Ambiente virtual de simulação e controle de redes ópticas baseado em softwares de código aberto e modelos OpenROADM

Niudomar S. A. Chaves[1], **Luciano Martins**[1]

[1]Centro de Pesquisa e Desenvolvimento em Telecomunicações (CPQD) – Campinas, SP – Brasil

{nchaves,lmartins}@cpqd.com.br

Abstract. *This article presents a proposal for a virtual environment to control optical networks composed of open source software, including an SDN controller and a simulator of optical transceivers and ROADMs. The virtual environment components use OpenROADM standard YANG data models. OpenDaylight is used as an SDN controller, complemented by the TransportPCE optical network control module. Optical equipment is simulated by Honeynode software. The proposed solution allows to previously test optical network control algorithms in a virtual environment before the effective application in a physical testbed, which usually have a restricted time allocation.*

Resumo. *Este artigo apresenta uma proposta de ambiente virtual para controle de redes ópticas composto por softwares de código aberto, incluindo um controlador SDN e um simulador de transceptores ópticos e ROADMs. Os componentes do ambiente virtual utilizam modelos de dados YANG do padrão OpenROADM. Como controlador SDN é utilizado o OpenDaylight, complementado pelo módulo de controle de redes ópticas TransportPCE. Os equipamentos ópticos são simulados pelo software Honeynode. A solução proposta permite testar previamente algoritmos de controle de rede óptica em ambiente virtual antes da efetiva aplicação em um testbed físico, que usualmente possuem alocação de tempo restrito.*

1. Introdução

A pesquisa e o desenvolvimento em SDN (*Software Defined Network*) em redes ópticas frequentemente são impactados negativamente pela dificuldade de acesso a *testbeds* ou plataformas de experimentação com topologia e equipamentos que se aproximem do cenário real de uma rede óptica DWDM (*Dense Wavelength-Division Multiplexing*) comercial. Em especial, devido ao alto custo, o acesso a equipamentos ROADM (*Reconfigurable Optical Add-Drop Multiplexer*) costuma ser o mais difícil. Por essa razão, alguns países optam por concentrar o investimento na construção de um *testbed* com infraestrutura mais diversificada e significativa numa determinada localidade, mas que é acessível remotamente por instituições de pesquisa de todo o país, como é o caso do *testbed* COSMOS (***C****loud enhanced* ***O****pen* ***S****oftware defined* ***MO****bile wireless testbed for city-****S****cale deployment*), na cidade de Nova Iorque [COSMOS 2022]. O COSMOS tem por objetivo promover pesquisa e experimentação em redes sem fio com integração com computação em nuvem na borda da rede e possui uma rede de transporte óptica composta por transceptores e ROADMs, bem como por chaves ópticas que permitem alterar a topologia

da rede por comandos remotos [Dipankar Raychaudhuri et al. 2020]. No Brasil, a RNP (Rede Nacional de Ensino e Pesquisa) coloca à disposição da comunidade acadêmica a infraestrutura do *testbed* SDN Multicamadas, localizado no Rio de Janeiro. Esse *testbed* possui *transponders* Cassini e *switches* programáveis por P4, mas não possui, atualmente, ROADMs na sua infraestrutura. Os autores deste trabalho desconhecem a existência de algum *testbed* físico no Brasil, disponível para a comunidade acadêmica, que possua ROADMs.

Este artigo apresenta uma proposta de um conjunto de ferramentas de software de código aberto e de uso gratuito que permitem criar e controlar um ambiente virtual que simula uma rede óptica DWDM composta por transceptores e ROADMs. O ambiente proposto utiliza os modelos YANG de dispositivos ópticos padronizados pelo consórcio OpenROADM Multi-Source Agreement (MSA) [OpenROADM MSA 2022]. A proposta permite o desenvolvimento, teste e depuração prévios de algoritmos de controle de redes ópticas baseadas em modelos YANG OpenROADM em ambiente simulado, antes da efetiva aplicação em redes ou *testbed* físicos. O teste prévio em ambiente simulado pode proporcionar maior efetividade nos experimentos, uma vez que o acesso a *testbed* costuma ocorrer com alocação de períodos de tempo limitados. Espera-se que os algoritmos desenvolvidos e testados no ambiente proposto funcionem sem alterações em qualquer *testbed* composto por equipamentos que adotem o padrão OpenROADM, desde que seja utilizado o controlador TransportPCE/OpenDaylight, uma vez que os algoritmos utilizam a API (*Application Programming Interface*) desse controlador. Mesmo na indisponibilidade de acesso a *testbed* físicos, o ambiente proposto oferece o benefício de permitir a universalização da realização de atividades de ensino, pesquisa e desenvolvimento de algoritmos de controle de rede óptica, por não haver custo de licenciamento de software ou aquisição de caros equipamentos ópticos, sendo necessário apenas capacidade computacional para executar os softwares necessários em ambiente Linux.

Como controlador SDN é utilizado o OpenDaylight [Networking 2022a], complementado pelo módulo de controle de rede óptica TransportPCE [Networking 2022b]. A simulação dos transceptores ópticos e ROADMs é realizada pelo software Honeynode [Orange 2022a]. Adicionalmente, é utilizado o software TPCE_GUI [Orange 2022b], que provê uma interface gráfica de usuário para visualização da rede controlada pelo TransportPCE.

Na seção 2 são apresentados os padrões de modelos de dados YANG dos consórcios OpenROADM e OpenConfig. Os diferentes componentes de software da solução proposta são descritos na seção 3. A seção 4 contém a descrição do ambiente virtual de experimentação proposto, apresentando detalhes da sequência de etapas na execução de uma configuração de circuito óptico. Por fim, na seção 5, são apresentados os possíveis benefícios e aplicações do ambiente proposto, bem como ideias para a evolução deste trabalho.

2. Modelos OpenROADM e OpenConfig

Tradicionalmente, uma rede DWDM precisa ser implantada com equipamentos de um único fornecedor para se obter completa interoperabilidade. Essa situação restringe a competição e desacelera o ritmo de inovação. Para atacar esses problemas, diversos fabricantes de equipamentos de rede óptica e operadoras de redes de telecomunicações

decidiram se agrupar num consórcio para definir padrões de modelos de equipamentos e serviços de redes ópticas baseados na linguagem de modelagem de dados YANG, para que sejam controlados no paradigma SDN. Esse consórcio, chamado de OpenROADM MSA, tem desenvolvido e atualizado com regularidade esses padrões [Renais et al. 2019].

Os modelos OpenROADM são definidos utilizando a versão 1.1 da linguagem YANG, descrita na RFC 7950 do IETF. O consórcio define modelos YANG para os dispositivos ópticos, para a topologia de rede e para os serviços. O consórcio define também a especificação de uma interface óptica que opera com um único comprimento de onda, para assegurar a interoperabilidade entre dispositivos terminais (transceptores e módulos ópticos), e de uma interface que opera com múltiplos comprimentos de onda, para promover a interoperabilidade entre dispositivos intermediários (ROADMs e amplificadores de linha), rompendo a dependência de um único fornecedor. A versão mais atual dos modelos é a 10.1.0. Quatro tipos de nós ópticos são definidos:

- rdm: ROADM - os modelos suportam os modos *colorless-directionless* e *colorless-directionless-contentionless*;
- xpdr: xponder - dispositivo que converte o sinal cliente para a interface de canal óptico;
- ila: amplificador de linha;
- extplug: módulo óptico inserido num dispositivo não óptico, como um roteador, por exemplo.

A especificação da interface W (comprimento de onda único) já contempla as taxas de 100G, 200G, 300G e 400G e as modulações DP-QPSK e DP-mQAM. Especificamente, as taxas de 100G e 200G utilizam modulação QPSK, a taxa de 300G usa 8 QAM e a taxa de 400G utiliza 16 QAM.

Os modelos YANG dos dispositivos permitem a criação automática em controladores SDN de interfaces *southbound* NETCONF, que são utilizadas para administrar os equipamentos. Cada equipamento é representado dentro do controlador por uma instância em JSON ou XML que segue o modelo do equipamento descrito em YANG. Essa interface padronizada, guiada por modelo de dados, permite o controle de qualquer equipamento, de qualquer fabricante, que siga o modelo YANG OpenROADM sem necessidade de alteração de código.

Os modelos YANG de serviço são utilizados para criar uma interface *northbound* no controlador, normalmente no padrão RESTCONF, descrito na RFC 8040 do IETF. Essa API permite que aplicações ou orquestradores solicitem, excluam ou alterem serviços de conectividade óptica. Os detalhes de configuração dos equipamentos ópticos são definidos por algoritmos implementados dentro do controlador e devem ser transparentes para a entidade que solicita o serviço.

O modelo de rede define uma topologia abstrata e independente de fabricante dentro do controlador SDN. Esse modelo é fundamental para o desenvolvimento da funcionalidade de cálculo de caminho óptico, chamada *Path Computation Element* (PCE). Esse modelo, em conjunto com a API *northbound*, permite que as aplicações de camadas superiores possam ser desenvolvidas sem qualquer dependência de modelo e marca de equipamento óptico a ser controlado, desde que siga o padrão OpenROADM. É no modelo de rede que são visíveis e gerenciáveis os graus e grupos *add/drop* dos ROADMs. A versão 1.2.1 do OpenROADM oferecia suporte apenas ao modelo de grade fixa de canais. A

partir da versão 2.2.1 já é suportado o modelo FlexGrid, com espaçamento variável de canais. Nessa versão também surgiu o suporte inicial a OTN (*Optical Transport Network*), que seguiu em evolução nas versões seguintes.

Existe um outro padrão de modelos YANG para dispositivos ópticos chamado OpenConfig [OpenConfig 2022], mas que também contempla outros domínios tecnológicos como pacotes e rádio. Esses modelos foram gerados por um consórcio liderado por operadoras de telecomunicações e grandes empresas de TI e *cloud*. Enquanto a proposta do OpenROADM é desenvolver modelos mais completos e atualizados acompanhando a evolução da tecnologia, a proposta do OpenConfig busca atender aos modelos operacionais praticados por essas operadoras e empresas de *cloud*, sem a intenção de abranger todos os recursos disponíveis nos equipamentos. Dessa forma, são modelos que tratam das funcionalidades essenciais e mais comuns, apenas. Atualmente o OpenROADM utiliza o padrão OpenConfig num caso bem específico, para telemetria por *streaming* de transceptores ópticos que não são aderentes ao padrão OpenROADM.

3. Componentes da Solução

3.1. OpenDaylight e TransportPCE

O OpenDaylight é um controlador SDN de código aberto desenvolvido de forma colaborativa em projeto hospedado na Linux Foundation Networking. Ele é capaz de controlar equipamentos e funções de rede utilizando vários protocolos na sua interface *southbound*, tais como: NETCONF, BGP-LS, PCEP, SNMP, OpenFlow e OVSDB. Uma aplicação pode fazer solicitações para o OpenDaylight por meio de sua interface *northbound* usando os protocolos IETF NETCONF ou RESTCONF. O controlador é direcionado por modelo, ou seja, os elementos de rede gerenciados, que podem ser físicos ou virtualizados, são modelados em YANG, bem como os serviços ofertados na sua API *northbound* [Networking 2022a].

O TransportPCE é um subprojeto do OpenDaylight onde é desenvolvido um módulo adicional específico para o controle de redes ópticas [Networking 2022b]. Ele funciona em conjunto com o OpenDaylight e já vem empacotado com ele. O TransportPCE é capaz de controlar equipamentos que utilizam os modelos de dados definidos pelo consórcio OpenROADM. Atualmente, o TransportPCE oferece suporte às versões 1.2.1, 2.2.1 e 7.1 dos modelos OpenROADM, com API *southbound* NETCONF e API *northbound* RESTCONF. Segundo o projeto, essas três versões foram aquelas adotadas pelos fabricantes até o momento.

O OpenDaylight e o TransportPCE não possuem GUI (*Graphical User Interface*) nativas, mas há uma interface gráfica disponível para o TransportPCE no projeto TPCE-GUI [Orange 2022b]. Com essa interface é possível ver a topologia carregada no TransportPCE, identificando os sítios, equipamentos e enlaces, mas não é possível realizar ações de configuração e controle.

3.2. Simulador Honeynode

O simulador Honeynode é capaz de emular a troca de mensagens NETCONF que equipamentos ópticos físicos teriam com um controlador SDN, de acordo com o padrão OpenROADM. O simulador é capaz de representar transceptores e ROADMs. As características técnicas de cada equipamento emulado são definidas por meio de um arquivo

de configuração no formato XML e cada instância em execução do Honeynode emula um único equipamento. Dessa forma, é possível configurar ROADMs com quantos graus e interfaces *add/drop* forem desejados, bem como várias outras características técnicas disponíveis no modelo YANG que descreve um ROADM. Com várias instâncias do simulador em execução simultânea, representando diferentes elementos de rede, é possível ter uma rede virtual respondendo a comandos NETCONF, com manutenção de estado das configurações, possibilitando a criação de circuitos ópticos nos níveis DWDM e OTN. A versão atual do Honeynode é capaz de simular equipamentos nas versões 1.2.1, 2.2.1 e 7.1 dos modelos OpenROADM [Orange 2022a], correspondendo às versões suportadas no TransportPCE.

O projeto ODTN, da Open Networking Foundation, possui um simulador do *transponder* Cassini, do Telecom Infra Project. No entanto, diferentemente do Honeynode, ele simula apenas o *transponder*, permitindo, portanto, apenas a criação de uma conexão fim a fim, sem visibilidade e controle de nenhum elemento intermediário. Por fim, o desenvolvimento desse simulador foi abandonado muito precocemente e ele possui funcionalidade bastante limitada.

4. Ambiente Virtual de Experimentação

Este trabalho propõe um ambiente virtual de experimentação em controle SDN de redes ópticas baseadas em modelos OpenROADM utilizando o controlador TransportPCE/OpenDaylight, a interface gráfica de usuário TCPE-GUI e o emulador de dispositivos ópticos Honeynode. Todos os componentes estão disponíveis com licenças *open source* e são executados sobre sistema Linux. Esta seção contém uma descrição de todas as etapas executadas para realizar o estabelecimento de um circuito óptico DWDM fim a fim. Resumidamente, essas etapas são:

1. Execução do controlador OpenDaylight (o pacote TransportPCE foi previamente habilitado no OpenDaylight);
2. Execução de uma instância do simulador Honeynode para cada elemento da topologia, composta por dois transceptores ópticos e quatro ROADMs. Cada instância representa um equipamento de acordo com uma configuração previamente realizada em arquivo XML;
3. Envio de requisição RESTCONF para instruir o OpenDaylight a se conectar a cada um dos dispositivos. Dessa forma, o controlador SDN passa conhecer os equipamentos e as conexões entre os ROADMs;
4. Envio de requisição RESTCONF para criar as conexões entre os transceptores e os ROADMs nos dois sentidos (a conexão entre os ROADMs é realizada de outra forma, explicada mais adiante);
5. Envio de requisição RESTCONF para instruir o OpenDaylight a estabelecer o circuito óptico entre os dois transceptores.

O repositório do projeto TransportPCE [Networking 2022c] contém vários recursos que podem auxiliar na montagem do cenário desejado. Há *scripts* em Python com as principais requisições em RESTCONF a serem enviadas para o TransportPCE para controlar a rede. O simulador Honeynode está incluso no repositório e há um *script* para a compilação e montagem nas versões desejadas (1.2.1, 2.2.1 e 7.1). O repositório também contém arquivos XML com a configuração de transceptores e ROADMs. Esses arquivos

podem ser alterados para adaptar as configurações dos equipamentos emulados conforme o interesse ou podem ser usados como ponto de partida para a configuração de novos elementos, permitindo a construção de topologias mais complexas. Vale ressaltar que num ROADM físico os enlaces são descobertos de forma automática por meio do protocolo LLDP executando sobre o canal de supervisão óptica. No caso do simulador Honeynode, os enlaces devem ser configurados manualmente na seção de parâmetros LLDP do arquivo XML.

A interface gráfica TPCE-GUI não está inclusa no repositório do TransportPCE. Ela precisa ser clonada separadamente e pode-se gerar uma imagem de *container* Docker para executá-la. O simulador Honeynode pode, opcionalmente, ser utilizado na forma de *containers* Docker, mas os *scripts* de teste existentes no repositório os executam diretamente, onde cada instância representa um transceptor ou ROADM.

```
starting OpenDaylight...
starting KARAF TransportPCE build...
Searching for pattern 'Blueprint\ container\ for\ bundle\ org\.opendaylight\.netconf\.restconf.* wa
s successfully created' in karaf.log... Pattern found! OpenDaylight started !
starting simulator xpdra in OpenROADM device version 2.2.1...
Searching for pattern 'Netconf SSH endpoint started successfully at 0.0.0.0' in xpdra-221.log... Pa
ttern found! simulator for xpdra started
starting simulator roadma in OpenROADM device version 2.2.1...
Searching for pattern 'Netconf SSH endpoint started successfully at 0.0.0.0' in roadma-221.log... P
attern found! simulator for roadma started
```

Figura 1. Execução do OpenDaylight/TransportPCE e do simulador Honeynode

A Figura 1 contém parte do log de saída da execução do script, onde é possível ver que o OpenDaylight foi iniciado, bem como duas instâncias do simulador Honeynode, a primeira representando o transceptor chamado “xpdra” e a segunda representando o ROADM chamado “roadma”. Cada instância do Honeynode utiliza uma porta TCP diferente, definida no comando de execução do processo. É possível ver, também, que as duas instâncias exibidas do Honeynode estão usando a versão 2.2.1 do modelo OpenROADM.

A topologia completa pode ser visualizada na Figura 2. Essa é a interface gráfica provida pelo TPCE-GUI. Pode-se ver que a topologia é composta de quatro ROADMS, conectados em anel. De acordo com o modelo OpenROADM os elementos NodeA, NodeB, NodeC e NodeD são localidades ou estações de telecom, onde é possível ter diversos equipamentos. No cenário apresentado há apenas um ROADM em cada sítio. No caso do NodeA, há um transceptor e um ROADM de grau 2. É possível configurar nos arquivos XML características técnicas dos enlaces entre ROADMs, como comprimento e atenuação, entre outros parâmetros. Na imagem exibida, o elemento ROADM-A1 está selecionado, permitindo visualizar algumas informações desse elemento na lateral direita.

Até este momento o OpenDaylight não tem conhecimento desses equipamentos. A Figura 3 exibe a saída do OpenDaylight no momento em que ele se conecta a esses equipamentos e carrega seus modelos de dados YANG populados com suas configurações atuais. Normalmente esse processo é chamado de descoberta e a partir desse momento o controlador SDN passa a ter uma representação virtual da topologia física. Apesar de ser chamado de descoberta, esse processo é disparado quando é enviada para o controlador uma requisição RESTCONF em que são fornecidos o endereço IP, porta e os dados de autenticação para que o OpenDaylight possa se conectar ao equipamento.

A conexão entre um transceptor e um ROADM precisa ser criada manualmente,

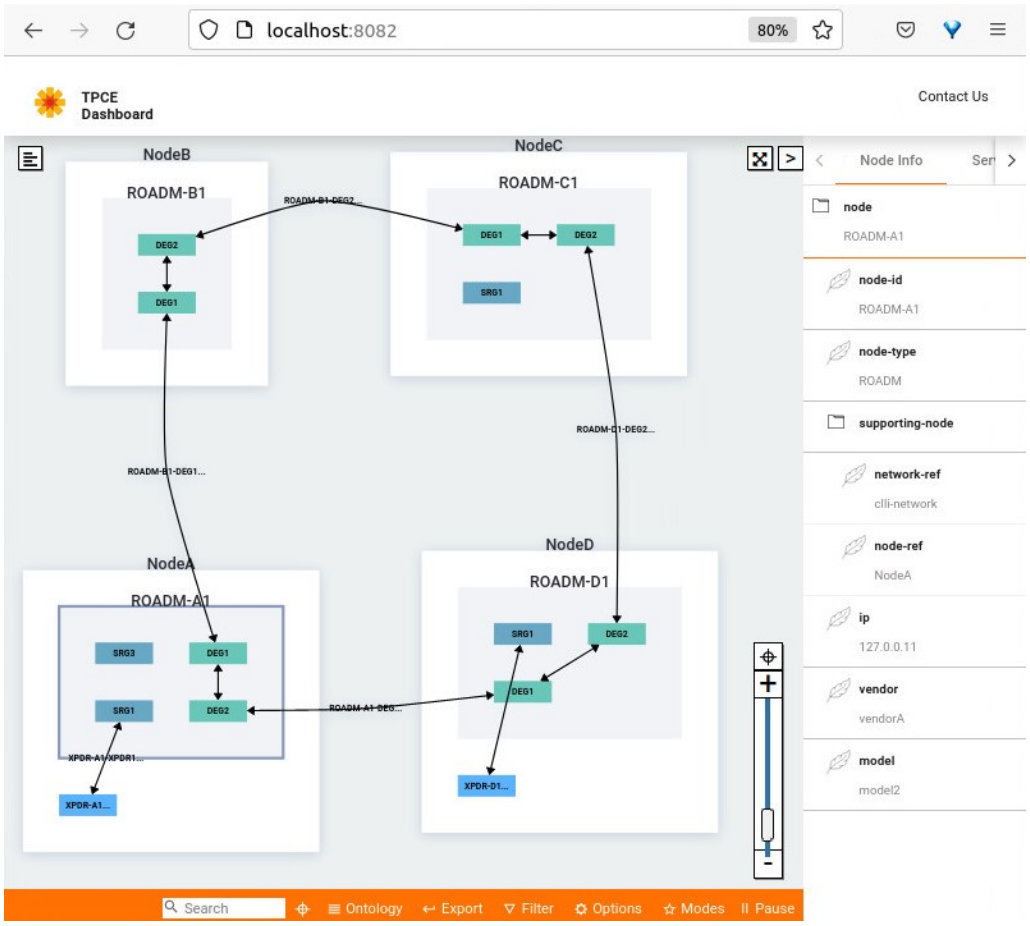

Figura 2. Topologia com quatro ROADMs

enviando uma requisição RESTCONF para o controlador. Como essa conexão é unidirecional, é preciso solicitar a conexão em cada um dos sentidos. As Figuras 4 e 5 mostram o momento em que essas conexões são criadas.

```
Searching for pattern 'Triggering\ notification\ stream\ NETCONF\ for\ node\ XPDR\-A1' in karaf.log
... Pattern found! Node XPDR-A1 correctly added to tpce topology...

Searching for pattern 'Triggering\ notification\ stream\ NETCONF\ for\ node\ ROADM\-A1' in karaf.lo
g... Pattern found! Node ROADM-A1 correctly added to tpce topology...

Searching for pattern 'Triggering\ notification\ stream\ NETCONF\ for\ node\ ROADM\-B1' in karaf.lo
g... Pattern found! Node ROADM-B1 correctly added to tpce topology...
```

Figura 3. Descoberta dos dispositivos pelo OpenDaylight

```
{"networkutils:input": {"networkutils:links-input": {"networkutils:xpdr-node": "XPDR-A1", "networku
tils:xpdr-num": "1", "networkutils:network-num": "1", "networkutils:rdm-node": "ROADM-A1", "network
utils:srg-num": "1", "networkutils:termination-point-num": "SRG1-PP1-TXRX"}}}

Conexão da porta do transponder A com o ROADM A - Status da resposta: 200

Xponder Roadm Link created successfully
```

Figura 4. Criação da conexão entre transceptor e ROADM

Uma vez que a topologia esteja pronta, é possível solicitar por meio de requisição RESTCONF a criação de um serviço de conectividade fim a fim, ou seja, entre dois trans-

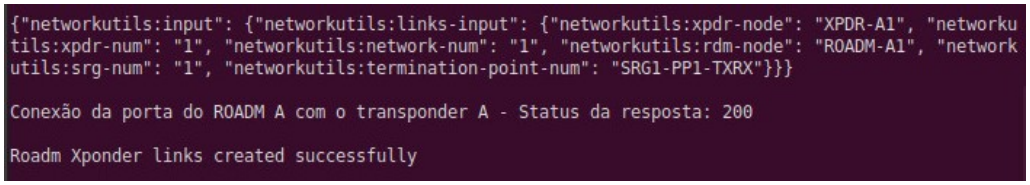

Figura 5. Criação da conexão entre ROADM e transceptor

ceptores. A Figura 6 mostra a saída do OpenDaylight no momento em que a criação da conexão é solicitada. A mensagem "PCE calculation in progress" significa que o TransportPCE está analisando a topologia para realizar a escolha do caminho. A métrica padrão para essa decisão é a menor quantidade de saltos.

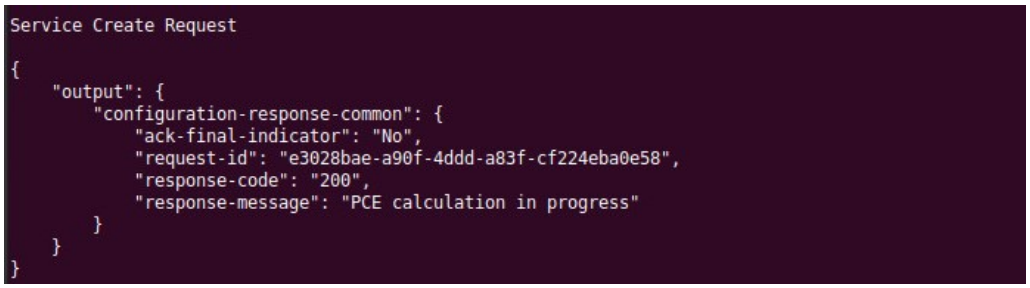

Figura 6. Criação do serviço de conectividade óptica

Após criado o serviço, os dois extremos são chamados de "a end" e "z end". A Figura 7 exibe detalhes do serviço na "a end", enquanto a Figura 8 exibe os detalhes na "z end". É possível verificar que foi criado um serviço no formato Ethernet, com taxa de 100 Gbps e que a "a end" fica no sítio NodeA, no transceptor XPDR-A1.

Figura 7. Detalhamento do serviço - Ponta A

Na Figura 8 pode-se ver que o serviço foi chamado de "service1". Todas essas informações foram anexadas na requisição RESTCONF, usando formatação JSON ou XML. Como o cenário aqui apresentado está usando a versão 2.2.1 dos modelos OpenROADM, está sendo utilizado o modelo FlexGrid. A Figura 9 apresenta a informação da

```
"service-path-name": "service1",
"service-z-end": {
    "clli": "NodeD",
    "node-id": "XPDR-D1",
    "rx-direction": {
        "port": {
            "port-device-name": "ROUTER_SNJSCAMCJT4_000000.00_00",
            "port-name": "Gigabit Ethernet_Rx.ge-1/0/0.0",
            "port-rack": "000000.00",
            "port-shelf": "00",
            "port-type": "router"
        }
    },
```


Figura 8. Detalhamento do serviço - Ponta Z

frequência inicial e da frequência final alocadas para esse canal. É definido, também, um número para identificar esse canal. Essas informações não foram fornecidas na requisição, mas são definidas pelo controlador SDN a partir do estado da rede no momento da requisição. A figura também exibe que está sendo utilizada modulação dp-qpsk, que é consequência direta da velocidade solicitada para o serviço (100 Gbps), vinculação essa definida pelo padrão OpenROADM.


```
"aToZ-max-frequency": 196.125,
"aToZ-min-frequency": 196.075,
"aToZ-wavelength-number": 1,
"modulation-format": "dp-qpsk",
"rate": 100
```


Figura 9. Detalhamento do serviço - Canal, modulação e taxa


```
"service-path-list": {
    "service-paths": [
        {
            "path-description": {
                "aToZ-direction": {
                    "aToZ": [
                        {
                            "id": "20",
                            "resource": {
                                "state": "inService",
                                "tp-id": "XPDR1-NETWORK1",
                                "tp-node-id": "XPDR-D1-XPDR1"
                            }
                        },
                        {
                            "id": "10",
                            "resource": {
                                "state": "inService",
                                "tp-id": "DEG2-TTP-TXRX",
                                "tp-node-id": "ROADM-A1-DEG2"
                            }
                        },
                        {
                            "id": "21",
                            "resource": {
                                "node-id": "XPDR-D1-XPDR1",
                                "state": "inService"
                            }
                        },
                        {
                            "id": "11",
                            "resource": {
                                "link-id": "ROADM-A1-DEG2-DEG2-TTP-TXRXtoROADM-D1-DEG1-DEG1-TTP-TXRX",
                                "state": "inService"
                            }
                        },
```


Figura 10. Detalhamento do caminho do serviço

O TPCE-GUI tem a proposta de exibir uma linha no desenho da topologia para indicar a conexão fim a fim estabelecida. No entanto, isso só funcionou num cenário com

uma topologia simples, com apenas dois sítios e dois ROADMs, sem opção de caminho. Na topologia aqui apresentada, foi necessário solicitar ao controlador SDN OpenDaylight o detalhamento do caminho escolhido para o serviço. Parte desse detalhamento é exibido na Figura 10. Contudo, identificar o caminho por esse detalhamento exige uma análise um pouco mais cuidadosa e demorada do que era de se esperar, por dois motivos. O primeiro motivo é que muitos pontos intermediários do caminho correspondem a pontos de conexão internos aos elementos e que são definidos pelos modelos OpenROADM. O conhecimento de como esses elementos de rede são modelados no OpenROADM ajuda nessa análise. O segundo motivo é que os elementos não são exibidos na mesma sequência em que eles estão posicionados no caminho da conexão.

5. Conclusões e trabalhos futuros

Este artigo apresentou uma proposta de ambiente virtual de experimentação em redes ópticas controladas no modelo SDN utilizando os modelos de dados OpenROADM e componentes de software *open source*. Todo o ambiente é executado numa única máquina com sistema operacional Linux. Essa proposta permite o desenvolvimento, o teste e a depuração de algoritmos de controle de redes ópticas compostas por equipamentos que usem os modelos OpenROADM, controlados pelo OpenDaylight, inicialmente numa ambiente virtual, evitando consumir janelas de tempo alocadas para acesso a *testbeds* físicos, uma vez que, mesmo que haja acesso a algum *testbed* dentro ou fora do país, esses ambientes costumam ser concorridos e precisam ser compartilhados por muitas equipes diferentes. Mesmo nesse cenário, o ambiente aqui proposto pode permitir a otimização do uso do *testbed* físico, de forma que os desenvolvimentos e testes iniciais podem ser realizados em ambiente simulado, adiando o uso do *testbed* físico até que o experimento esteja suficientemente depurado e maduro. Na educação, o ambiente também apresenta a grande vantagem de democratizar o processo de ensino e aprendizagem, permitindo o estudo e experimentação prática de conceitos e protocolos SDN sem depender de infraestruturas laboratoriais caras.

Como evolução deste trabalho, é possível explorar o uso de outras métricas para escolha de caminho. Apesar do projeto informar que já há suporte para escolha de caminho pela menor latência, esse recurso não funcionou nos testes realizados. Como o TransportPCE e o OpenDaylight são projetos de código aberto, é possível desenvolver e agregar novas funcionalidades ao controlador. Pode-se também explorar o estabelecimento de conexões OTN, já disponíveis, mas não contempladas neste trabalho. Por fim, um interessante campo de investigação é em mecanismos de proteção e recuperação de falhas.

Referências

COSMOS (2022). Página do projeto COSMOS. `https://www.cosmos-lab.org/`.

Dipankar Raychaudhuri et al. (2020). Challenge: Cosmos: A city-scale programmable testbed for experimentation with advanced wireless. `https://wimnet.ee.columbia.edu/wp-content/uploads/2020/02/MobiCom2020_COSMOS.pdf`.

Networking, L. F. (2022a). OpenDaylight. `www.opendaylight.org`.

Networking, L. F. (2022b). TransportPCE. https://wiki.opendaylight.org/display/ODL/TransportPCE.

Networking, L. F. (2022c). TransportPCE repository. https://git.opendaylight.org/gerrit/admin/repos/transportpce.

OpenConfig (2022). OpenConfig. https://www.openconfig.net/.

OpenROADM MSA (2022). OpenROADM. http://www.openroadm.org.

Orange (2022a). Simulador honeynode. https://gitlab.com/Orange-OpenSource/lfn/odl/honeynode-simulator.

Orange (2022b). TPCE_GUI. https://gitlab.com/Orange-OpenSource/lfn/odl/tpce_gui.

Renais, O., Lambert, G., Betoule, C., Thouenon, G., Triki, A., Bhardwaj, D., Vachhani, S., Padi, N., and Tse1, S. (2019). The openroadm initiative [invited]. *Journal of Optical Communications and Networking*.