-RxAMP-IN do *circuit-pack* Dir4-RxAMP o valor lógico de Dir4-TTP-Rx, e ao sentido oposto, o valor de Dir4-TTP-Tx para a porta Dir4-TxAMP-OUT do *circuit-pack* Dir4-TxAMP. Finalmente, os pares de portas SRG1-MUX-DEMUX-IN27 e SRG1-MUX-DEMUX-OUT27 da placa MUX/DEMUX receberem as terminações lógicas SRG1-PP27-IN e SRG1-PP27-OUT respectivamente.

Após a definição das terminações lógicas no nó SP-ROADM-1, o controlador SDN possui as informações necessárias para provisionar as configurações do serviço. A mesma dinâmica de criação de interfaces apresentada na configuração do nó transponder também acontece no nó ROADM. Porém, as camadas do protocolo de transporte são superiores e incorporam as camadas recebidas do nó transponder.

Primeiramente, as definições das interfaces de canal de mídia (MC, *Media Channel*) e canal de mídia da rede (NMC, *Network Media Channel*) são importantes para compreender como o sinal cliente é encapsulado nas camadas definidas no modelo de equipamentos. A Figura 25 ilustra como é a relação entre tais interfaces. A interface MC define uma banda de frequência no meio físico por meio dos parâmetros de frequências mínima e máxima. Por outro lado, as interfaces NMC ocupam um espaço dentro de uma interface MC exceto nas bandas mortas como mostrado na Figura 25. Diante disso, a combinação dos tamanhos de cada interface permite que a tecnologia de espaçamento flexível seja abstraída no modelo. Por exemplo, uma interface MC de tamanho de 50 GHz contendo apenas uma interface NMC implica na tecnologia de espaçamento fixo de 50 GHz.

Figura 25 – Exemplo das entidades canal de mídia (MC) e canal de mídia da rede (NMC).



Fonte: Elaborada pelo autor, baseado em [59].

A primeira interface que o controlador SDN deve instanciar no nó SP-ROADM-1 é a terminação lógica PP27 do bloco SRG 1. O Quadro 36 ilustra as informações de tal interface que foi criada no ConfD com o nome (*name*) NMC-CTP-SRG1-PP27-RX-192.7 e tipo (*type*) NMC-CTP, nomenclatura originada da combinação das entidades lógicas. A terminação lógica PP27 no refere-se à porta SRG1-MUX-DEMUX-IN27 do *circuit-pack* SRG1-MUX-DEMUX de acordo com a mensagem RPC. O tipo de interface NMC-CTP permite que seja informado as informações de frequência e espaçamento do canal, os quais estão declarados com os valores de 192.7 THz (C27) e 50 GHz nesta ordem.

Quadro 36 – Resposta RPC do ConfD para a interface NMC-CTP do bloco SRG 1.

```
# netconf-console --rpc=query_nmc_ctp_srg.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <org-openroadm-device xmlns="http://org/openroadm/device">
      <interface>
        <name>NMC-CTP-SRG1-PP27-RX-192.7</name>
        <type>networkMediaChannelConnectionTerminationPoint</type>
        <administrative-state>inService</administrative-state>
        <supporting-circuit-pack-name>SRG1-MUX-DEMUX</supporting-circuit-pack-name>
        <supporting-port>SRG1-MUX-DEMUX-IN27</supporting-port>
        <nmc-ctp xmlns="http://org/openroadm/network-media-channel-interfaces">
          <frequency>192.7</frequency>
          <width>50.0</width>
        </nmc-ctp>
      </interface>
    </org-openroadm-device>
  </data>
</rpc-reply>
```

Fonte: Elaborada pelo autor.

Após a configuração da interface NMC-CTP no bloco SRG 1, o controlador cria uma interface do tipo OTS para a terminação lógica TTP do bloco de direção 4 do nó SP-ROADM-1. A camada OTS no protocolo OTN é a mais camada de mais alta ordem e é responsável por representar os enlaces ópticos entre os amplificadores na linha de transmissão. A interface que implementa tal camada está definida no Quadro 37 com o nome (*name*) OTS-Dir4-TTP-TX do tipo (*type*) transporte óptico (*opticalTransport*). Percebe-se ainda na mensagem RPC que a porta Dir4-TxAMP-OUT do *circuit-pack* Dir4-TxAMP representa a terminação lógica TTP-TX. Ainda por cima, o atributo *ots* modela algumas propriedades de tal camada, por exemplo, o tipo da fibra (*fiber-type*) definido como fibra óptica monomodo padrão (*smf*) na mensagem RPC e as perdas de atenuação de recepção (*span-loss-receive*) e transmissão (*span-loss-transmit*). Ambos parâmetros são necessários para que o controlador SDN consiga configurar o ganho e potência de lançamento dos amplificadores corretamente.

Quadro 37 – Resposta RPC do ConfD para a interface OTS do bloco de direção 4.

```
# netconf-console --rpc=query_ots_dir4.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <org-openroadm-device xmlns="http://org/openroadm/device">
      <interface>
        <name>OTS-Dir4-TTP-TX</name>
        <type>opticalTransport</type>
        <administrative-state>inService</administrative-state>
        <supporting-circuit-pack-name>Dir4-TxAMP</supporting-circuit-pack-name>
        <supporting-port>Dir4-TxAMP-OUT</supporting-port>
        <ots xmlns="http://org/openroadm/optical-transport-interfaces">
          <fiber-type>smf</fiber-type>
          <span-loss-receive>15.0</span-loss-receive>
          <span-loss-transmit>3.0</span-loss-transmit>
        </ots>
      </interface>
    </org-openroadm-device>
  </data>
</rpc-reply>
```

Fonte: Elaborada pelo autor.

A posterior interface a ser criada pelo controlador SDN para abstrair a camada OMS que é incluída abaixo da camada OTS no protocolo OTN está ilustrada no Quadro 38. É possível interpretar na mensagem RPC que tal interface tem nome (*name*) OMS-Dir4-TTP-TX de tipo (*type*) multiplexação óptica (*openROADMOpticalMultiplex*). Além disso, os valores de porta e de *circuit-pack* de suportes são idênticos a interface OTS pois fazem referência a mesma terminação lógica TTP-TX. Para indicar a correlação entre as camadas OTS e OMS, o atributo de lista de

suporte de interfaces (*supporting-interface-list*) é usado e tem o nome da interface OTS. O modelo de equipamentos não oferece nenhum atributo adicional para as interfaces do tipo OMS diferentes dos outros tipos de interface.

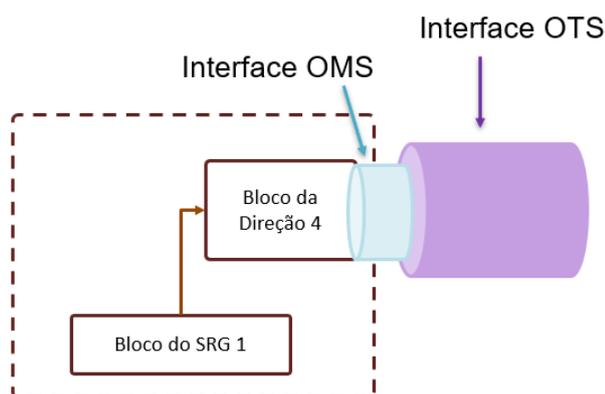
Quadro 38 – Resposta RPC do ConfD para a interface OMS do bloco de direção 4.

```
# netconf-console --rpc=query_oms_dir4.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <org-openroadm-device xmlns="http://org/openroadm/device">
      <interface>
        <name>OMS-Dir4-TTP-TX</name>
        <type>openROADMOpticalMultiplex</type>
        <administrative-state>inService</administrative-state>
        <supporting-circuit-pack-name>Dir4-TxAMP</supporting-circuit-pack-name>
        <supporting-port>Dir4-TxAMP-OUT</supporting-port>
        <supporting-interface-list>OTS-Dir4-TTP-TX</supporting-interface-list>
      </interface>
    </org-openroadm-device>
  </data>
</rpc-reply>
```

Fonte: Elaborada pelo autor.

A Figura 26 ilustra a representação em forma de tubo das interfaces OMS e OTS criadas no bloco de direção 4 do nó ROADM. É possível observar a relação de hierarquia das interfaces de transporte onde a interface OMS está encapsulada dentro da interface OTS, de ordem maior.

Figura 26 – Interfaces OMS e OTS no bloco de direção.



Fonte: Elaborada pelo autor.

Com as camadas OTS e OMS disponíveis, o modelo do equipamento define que a interface MC seja criada na interface cuja terminação lógica é o TTP. A interface MC, como mencionado anteriormente, aloca uma banda de frequência na fibra óptica de transmissão. Portanto, a nomenclatura do tipo lógico combinado é dita como MC-TTP. O Quadro 39 apresenta a resposta do ConfD para a interface de nome MC-TTP-Dir4-TTP-TX-192.7 e tipo MC-TTP

(*mediaChannelTrailTerminationPoint*). É possível encontrar na mensagem RPC que a interface de suporte da camada MC-TTP é a interface OMS de nome OMS-Dir4-TTP-TX. Esse tipo de interface permite que as informações de frequência mínima (*min-freq*) e máxima (*max-freq*) sejam definidas para indicar a banda de frequência utilizada. O ponto médio dos valores de frequências indicados na resposta RPC é equivalente ao valor de 192.7 THz, definindo o canal C27 desejado.

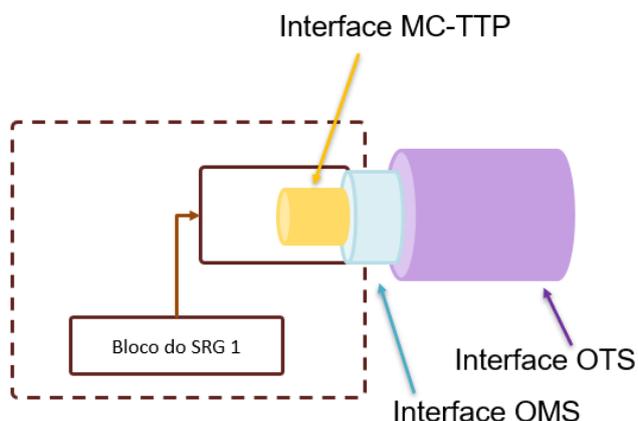
Quadro 39 – Resposta RPC do ConfD para a interface MC-TTP do bloco de direção 4.

```
# netconf-console --rpc=query_mc_dir4.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <org-openroadm-device xmlns="http://org/openroadm/device">
      <interface>
        <name>MC-TTP-Dir4-TTP-TX-192.7</name>
        <type>mediaChannelTrailTerminationPoint</type>
        <administrative-state>inService</administrative-state>
        <supporting-circuit-pack-name>Dir4-TxAMP</supporting-circuit-pack-name>
        <supporting-port>Dir4-TxAMP-OUT</supporting-port>
        <supporting-interface-list>OMS-Dir4-TTP-TX</supporting-interface-list>
        <mc-ttp xmlns="http://org/openroadm/media-channel-interfaces">
          <min-freq>192.675</min-freq>
          <max-freq>192.725</max-freq>
        </mc-ttp>
      </interface>
    </org-openroadm-device>
  </data>
</rpc-reply>
```

Fonte: Elaborada pelo autor.

A Figura 27 ilustra a representação em forma de tubo da interface MC-TTP que está contida pela interface OMS previamente criada no bloco de direção 4 do nó ROADM.

Figura 27 – Interface MC-TTP do bloco de direção.



Fonte: Elaborada pelo autor.

A última interface criada pelo controlador SDN na terminação lógica TTP do bloco de direção 4 é a interface NMC, que está contida em uma interface MC previamente instanciada.

Como o espaçamento da simulação é fixo de 50 GHz, poderá apenas existir uma interface NMC no ConfD para transportar o sinal cliente. O Quadro 40 exibe a resposta RPC da interface de nome NMC-CTP-Dir1-TTP-TX-192.7 e tipo NMC-CTP assim como a interface de suporte MC, indicando o relacionamento entre as camadas. O controlador SDN deve configurar nesse tipo de interface a frequência e a largura do canal por meio dos atributos *frequency* e *width*. Os valores 192.7 THz e 50.0 GHz foram configurados para tais atributos para a simulação do provisionamento de serviço.

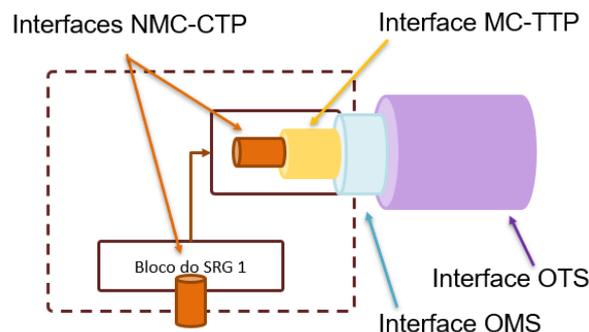
Quadro 40 – Resposta RPC do ConfD para a interface NMC-TTP do bloco de direção 4.

```
# netconf-console --rpc=query_nmc_dir4.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:iETF:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <org-openroadm-device xmlns="http://org/openroadm/device">
      <interface>
        <name>NMC-CTP-Dir4-TTP-TX-192.7</name>
        <type>networkMediaChannelConnectionTerminationPoint</type>
        <administrative-state>inService</administrative-state>
        <supporting-circuit-pack-name>Dir4-TxAMP</supporting-circuit-pack-name>
        <supporting-port>Dir4-TxAMP-OUT</supporting-port>
        <supporting-interface-list>MC-TTP-Dir4-TTP-TX-192.7</supporting-interface-
list>
        <nmc-ctp xmlns="http://org/openroadm/network-media-channel-interfaces">
          <frequency>192.7</frequency>
          <width>50.0</width>
        </nmc-ctp>
      </interface>
    </org-openroadm-device>
  </data>
</rpc-reply>
```

Fonte: Elaborada pelo autor.

A Figura 28 ilustra a representação em forma de tubo das interfaces NMC-CTP criadas nos blocos de direção 4 e SRG 1 do nó ROADM. É possível observar que a interface NMC-CTP está contida pela interface MC-TTP, mostrando sua relação hierárquica de camadas.

Figura 28 – Interface NMC-CTP do bloco de direção.



Fonte: Elaborada pelo autor.

O módulo do conexões ROADMs (*roadm-connections*) do modelo do equipamento é utilizado para representar as crossconexões de interfaces NMC-CTP que foram criadas do nó SP-ROADM-1. Tal informação criada pelo controlador SDN é necessária para o provisionamento de serviços, indicando quais são as interfaces lógicas por onde o sinal cliente trafega. O Quadro 41 ilustra a mensagem RPC do ConfD com a conexão ROADM criada com seus atributos. Recomenda-se um padrão para o nome da conexão dado pelo atributo *connection-name* para conter todas as informações das portas NMC-CTP. Além do mais, a conexão tem como interface origem (*src-if*) a NMC-CTP-SRG1-PPN-RX-192.7 e a porta destino (*dst-if*) a NMC-CTP-Dir4-TTP-TX-192.7. Tais interfaces, respectivamente, pertencem aos blocos SRG1 e de direção 4.

Quadro 41 – Resposta RPC do ConfD da crossconexão de interfaces do nó SP-ROADM-1.

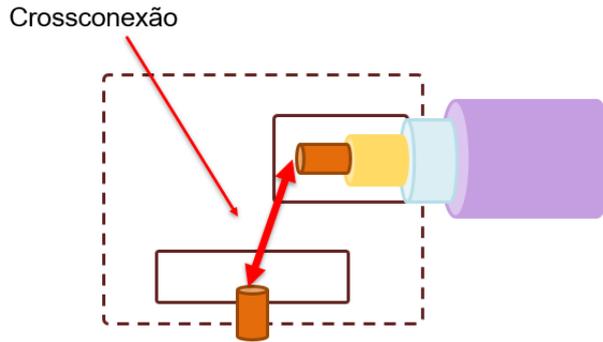
```
# netconf-console --rpc=query_roadm_connections.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:iETF:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <org-openroadm-device xmlns="http://org/openroadm/device">
      <roadm-connections>
        <connection-name>NMC-CTP-SRG1-PP27-RX-192.7-to-NMC-CTP-Dir4-TTP-TX-
192.7</connection-name>
        <opticalControlMode>off</opticalControlMode>
        <target-output-power>0.0</target-output-power>
        <source>
          <src-if>NMC-CTP-SRG1-PP27-RX-192.7</src-if>
        </source>
        <destination>
          <dst-if>NMC-CTP-Dir4-TTP-TX-192.7</dst-if>
        </destination>
      </roadm-connections>
    </org-openroadm-device>
  </data>
</rpc-reply>
```

Fonte: Elaborada pelo autor.

É ainda possível notar no Quadro 41 os atributos modo de controle óptico (*opticalControlMode*) e potência de saída alvo (*target-output-power*). O controlador SDN utiliza tais atributos para verificar se os amplificadores conseguem fornecer potência óptica suficiente para transmissão do sinal. De qualquer modo, as configurações de potência das conexões não serão abordadas na simulação.

A Figura 29 ilustra a crossconexão entre as camadas NMC-CTP do bloco do SRG 1 e do bloco de direção 4 do nó SP-ROADM-1, permitindo a passagem do tráfego de forma transparente do canal C27 que contém o sinal cliente ODU4.

Figura 29 – Crossconexão entre as interfaces NMC-CTP do bloco de direção.

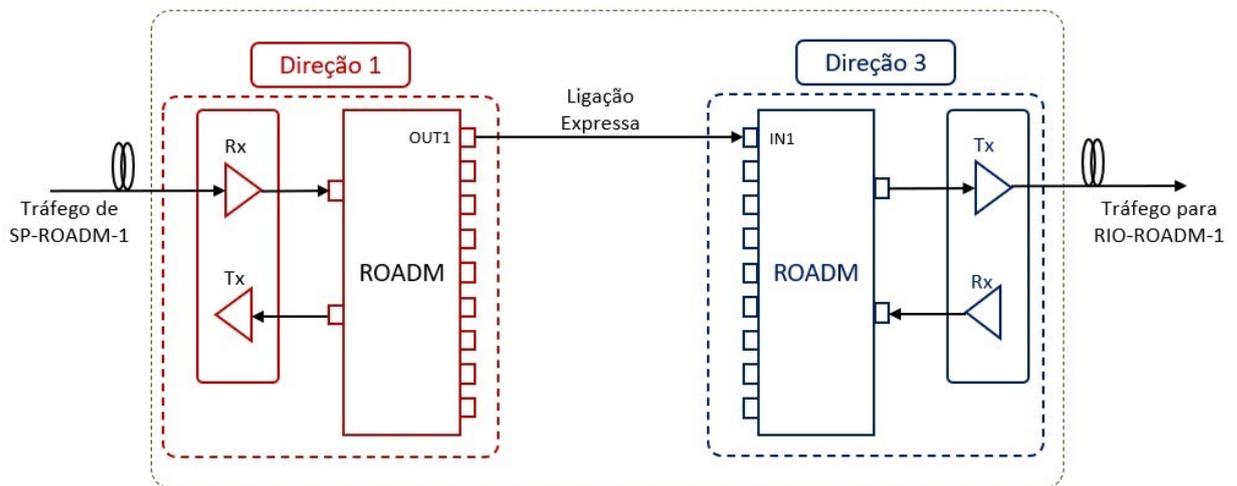


Fonte: Elaborada pelo autor.

5.1.4. Nó ROADM da Ligação Expressa

Após a configuração dos nós localizados em São Paulo, o controlador SDN irá processar o nó BH-ROADM-1 localizado em Belo Horizonte. A função deste nó é servir como nó de passagem do tráfego que vem da estação de São Paulo para a estação do Rio de Janeiro. Para isso é necessário existir uma ligação física do tipo expressa para conectar o bloco da direção 1 ao bloco da direção 3, apenas nesse sentido pois essa simulação considera apenas o serviço fim-a-fim unidirecional. A Figura 30 ilustra a configuração física inserida no ConfD para esse nó, considerando a ligação expressa da porta OUT1 para a porta IN1 dos blocos de direções 1 e 3 respectivamente.

Figura 30 – Configuração física do nó BH-ROADM-1.



Nó: BH-ROADM-1

Localização: Belo Horizonte, MG

Fonte: Elaborada pelo autor.

De acordo com a recomendação [59], o controlador SDN cria interfaces lógicas nas portas com a terminação lógica TTP dos dois blocos de direção desse nó ROADM. No cenário da simulação, a porta TTP-RX é a porta do amplificador Rx do bloco da direção 1, enquanto a porta

OUT do amplificador Tx é a porta TTP-TX assim como está exposto na Figura 30. As interfaces lógicas que o controlador SDN cria representam as camadas OTS, OMS, MC-TTP e NMC-CTP, as quais já foram exploradas na subseção anterior. Portanto, é necessário instanciar todas essas interfaces associadas nas duas portas lógicas TTP-RX e TTP-TX, e então finalmente, criar uma crossconexão entre as interfaces NMC-CTP.

O Quadro 42 exibe todas as interfaces instanciadas no ConfD para o bloco de direção 1 do nó BH-ROADM-1. Os mesmos tipos de interfaces são também criados para o bloco de direção 3, mas não serão apresentadas pois tem as mesmas características já exibidas na resposta RPC. Os nomes das interfaces são OTS-Dir1-TTP-RX, OMS-Dir1-TTP-RX, MC-TTP-Dir1-TTP-RX-192.7 e NMC-CTP-Dir1-TTP-RX-192.7 representando as camadas lógicas OTS, OMS, MC e NMC-CTP nesta ordem. Também é possível identificar que as interfaces MC e NMC-CTP delimitam a banda de frequência do canal C27, o qual está transportando o sinal cliente.

Quadro 42 – Respostas RPC do ConfD para as interfaces da porta TTP-RX da direção 1.

```
# netconf-console --rpc=query_bh_dir1_interfaces.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:iETF:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <org-openroadm-device xmlns="http://org/openroadm/device">

      <interface>
        <name>OTS-Dir1-TTP-RX</name>
        <type>opticalTransport</type>
        <administrative-state>inService</administrative-state>
        <supporting-circuit-pack-name>Dir1-RxAMP</supporting-circuit-pack-name>
        <supporting-port>Dir1-RxAMP-IN</supporting-port>
        <ots xmlns="http://org/openroadm/optical-transport-interfaces">
          <fiber-type>smf</fiber-type>
          <span-loss-receive>15.0</span-loss-receive>
          <span-loss-transmit>3.0</span-loss-transmit>
        </ots>
      </interface>

      <interface>
        <name>OMS-Dir1-TTP-RX</name>
        <type>openROADMOpticalMultiplex</type>
        <administrative-state>inService</administrative-state>
        <supporting-circuit-pack-name>Dir1-RxAMP</supporting-circuit-pack-name>
        <supporting-port>Dir1-RxAMP-IN</supporting-port>
        <supporting-interface-list>OTS-Dir1-TTP-RX</supporting-interface-list>
      </interface>

      <interface>
        <name>MC-TTP-Dir1-TTP-RX-192.7</name>
        <type>mediaChannelTrailTerminationPoint</type>
        <administrative-state>inService</administrative-state>
        <supporting-circuit-pack-name>Dir1-RxAMP</supporting-circuit-pack-name>
        <supporting-port>Dir1-RxAMP-IN</supporting-port>
        <supporting-interface-list>OMS-Dir1-TTP-RX</supporting-interface-list>
        <mc-ttp xmlns="http://org/openroadm/media-channel-interfaces">
```

```

        <min-freq>192.675</min-freq>
        <max-freq>192.725</max-freq>
    </mc-ctp>
</interface>

<interface>
  <name>NMC-CTP-Dir1-TTP-RX-192.7</name>
  <type>networkMediaChannelConnectionTerminationPoint</type>
  <administrative-state>inService</administrative-state>
  <supporting-circuit-pack-name>Dir1-RxAMP</supporting-circuit-pack-name>
  <supporting-port>Dir1-RxAMP-IN</supporting-port>
  <supporting-interface-list>MC-TTP-Dir1-TTP-RX-192.7</supporting-interface-
list>
  <nmc-ctp xmlns="http://org/openroadm/network-media-channel-interfaces">
    <frequency>192.7</frequency>
    <width>50.0</width>
  </nmc-ctp>
</interface>

</org-openroadm-device>
</data>
</rpc-reply>

```

Fonte: Elaborada pelo autor.

Após a disponibilidade de todas as interfaces para os dois blocos de direções 1 e 3 no ConfD, o controlador SDN estabelece uma crossconexão entre os blocos usando as interfaces NMC-CTP. Tal conexão é apresentada no Quadro 43 pela mensagem RPC do módulo de conexões ROADM. É possível interpretar que esse crossconexão faz apenas um sentido do circuito de origem (*src-if*) NMC-CTP-Dir1-TTP-RX-192.7 e destino (*dst-if*) NMC-CTP-Dir3-TTP-TX-192.7. Em outros termos, a conexão vem do bloco de direção 1 para o bloco de direção 3. Para as aplicações de serviço bidirecional, seria necessário criar outra crossconexão como origem do bloco de direção 3 para o bloco de direção 1. Porém, a simulação está estabelecendo um serviço unidirecional.

Quadro 43 – Resposta RPC do ConfD da crossconexão de interfaces do nó BH-ROADM-1.

```

# netconf-console --rpc=query_roadm_connections.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <org-openroadm-device xmlns="http://org/openroadm/device">
      <roadm-connections>
        <connection-name>NMC-CTP-Dir1-TTP-RX-192.7-to-NMC-CTP-Dir3-TTP-TX-
192.7</connection-name>
        <source>
          <src-if>NMC-CTP-Dir1-TTP-RX-192.7</src-if>
        </source>
        <destination>
          <dst-if>NMC-CTP-Dir3-TTP-TX-192.7</dst-if>
        </destination>
      </roadm-connections>
    </org-openroadm-device>

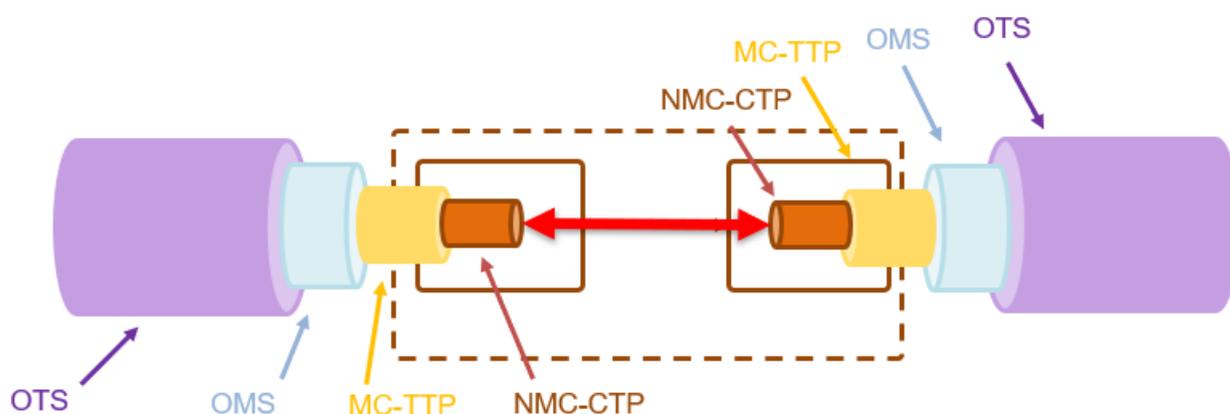
```

</data>
</rpc-reply>

Fonte: Elaborada pelo autor.

A Figura 31 ilustra a crossconexão entre as camadas NMC-CTP do bloco de direções 1 e 3 do nó BH-ROADM-1, permitindo a passagem do tráfego de forma transparente do canal C27 que contém o sinal cliente ODU4. É possível observar também a relação hierárquica decrescente das camadas de transporte OTS, OMS, MC-TTP e NMC-CTP, instanciadas em ambos os blocos de direções.

Figura 31 – Crossconexão e interfaces dos blocos de direção.



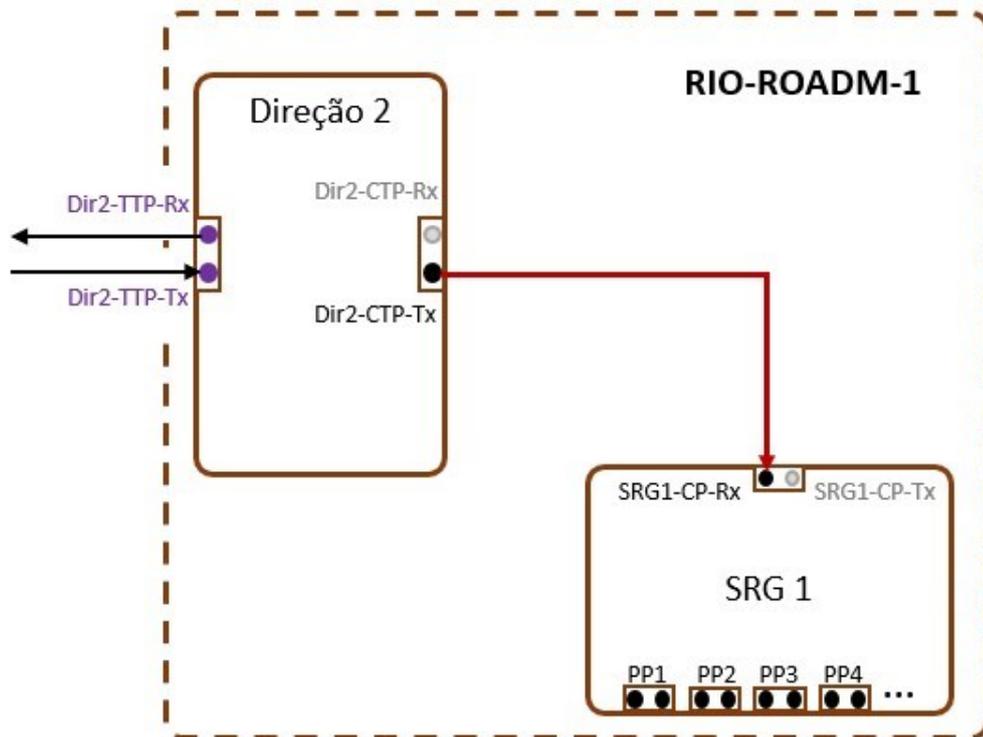
Fonte: Elaborada pelo autor.

5.1.5. Nó ROADM da Ligação da Remoção

O controlador SDN atua nos nós localizados no Rio de Janeiro após a configuração das nós anteriores localizados em São Paulo e Belo Horizonte. O primeiro nó é um nó do tipo ROADM com o nome RIO-ROADM-1 e sua função no provisionamento do serviço ODU4 fim-a-fim é prover uma ligação de remoção, isolando e recuperando o canal C27 da malha de transmissão para disponibilizá-lo para o nó transponder.

Similarmente a ligação de adição, a ligação de remoção utiliza as terminações lógicas Dir2-TTP e SRG1-PP27 do nó RIO-ROADM-1. O controlador SDN identifica qual é a porta cuja a terminação lógica é Dir2-TTP-Rx no bloco de direção 2 e a outra porta no bloco SRG1 de terminação lógica SRG1-PP27-OUT, como ilustrado na Figura 32. Em outros termos, a porta Dir2-TTP-Rx é responsável por receber o sinal óptico do nó BH-ROADM-1 enquanto a porta SRG1-PP27-OUT tem a função de entregar somente o canal C27 para nó transponder.

Figura 32 – Ligação de remoção do bloco de direção 2 do nó RIO-ROADM-1



Fonte: Elaborada pelo autor, baseado em [56].

O controlador SDN deve criar todos os tipos de interfaces necessárias para prover a função de remoção de canal. Para isso, o bloco de direção 2 deve conter as camadas OTS, OMS, MC-TTP e NMC-CTP, e também o bloco SRG 1 com uma interface NMC-CTP. Todos esses tipos de interfaces já foram expostos e tem configuração similar ao nó de ligação de adição, mas com sentido oposto.

O Quadro 44 apresenta as interfaces instanciadas no ConfD para o nó RIO-ROADM-1. É possível identificar na mensagem RPC que as portas de suporte (*supporting-port*) referenciadas são a porta Dir2-RxAMP-IN do *circuit-pack* Dir2-RxAMP e a outra porta SRG1-MUX-DEMUX-OUT27 do *circuit-pack* SRG1-MUX-DEMUX. De fato, tais portas possuem as terminações lógicas Dir2-TTP-Rx e SRG1-PP27-OUT respectivamente. Além disso, existem quatro interfaces do tipo OTS, OMS, MC-TTP e NMC-CTP para o bloco de direção 2 e uma interface NMC-CTP para o bloco SRG 1, com configurações que delimitam o canal C27 para recuperar o sinal cliente.

Quadro 44 – Respostas RPC do ConfD para as interfaces das portas do nó RIO-ROADM-1.

```
# netconf-console --rpc=query_rio_dir2_interfaces.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <org-openroadm-device xmlns="http://org/openroadm/device">
```

```

<interface>
  <name>OTS-Dir2-TTP-RX</name>
  <type>opticalTransport</type>
  <administrative-state>inService</administrative-state>
  <supporting-circuit-pack-name>Dir2-RxAMP</supporting-circuit-pack-name>
  <supporting-port>Dir2-RxAMP-IN</supporting-port>
  <ots xmlns="http://org/openroadm/optical-transport-interfaces">
    <fiber-type>smf</fiber-type>
    <span-loss-receive>15.0</span-loss-receive>
    <span-loss-transmit>3.0</span-loss-transmit>
  </ots>
</interface>

<interface>
  <name>OMS-Dir2-TTP-RX</name>
  <type>openROADMOpticalMultiplex</type>
  <administrative-state>inService</administrative-state>
  <supporting-circuit-pack-name>Dir2-RxAMP</supporting-circuit-pack-name>
  <supporting-port>Dir2-RxAMP-IN</supporting-port>
  <supporting-interface-list>OTS-Dir2-TTP-RX</supporting-interface-list>
</interface>

<interface>
  <name>MC-TTP-Dir2-TTP-RX-192.7</name>
  <type>mediaChannelTrailTerminationPoint</type>
  <administrative-state>inService</administrative-state>
  <supporting-circuit-pack-name>Dir2-RxAMP</supporting-circuit-pack-name>
  <supporting-port>Dir2-RxAMP-IN</supporting-port>
  <supporting-interface-list>OMS-Dir2-TTP-RX</supporting-interface-list>
  <mc-ttp xmlns="http://org/openroadm/media-channel-interfaces">
    <min-freq>192.675</min-freq>
    <max-freq>192.725</max-freq>
  </mc-ttp>
</interface>

<interface>
  <name>NMC-CTP-Dir2-TTP-RX-192.7</name>
  <type>networkMediaChannelConnectionTerminationPoint</type>
  <administrative-state>inService</administrative-state>
  <supporting-circuit-pack-name>Dir2-RxAMP</supporting-circuit-pack-name>
  <supporting-port>Dir2-RxAMP-IN</supporting-port>
  <supporting-interface-list>MC-TTP-Dir2-TTP-RX-192.7</supporting-interface-
list>
  <nmc-ctp xmlns="http://org/openroadm/network-media-channel-interfaces">
    <frequency>192.7</frequency>
    <width>50.0</width>
  </nmc-ctp>
</interface>

<interface>
  <name>NMC-CTP-SRG1-PP27-TX-192.7</name>
  <type>networkMediaChannelConnectionTerminationPoint</type>
  <administrative-state>inService</administrative-state>
  <supporting-circuit-pack-name>SRG1-MUX-DEMUX</supporting-circuit-pack-name>
  <supporting-port>SRG1-MUX-DEMUX-OUT27</supporting-port>
  <nmc-ctp xmlns="http://org/openroadm/network-media-channel-interfaces">
    <frequency>192.7</frequency>
    <width>50.0</width>
  </nmc-ctp>

```

```
</interface>

</org-openroadm-device>
</data>
</rpc-reply>
```

Fonte: Elaborada pelo autor.

Após a criação das interfaces NMC-CTP nos blocos do nó RIO-ROADM-1, o controlador SDN estabelecer uma crossconexão usando o módulo de conexões ROADMs (*roadm-connections*) no sentido unidirecional de origem do bloco de direção 2 para o bloco SRG 1. Tal crossconexão é ilustrada no Quadro 45 onde a interface de origem (*src-if*) é a NMC-CTP-Dir2-TTP-RX-192.7 e a interface de destino (*dst-if*) é a NMC-CTP-SRG1-PP27-TX-192.7. Identifica-se também que o nome da conexão segue o mesmo padrão utilizado nas outras crossconexões previamente exploradas.

Quadro 45 – Resposta RPC do ConfD da crossconexão de interfaces do nó RIO-ROADM-1.

```
# netconf-console --rpc=query_roadm_connections.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <org-openroadm-device xmlns="http://org/openroadm/device">
      <roadm-connections>
        <connection-name>NMC-CTP-Dir2-TTP-RX-192.7-to-NMC-CTP-SRG1-PP27-TX-
192.7</connection-name>
        <source>
          <src-if>NMC-CTP-Dir2-TTP-RX-192.7</src-if>
        </source>
        <destination>
          <dst-if>NMC-CTP-SRG1-PP27-TX-192.7</dst-if>
        </destination>
      </roadm-connections>
    </org-openroadm-device>
  </data>
</rpc-reply>
```

Fonte: Elaborada pelo autor.

5.1.6. Nó Transponder de Recepção

O último nó que o controlador SDN configura é o nó RIO-Transponder-1, responsável por remover todas as camadas ópticas de transporte e entregar o sinal cliente nas mesmas características iniciais. Esse nó transponder, assim como o nó SP-Transponder-1, é composto por um transponder com uma porta cliente e outra rede na taxa de linha de 100 Gbps.

Como o serviço fim-a-fim é do tipo ODU4, o controlador deve primeiramente configurar a porta rede com as camadas ópticas de transporte OCh, OTU e ODU. Logo, ajustar a

porta cliente com a camada ODU para ser disponibilizada, e então criar uma crossconexão usando as interfaces da camada ODU para finalizar o provisionamento do serviço em todos os nós da rede.

As interfaces que foram instanciadas no ConfD para o nó RIO-Transponder-1 tem o mesmo conjunto de dados do nó SP-Transponder-1 já apresentados nos Quadro 30, Quadro 31, Quadro 32 e Quadro 33 para ambas as portas rede e cliente. A causa da idêntica configuração é que o nó transponder foi modelado com caixas de formato de transceptores CFP bidirecional. Em outros termos, as interfaces criadas já suportam duas direções Tx e Rx satisfazendo as funções do transponder de transmissão (Tx) do nó SP-Transponder-1 e do transponder de recepção (Rx) do nó RIO-Transponder-1.

A única configuração distinta entre os nós transponders está ilustrada no Quadro 46 que consiste na crossconexão do nó RIO-Transponder-1 usando as interfaces ODU4. Ainda na mensagem RPC, é possível verificar que a origem (*src-if*) da conexão é a interface ODU4-XPDR1-NETWORK1-TXRX e o destino (*dst-if*) é a interface ODU4-XPDR1-CLIENT1-TXRX. Apesar que as interfaces servirem no sentido bidirecional, o controlador SDN fixa a sequência das interfaces do sentido unidirecional, da porta rede para a porta cliente. Portanto, essa é a última configuração necessária para o provisionamento fim-a-fim do serviço ODU4.

Quadro 46 – Resposta RPC do ConfD da crossconexão de interfaces do nó RIO-Transponder-1.

```
# netconf-console --rpc=query_odu_connections.xml
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <org-openroadm-device xmlns="http://org/openroadm/device">
      <odu-connection>
        <connection-name>ODU4-XPDR1-NETWORK1-TXRX-to-ODU4-XPDR1-CLIENT1-
TXRX</connection-name>
        <source>
          <src-if>ODU4-XPDR1-NETWORK1-TXRX</src-if>
        </source>
        <destination>
          <dst-if>ODU4-XPDR1-CLIENT1-TXRX</dst-if>
        </destination>
      </odu-connection>
    </org-openroadm-device>
  </data>
</rpc-reply>
```

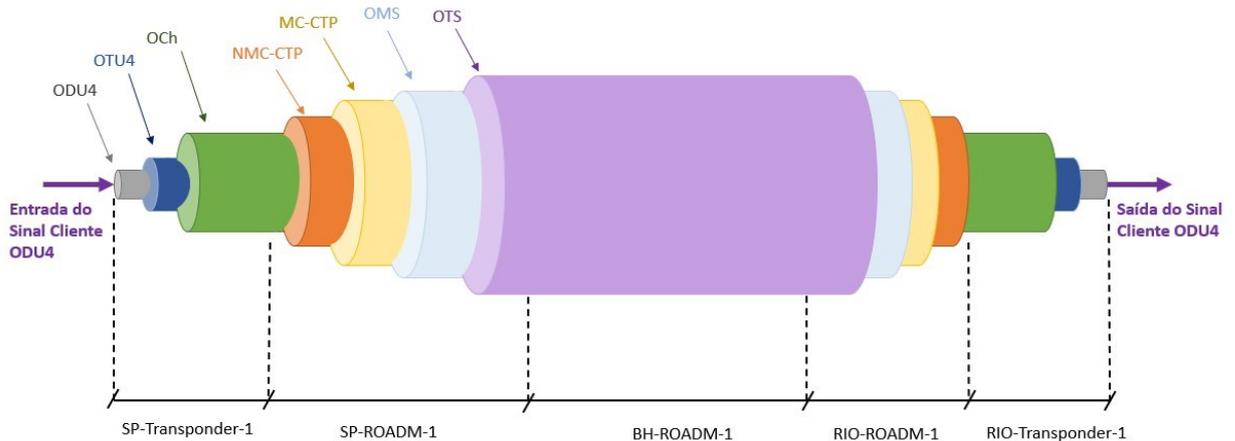
Fonte: Elaborada pelo autor.

5.2. Resultados

Todas as configurações inseridas nos nós do cenário da simulação permitem que o controlador SDN visualize o serviço na topologia como apresentado na recomendação [59]. A Figura 33 ilustra todas as camadas ópticas que foram implementadas ao longo da simulação em

uma visão macroscópica. Inicialmente o sinal cliente é inserido no nó SP-Transponder-1 que é responsável por prover as camadas ODU4, OTU4 e OCh. Logo, o nó SP-ROADM-1 auxilia o transporte do canal óptico com as camadas NMC-CTP, MC-CTP, OMS e OTS. O nó intermediário BH-ROADM-1 somente regenera as camadas de transporte e envia o tráfego da camada OTS para o nó RIO-ROADM-1. Este nó é responsável por remover as camadas de transporte nesta ordem: OTS, MS, MC-CTP e NMC-CTP. Por fim, o nó transponder do RIO-Transponder-1 recebe o sinal OCh e remove as camadas ópticas OTU4 e ODU4 e entrega o sinal cliente na sua porta cliente configurada.

Figura 33 – Representação de todas as camadas do serviço fim-a-fim ODU4.



Fonte: Elaborada pelo autor.

Esse capítulo apresentou como os nós da rede são configurados para provisionar um serviço entre duas estações de telecomunicações de acordo com a especificação [59]. Além disso, foi exposto todas as interfaces de transporte que são instanciadas para permitir a passagem do tráfego.

Capítulo 6

Conclusões e Estudos Futuros

6.1. Conclusões

Ao decorrer desse trabalho, foi demonstrado os principais módulos do modelo YANG OpenROADM para desagregar um conjunto de equipamentos DWDM em nós do tipo ROADM e transponder. Além disso, a simulação consistiu em implementar os passos que o controlador SDN deve executar ao provisionar um serviço não protegido em vários nós desagregados de uma rede de transporte. Dado o exposto, é possível levantar vantagens e desvantagens do modelo de desagregação OpenROADM e serão detalhados a seguir.

A primeira vantagem do modelo OpenROADM é que permitiu modelar as características mais peculiares da camada óptica. Por exemplo, é possível escolher o tipo da modulação dos transponders para diversas modulações como o DP-QPSK e QAM16, bem como, os modos de operação dos amplificadores para controlar a potência de lançamento. Outra tecnologia suportada no modelo é o espaçamento flexível de canais no espectro óptico por meio da implementação das interfaces MC e NMC. Isso permite a desagregação dos equipamentos DWDM da última geração usando tal tecnologia.

Outro ponto relevante é a adaptabilidade e flexibilidade do modelo do equipamento para desagregar tecnologias proprietárias específicas. Pode-se presumir que para atender uma quantidade maior de provedores de telecomunicações, os fabricantes devem investir em produtos flexíveis como transponders com portas suportando tipos de transceptores de formatos diferentes. Como foi exposto anteriormente, é possível modelar qualquer restrição de um transponder usando os módulos de capacidade, de restrição e da matriz de conexão, definido qualquer limitação da tecnologia proprietária.

É notável que as interfaces criadas durante o provisionamento do serviço compilam as recomendações ITU [12] do protocolo OTN da rede de transporte. Cada nó da simulação foi responsável por abstrair as camadas ópticas para transportar o sinal cliente de forma transparente não intrusivo. Em adição, o modelo do equipamento permite que algumas das camadas sejam configuradas como a escolha do tipo da fibra da malha de transmissão para a camada OTS, bem como a perda de atenuação na recepção e transmissão.

Apesar de não ter sido demonstrado detalhadamente, as interfaces do tipo ODU oferecem uma abundância de funcionalidades de monitoração e operação previstas no quadro do protocolo ODU. É possível configurar o tipo de monitoração, SAPI, DAPI e TCM, assim como atuar no tipo de carga e o formato de multiplexação dentro do quadro OPU, definindo a posição e a porta dos tributários (sinais clientes). Por esse e outros motivos mencionados nesse trabalho, o modelo OpenROADM é eficiente para abstrair as camadas do protocolo OTN.

Durante a inserção de dados fictícios em diferentes simuladores, pode-se assumir que o modelo do equipamento permite a desagregação dos equipamentos DWDM em níveis bem realísticos, convertendo em alguns casos todas as funções ópticas e de configuração. Por um lado, isso é a expectativa mais promissora de um modelo de dados desagregador para a rede óptica de transporte, mas por outro lado, aumenta a complexidade e tamanho do modelo.

O modelo OpenROADM não oferece nenhuma ferramenta ou programa automatizado para auxiliar no processo de desagregação dos equipamentos e suas ligações físicas. Os fabricantes, de fato, devem prover as informações de inventário, capacidades e funções disponíveis para seus equipamentos desagregados. Entretanto, os dados de ligações físicas e externas entre as placas é providenciado pelo usuário manualmente podendo usar o controlador SDN para tal finalidade. De qualquer maneira, isso aumenta a complexidade do comissionamento dos equipamentos da rede para detalhar todos as configurações físicas.

Foi mencionado as principais vantagens e desvantagens da utilização e aplicação dos modelos OpenROADM e é possível concluir que esse modelo atende as necessidades de um modelo desagregador da rede de transporte. Mesmo assim, é sabido que as grandes redes legadas de transporte atualmente instaladas em campo podem não ser passíveis de uma transição para a arquitetura de desagregação. Então, é esperado que as redes de transporte lentamente irão evoluir para equipamentos desagregados de última geração à medida que os equipamentos legados forem sendo substituídos nas estações de telecomunicações.

6.2. Estudos Futuros

Com a disponibilização do modelo OpenROADM que permite a desagregação das características ópticas de vários equipamentos, os maiores desafios estarão nas aplicações de rede da camada de aplicação da arquitetura SDN. Por exemplo, umas das funcionalidades mais desejadas de um provedor de serviços de telecomunicações é ter uma rede autônoma e com inteligência artificial para reduzir o custo de operação e manutenção das redes. Como o controlador

SDN tem controle e visão de todos os equipamentos da rede, poderia ser possível executar manutenção preventiva de acordo com dados históricos de rompimento de fibras.

Outro tópico em aberto é a otimização de recursos com equipamentos de espaçamento flexível. Além de calcular os tributários disponíveis no quadro ODU de alta ordem para multiplexação de sinais clientes, o controlador SDN também deve se preocupar com a multiplexação por comprimento de onda. O espaçamento fixo no espectro óptico simplifica razoavelmente o cálculo do provisionamento de serviços, ao contrário do espaçamento flexível que abre mais possibilidades de rotas aumentando o custo computacional. Em adição ao argumento anterior, o controlador SDN também deve considerar a potência de cada canal sendo transmitido na fibra óptica de forma a evitar a interferência e efeitos não lineares.

Outro ponto importante para futuros estudos é o desenvolvimento e implementação da fase 2 e 3 do projeto ODTN. Apesar que já foi apresentado alguns experimentos em topologias reais demonstrando a desagregação inicial dos equipamentos DWDM, ainda falta muita incerteza e implementação no controlador SDN. Por exemplo, não está definido como os alarmes e métricas ópticas serão definitivamente tratadas pelo controlador, bem como, a falta de aplicações mais robustas para o controle da topologia. Isso de certo modo dificulta a adesão da arquitetura de uma rede desagrega de transporte pelos provedores de telecomunicações.

Referências

- [1] M. Carroll, J. Roese and T. Ohara, “The Operator's View of OTN Evolution,” *IEEE Communications Magazine*, vol. 48, pp. 46-52, Set. 2010.
- [2] S. Haykin, *Sistemas de Comunicação Analógicos e Digitais*, 4ª Edição, Bookman, 2004.
- [3] G. P. Agrawal, “Optical Communication: Its History and Recent Progress,” em *Optics in Our Time*, Springer, Cham., 2016.
- [4] ITU-T Recommendations, “Guide on the use of the ITU-T Recommendations related to optical fibres and systems technology,” *Series G: Transmission System and Media, Digital Systems and Networks*, Out. 2018.
- [5] C. C. Davis and T. E. Murphy, “Fiber-Optic Communications,” *IEEE Signal Processing Magazine*, vol. 28, nº 4, pp. 152-150, Jul. 2011.
- [6] Cisco Systems, Inc., “Introduction to DWDM Technology,” Text Part Number: OL-0884-01, 2000.
- [7] ITU-T Recommendations, “G.652: Characteristics of a single-mode optical fibre,” *Series G: Transmission System and Media, Digital Systems and Networks*, Nov. 2016.
- [8] K. Kikuchi, “Fundamentals of Coherent Optical Fiber Communications,” *Journal of Lightwave Technology*, vol. 34, nº 1, pp. 157-179, Jan. 2016.
- [9] ITU-T Recommendations, “G.694.1: Spectral grids for WDM applications: DWDM frequency grid,” *Series G: Transmission System and Media, Digital Systems and Networks*, Fev. 2012.
- [10] ITU-T Recommendations, “G.653: Characteristics of a dispersion-shifted, single-mode optical fibre and cable,” *Series G: Transmission System and Media, Digital Systems and Networks*, Jul. 2010.
- [11] Y. H. Choe and V. Mauree, “The Optical World,” *ITU-T Technology Watch Report*, Jun. 2011.

- [12] ITU-T Recommendations, “G.709: Interfaces for the optical transport network,” *Series G: Transmission System and Media, Digital Systems and Networks*, Jun. 2020.
- [13] ITU-T Recommendations, “G.872: Architecture of optical transport networks,” *Series G: Transmission System and Media, Digital Systems and Networks*, Dez. 2019.
- [14] A. K. Garg and R. Vashishath, “Comparative Analysis of Nonlinear Transmission Impairments in WDM Optical Network,” *International Journal of Engineering and Innovative Technology (IJEIT)*, vol. 3, n° 11, pp. 165-172, Maio 2014.
- [15] ITU-T Recommendations, “G.655: Characteristics of a non-zero dispersion-shifted single-mode optical fibre and cable,” *Series G: Transmission System and Media, Digital Systems and Networks*, Nov. 2009.
- [16] E. Pincemin et. al, “Challenges of 40/100 Gbps and higher-rate deployments over long-haul,” *Optical Fiber Technology*, vol. 17, n° 5, pp. 335-362, Ago. 2011.
- [17] CFP MSA, “CFP MSA Hardware Specification,” Jun. 2010. [Online]. Available: <http://www.cfp-msa.org/Documents/CFP-MSA-HW-Spec-rev1-40.pdf>. [Acesso em 2020].
- [18] SFF Committee, “Specification for SFP+ 10 Gb/s and Low Speed Electrical Interface,” Set. 2013. [Online]. Available: <https://www.fluxlight.com/content/Tech-Docs/SFPPlus%20MSA.PDF>. [Acesso em 2020].
- [19] E. P. da Silva et al, “450 Gb/s CO-OFDM DP-QPSK superchannels for long-haul transmission with High Spectral Efficiency,” *SBMO/IEEE MTT-S International Microwave & Optoelectronics Conference (IMOC)*, pp. 1-3, 2013.
- [20] ITU-T Recommendations, “M.3348: Requirements of the NMS-EMS management interface for NGN service platforms,” *Series M: Telecommunication Management, including TMN and Network Maintenance*, Jan. 2011.
- [21] Open ROADM MSA, “OpenROADM - Optical Specs, Data Models and Documentation,” [Online]. Available: <http://openroadm.org/download.html>. [Acesso em Out. 2020].
- [22] G. Pujolle, *Software Networks Virtualization, SDN, 5G and Security*, 2nd Edition, Wiley, 2020.

- [23] P. Goransson, C. Black, and T. Culver, *Software Defined Network A Comprehensive Approach*, 2nd Edition, Elsevier, 2017.
- [24] V. K. Mishra, *Software Defined Networks*, Momentum Press, 2019.
- [25] F. Hu, Q. Hao and K. Bao, "A Survey on Software-Defined Network and OpenFlow: From Concept to Implementation," *IEEE Communications Surveys & Tutorials*, vol. 16, n° 4, pp. 2181-2206, Maio 2014.
- [26] H. Kim and N. Feamster, "Improving network management with software defined networking," *IEEE Communications Magazine*, vol. 51, n° 2, pp. 114-119, Fev. 2013.
- [27] A. Lara, A. Kolasani and B. Ramamurthy, "Network Innovation using OpenFlow: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 16, n° 1, pp. 493-512, 2013.
- [28] N. Feamster, J. Rexford, and E. Zegura, "The road to SDN: an intellectual history of programmable networks," *ACM SIGCOMM Computer Communication Review*, vol. 44, n° 2, pp. 87-98, Abril 2014.
- [29] Internet Engineering Task Force (IETF), "RFC 6241 - Network Configuration Protocol (NETCONF)," *Standards Track*, Jun. 2011.
- [30] R. Alvizu et al., "Comprehensive Survey on T-SDN: Software-Defined Networking for Transport Networks," *IEEE Communications Surveys & Tutorials*, vol. 19, n° 4, pp. 2232-2283, 2017.
- [31] Internet Engineering Task Force (IETF), "RFC 6020 - YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)," *Standards Track*, Out. 2010.
- [32] R. Casellas et al., "Control, Management, and Orchestration of Optical Networks: Evolution, Trends, and Challenges," *Journal of Lightwave Technology*, vol. 36, n° 7, pp. 1390-1402, Abril 2018.
- [33] S. Badotra and J. Singh, "Open Daylight as a Controller for Software Defined Networking," *International Journal of Advanced Research in Computer Science*, vol. 8, n° 5, pp. 1105-1111, Jun. 2017.
- [34] Open Networking Foundation (ONF), "ONOS - Open Network Operating System," [Online]. Available: <https://www.opennetworking.org/onos/>. [Acesso em Out. 2020].

- [35] R. Casellas et al., “Metro-Haul: SDN Control and Orchestration of Disaggregated Optical Networks with Model-Driven Development,” *2018 20th International Conference on Transparent Optical Networks (ICTON)*, pp. 1-4, Set. 2018.
- [36] A. Giorgetti et al., “Control of open and disaggregated transport networks using the Open Network Operating System (ONOS),” *IEEE/OSA Journal of Optical Communications and Networking*, vol. 12, n° 2, pp. A171-A181, Fev. 2020.
- [37] R. Casellas et al., “Abstraction and Control of Multi-Domain Disaggregated Optical Networks With OpenROADM Device Models,” *Journal of Lightwave Technology*, vol. 38, n° 9, pp. 2606-2615, Maio 2020.
- [38] S. Rahman et al., “Virtualized controller placement for multi-domain optical transport networks using machine learning,” *Photonic Network Communications*, pp. 39-50, Jul. 2020.
- [39] A. S. Thyagaturu et al., “Software Defined Optical Networks (SDONs): A Comprehensive Survey,” *IEEE Communications Surveys & Tutorials*, vol. 18, n° 4, pp. 2738-2786, Jul. 2016.
- [40] Open Networking Foundation, “OpenFlow Specifications Files,” [Online]. Available: <https://www.opennetworking.org/software-defined-standards/specifications/>. [Acesso em Out. 2020].
- [41] Open Networking Foundation and Optical Internetworking Forum, “Global Transport SDN Prototype Demonstration,” Out. 2014. [Online]. Available: https://www.opennetworking.org/wp-content/uploads/2013/02/oif-p0105_031_18.pdf. [Acesso em Out. 2020].
- [42] N. F. de Sousa and et al., “Network Service Orchestration: A survey,” *Computer Communications*, Vols. %1 de %2142-143, pp. 69-94, Jun. 2019.
- [43] A. Sgambelluri et al., “OpenROADM-Controlled White Box Encompassing Silicon Photonics Integrated Reconfigurable Switch Matrix,” *Optical Fiber Communications Conference and Exhibition (OFC)*, pp. 1-3, Mar. 2020.
- [44] OpenConfig, “OpenConfig FAQ,” [Online]. Available: <https://www.openconfig.net/docs/faq/>.
- [45] Open Networking Foundation, “TR-547 - TAPI v2.1.3 Reference Implementation Agreement,” Jul. 2020. [Online]. Available: <https://www.opennetworking.org/wp->

content/uploads/2020/08/TR-547-TAPI-v2.1.3-Reference-Implementation-Agreement-1.pdf. [Acesso em 2020 Out].

- [46] R. Vilalta et. al, “Network Programmability and Automation in Optical Networks,” *Optical Network Design and Modeling (ONDM)*, vol. 11616, pp. 223-234, 2019.
- [47] Open Networking Foundation, “Open and Disaggregated Transport Network (ODTN),” [Online]. Available: <https://www.opennetworking.org/odtn/>. [Acesso em Out. 2020].
- [48] M. Leenheer and Y. Higuchi, “An open controller for the disaggregated optical network,” *International Conference on Optical Network Design and Modeling (ONDM)*, pp. 230-233, Maio 2018.
- [49] A. Campanella et al., “ODTN: Open Disaggregated Transport Network. Discovery and Control of a Disaggregated Optical Network through Open Source Software and Open APIs,” *2019 Optical Fiber Communications Conference and Exhibition (OFC)*, pp. 1-3, 2019.
- [50] A. Sgambelluri et al., “Fully Disaggregated ROADM White Box with NETCONF/YANG Control, Telemetry, and Machine Learning-based Monitoring,” *2018 Optical Fiber Communications Conference and Exposition (OFC)*, pp. 1-3, Mar. 2018.
- [51] E. C. Magalhães, “Equalização Global do Espectro de Canais Amplificados durante a Transmissão em uma Rede Óptica com difernete número de Nós,” 2015. [Online]. Available: http://repositorio.unicamp.br/bitstream/REPOSIP/260616/1/Magalhaes_EduardoCavalcanti_D.pdf. [Acesso em Novembro 2020].
- [52] Tail-f, a Cisco company, “ConfD Basic, NETCONF and YANG for Free!,” [Online]. Available: <https://www.tail-f.com/confd-basic/>. [Acesso em Nov. 2020].
- [53] Canonical Ltd. Ubuntu, “Ubuntu 18.04.5 LTS (Bionic Beaver),” [Online]. Available: <https://releases.ubuntu.com/18.04/>. [Acesso em Nov. 2020].
- [54] Oracle, “Virtual Box,” [Online]. Available: <https://www.virtualbox.org/>. [Acesso em Nov. 2020].
- [55] Open ROADM MSA, “OpenROADM models repository,” Oct. 2020. [Online]. Available: https://github.com/OpenROADM/OpenROADM_MSA_Public. [Acesso em Nov. 2020].

- [56] Open ROADM MSA, “OpenROADM v2 Network Model Whitepaper,” Oct 2018. [Online]. Available: https://0201.nccdn.net/4_2/000/000/06b/a1b/Open-ROADM-Network-Model-Whitepaper-v2-2-1.pdf. [Acesso em Nov 2020].
- [57] Open ROADM MSA, “ROADM Network Model and Device Model - An Example,” Jun 2016. [Online]. Available: https://0201.nccdn.net/1_2/000/000/173/842/Open-ROADM-Models-whitepaper---An-Example-v1-0.pdf. [Acesso em Nov 2020].
- [58] International Electrotechnical Commission, “IEC 60297-3-100:2008 - Mechanical structures for electronic equipment - Dimensions of mechanical structures of the 482,6 mm (19 in) series - Part 3-100: Basic dimensions of front panels, subracks, chassis, racks and cabinets,” Nov. 2008. [Online]. Available: <https://webstore.iec.ch/publication/1283>. [Acesso em Nov. 2020].
- [59] Open ROADM MSA, “OpenROADM v2 Device Whitepaper,” Aug 2018. [Online]. Available: https://0201.nccdn.net/1_2/000/000/134/c50/Open-ROADM-MSA-release-2-Device-White-paper-v1-1.pdf. [Acesso em Nov 2020].