Aplicação *Northbound* baseada em *Transport* API (TAPI) usando Controlador ONOS e *transponder* configurável de padrão aberto para a criação de circuito óptico fim a fim

Nadia A. Nassif[1], Luciano Martins[1], Luciano A. Domingos[1], Fernando N. N. Farias[2], José L. S. Lopes[1], Lucas Borges[3], Niudomar S. A. Chaves[1], Marcos Schwarz[2], Rafael C. Figueiredo[1]

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

[2]Rede Nacional de Pesquisa (RNP) – Campinas, SP – Brasil

[3]Grupo de Pesquisa em Redes de Computadores e Comunicação Multimídia (GERCOM)
Universidade Federal do Pará (UFPA) – Belém, PA – Brasil

{nadia,lmartins,domingos,joseluis,nchaves,rafaelcf}@cpqd.com.br

{fernando.farias,marcos.schwarz}@rnp.br,lucas.borges@itec.ufpa.br

Abstract. *This paper describes an Optical Network Control Architecture that provides end-to-end optical circuit establishment. For this purpose CPQD developed a northbound application that is based on Transport API (TAPI) and RESTCONF to communicate with ONOS controller to request services, including provisioning and deprovisioning of optical circuits. The tests were performed using an optical transponder emulator developed by ODTN Project from ONF and an extension developed by RNP to include a data plane. The application development considers SDN and disaggregation concepts and enables, through an emulated environment, previous analysis of possible future work to be tested on testbeds with real equipment.*

Resumo. *Este artigo descreve uma arquitetura para controle de redes ópticas que possibilita o aprovisionamento de circuitos ópticos fim a fim. Para isso, foi desenvolvida uma aplicação northbound pelo CPQD que utiliza Transport API (TAPI) e RESTCONF para interagir com o controlador ONOS e solicitar serviços, incluindo o aprovisionamento e desaprovisionamento de circuitos ópticos. Os testes foram realizados utilizando um emulador de transponders ópticos, desenvolvido pelo projeto ODTN do ONF, com uma extensão, desenvolvida pela RNP, para a inclusão do plano de dados. O desenvolvimento desta aplicação engloba os conceitos de SDN e desagregação e permite, através de um ambiente emulado, uma análise prévia de possíveis trabalhos futuros a serem realizados em testbeds com equipamentos físicos.*

1. Introdução

Nos últimos anos, instituições como a ONF[1] (*Open Networking Foundation*) têm trabalhado na desagregação dos equipamentos de rede. Na área de redes ópticas, o projeto ODTN (*Open Disaggregated Transport Network*) [ODTN 2021] do ONF vem se

[1]Organização ONF https://opennetworking.org/

esforçando para padronizar o conceito de desagregação em equipamentos da rede de transporte. Esta desagregação promove a saída de um modelo verticalizado, com soluções proprietárias, para um modelo que utiliza modelos de dados comuns e padrões abertos para as interfaces, facilitando a gerência e manutenção e diminuindo os custos de OPEX (*Operational expenditure*) e CAPEX (*Capital Expenditure*).

A desagregação e a utilização de padrões abertos trazem grandes benefícios para as operadoras, tais como: eliminação da dependência de um único fornecedor com soluções e APIs (*Application Programming Interfaces*) proprietárias (modelo verticalizado); rápida introdução de novos serviços na rede de produção; maior agilidade na implantação de inovações, sendo necessário trocar somente os elementos a serem inovados e não toda a cadeia do sistema de linha de um único fornecedor; melhor ciclo de vida dos equipamentos ópticos, sendo realizada a substituição de cada elemento de acordo com a sua vida útil individual. Além disso, permite a utilização de um modelo de controle de redes centralizado, por meio de um controlador que executa as funcionalidades de monitoração e configuração de elementos ópticos de forma coordenada, uma vez que o mesmo tem a visibilidade de toda a topologia da rede. Este conceito é conhecido como controle de Redes Definidas por Software (*Software-Defined Network - SDN*)[Kreutz et al. 2014]. Atualmente, as SDNs estão em sua segunda geração também conhecida como NG-SDN (*Next Generation Software-Defined Network*) [NG-SDN 2021] que tem como ponto forte a atualização dos protocolos de controle e maior integração entre as redes de acesso, borda e transporte.

Considerando as tendências de NG-SDN e desagregação, este artigo apresenta a implementação de uma arquitetura de controle de rede óptica onde podem ser testadas e avaliadas soluções de controle de código aberto para a monitoração e configuração de elementos ópticos programáveis. O grupo ODTN do ONF tem trabalhado em soluções para a implementação de controle de elementos ópticos utilizando o controlador ONOS. Várias demonstrações já foram realizadas com equipamentos *transponder* por diferentes fornecedores com participação de operadoras. Neste sentido, considerando o alto custo para aquisição de equipamentos ópticos, foi escolhida a utilização de um emulador de *transponder* configurável desenvolvido pelo projeto ODTN, juntamente com uma extensão, desenvolvida pela RNP, para a inclusão, nesse emulador, do plano de dados, conforme detalhado na Seção 2.

Através de uma aplicação *northbound* foi possível realizar a comunicação com o controlador ONOS para solicitar serviços de descoberta da topologia e de aprovisionamento e desaprovisionamento de circuitos ópticos fim a fim. A aplicação desenvolvida para a arquitetura proposta serve como base para etapas seguintes de validação em equipamentos físicos. Por meio desta aplicação será possível alocar e desalocar caminhos ópticos de forma automatizada e flexível, sendo este um requisito fundamental para as futuras redes ópticas elásticas [Tomkos et al. 2012].

O objetivo principal do trabalho é avaliar e testar a configuração de serviços na camada óptica por meio de uma aplicação de alto nível (na interface *northbound*) comunicando-se com o controlador ONOS, e este, por sua vez, comunicando-se com os *transponders* emulados (na interface *southbound*) usando padrões e modelos de dados abertos para estabelecer um circuito óptico fim a fim.

Além desta seção introdutória, o artigo está organizado da seguinte forma: a Seção 2 apresenta a especificação da arquitetura de controle de redes ópticas usada neste trabalho; a Seção 3 apresenta as ferramentas, protocolos e modelos usados para a implementação da aplicação para aprovisionamento de circuitos ópticos, descrição da Plataforma SDN Multicamada, detalhamento do emulador utilizado e do ambiente de testes; a Seção 4 apresenta os resultados obtidos; a Seção 5 apresenta vantagens e desvantagens do uso de emulação e do laboratório físico do testbed SDN Multicamadas, e, por fim, a Seção 6 apresenta as conclusões e indica trabalhos futuros.

2. Especificação da Arquitetura de Controle de Redes Ópticas

Esta seção apresenta a especificação da arquitetura de controle de redes ópticas, considerando a aplicação *northbound* desenvolvida pelo CPQD, a utilização do controlador ONOS e dos emuladores de *transponders* do ODTN com extensão de plano de dados, conforme apresentado na Figura 1.

O emulador considera o *transponder* Cassini[2] (Cassini AS7716-24SC), que é um *transponder* de padrão aberto *whitebox*, desenvolvido pela EdgeCore Networks em colaboração às atividades desenvolvidas no grupo OOPT (*Open Optical & Packet Transport*) do TIP (*Telecom Infra Project*). O Cassini foi projetado para suportar uma variedade de interfaces ópticas flexíveis, oferece uma combinação de portas 100 Gigabit Ethernet (GbE), interfaces ópticas 100/200 Gbps e permite uma vazão de 3.2 Tbps.

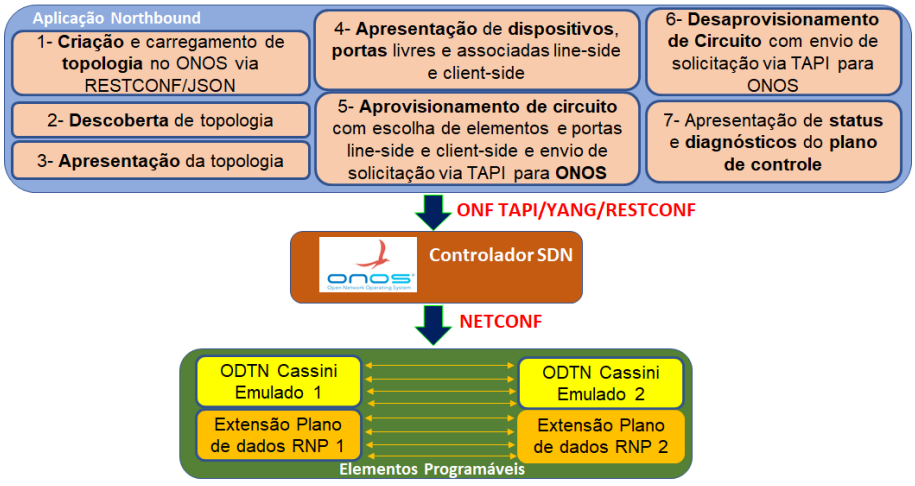

Figura 1. Arquitetura de controle de redes ópticas.

Na Figura 1, a caixa superior nomeada "Aplicação *Northbound*" apresenta as funcionalidades para controle e configuração dos emuladores Cassini. Esta aplicação se comunica com o controlador ONOS por meio de TAPI/RESTCONF (*Transport API/REpresentational State Transfer Config*) ou apenas pelo protocolo RESTCONF [Bierman et al. 2017]. O controlador ONOS, representado na Figura 1 pela caixa central, recebe as requisições da aplicação *northbound* e envia comandos NETCONF para os emuladores Cassini. Na mesma figura, tem-se a representação de dois *transponders*

[2]Edgecore Transponder Cassini: `https://www.edge-core.com/productsList.php?cls=291&cls2=347`

Cassini emulados baseados nas extensões desenvolvidas pela RNP[3] contendo a inclusão de um plano de dados.

A seguir são descritas as funcionalidades especificadas para o desenvolvimento da aplicação *northbound* da arquitetura de controle de redes ópticas, destacadas na Figura 1 e numeradas de 1 a 7:

1. **Criação e carregamento de topologia no ONOS via RESTCONF/JSON** - permite que a aplicação *northbound* envie para o ONOS, por meio de arquivos JSON (*JavaScript Object Notation*) e do protocolo RESTCONF, a topologia a ser criada com os emuladores Cassini, definindo os links físicos a serem criados entre os mesmos;
2. **Descoberta de topologia** - prevê a descoberta da topologia dos elementos ópticos emulados Cassini. Os mesmos podem já ter sido carregados no ONOS sem utilizar a aplicação, como por exemplo, por *scripts* em Linux;
3. **Apresentação da topologia** - permite que a aplicação identifique os links configurados entre os Cassini emulados e informações adicionais como portas *line-side*, portas *client-side*, velocidades e tipos, endereço IP, fabricante do equipamento, tipo do elemento óptico e sistema operacional utilizado;
4. **Apresentação de dispositivos, portas livres e associadas - *line-side e client-side*** - permite a visualização de todas as portas cliente e de linha de cada um dos Cassini emulados. Indica de forma visual se as portas estão sendo utilizadas na composição de algum circuito óptico fim a fim;
5. **Aprovisionamento de circuito com escolha de elementos e portas *line-side* e *client-side* e envio de solicitação via TAPI para ONOS** - permite que o usuário solicite a criação de um circuito óptico fim a fim, composto por um circuito óptico de linha entre os dois Cassini e associação de portas clientes. Após a solicitação pelo usuário, a aplicação envia, via TAPI/RESTCONF, um pedido de aprovisionamento para o ONOS;
6. **Desaprovisionamento de circuito com envio de solicitação via TAPI para ONOS** - permite que o usuário possa escolher um circuito fim a fim aprovisionado anteriormente e solicitar que o mesmo seja desfeito, liberando as respectivas portas de linha e de cliente associadas ao circuito escolhido.
7. **Apresentação de status e diagnóstico do plano de controle** - permite que a aplicação *northbound* solicite ao ONOS, via métodos RESTCONF, informações de status e estatísticas sobre as configurações realizadas nos Cassini emulados e do próprio ONOS.

3. Metodologia

Nesta seção, descreve-se a metodologia utilizada para implementação e teste incluindo as ferramentas, protocolos e modelos, descrição da Plataforma SDN, detalhamento do emulador ODTN/ONF com a extensão do plano de dados e do ambiente de teste usados neste trabalho.

[3]Projeto Transponder Cassini Emulado RNP: `https://git.rnp.br/cnar/sdn-multicamada/emulacao/transponder-cassini.git`

3.1. Controlador ONOS, TAPI e RESTCONF (*northbound*)

A versão do controlador ONOS utilizada foi a *Velociraptor* (2.5.1) lançada em janeiro de 2021 [4]. Esta versão suporta a versão 2.1.1 do TAPI (de dez/2018), sendo que a mais atual, no momento da escrita deste artigo, é a versão 2.1.3 (de jun/2020) [ONF-TR147 2020].

A *release* utilizada do TAPI suporta interfaces agnósticas à tecnologia para os módulos funcionais e interfaces de tecnologias específicas, mostrados na Figura 2 [ONF-TAPI 2021]. Para o desenvolvimento da aplicação *northbound* foram utilizados os módulos funcionais *Topology e Connectivity* e o perfil de interface usado foi o *Photonic Media (L0-WDM)*. Além dos modelos YANG listados, há o modelo *tapi-common@2018-12-10*, que contém definições comuns a todos os outros modelos.

Serviço	Topology	Connectivity	OAM	Path Computation	Virtual Network	Notification
Modelo YANG (@2018-12-10)	tapi-topology	tapi-connectivity	tapi-oam	tapi-path-computation	tapi-virtual-network	tapi-notification

Interfaces	Carrier Ethernet (L2)	Optical Transport Network (L1-ODU)	Photonic Media (L0-WDM)
Modelo YANG (@2018-12-10)	tapi-eth	tapi-odu	tapi-photonic-media, tapi-dsr

Figura 2. Módulos funcionais e perfis de interfaces do TAPI com os modelos YANG do ONOS.

A partir do método GET, disponível no protocolo RESTCONF, foi possível extrair as informações do ONOS. Já a operação para solicitar configuração foi feita pelo método POST e para deletar itens foi utilizado o método DELETE. Estas opções usam em suas chamadas as estruturas (*tree*) dos modelos YANG listados na Figura 2. Exemplos de operações utilizadas para o desenvolvimento da aplicação são apresentados na Figura 3. Para testes com as chamadas TAPI e a verificação das respostas que o ONOS envia, foram utilizadas as ferramentas SoapUI[5] 5.4.0 e a Postman[6] 8.0.3.

Operação da aplicação	Operação T-API/RESTCONF
Leitura da portas dos devices (RESTCONF)	*GET http://SERVER:PORT/onos/v1/devices/DEVICEID/ports*
Criação da conectividade entre interfaces line/client (T-API/RESTCONF)	*POST http://SERVER:PORT/onos/restconf/operations/tapi-connectivity:create-connectivity-service*

Figura 3. Exemplos de operações utilizadas no desenvolvimento da aplicação.

3.2. Ferramentas para o Desenvolvimento da Aplicação para Aprovisionamento de Circuito Óptico (interface *northbound*)

A Seção 2 apresentou as 7 funcionalidades que fazem parte da aplicação desenvolvida para o aprovisionamento de circuito óptico. A aplicação usou a interface TAPI 2.1.1 do sistema ONOS 2.5.1, e o TAPI utiliza chamadas RESTCONF.

A aplicação é executada totalmente desacoplada do controlador ONOS, que permite ao usuário utilizar as funcionalidades indicadas na Seção 2 através de uma interface gráfica para o usuário (GUI), apresentada na Seção 4. Essa aplicação utiliza o padrão Web com arquitetura multicamada. Como linguagem de desenvolvimento, adotou-se o Java /J2EE com o JDK 11 (*Java Development Kit*). Como servidor de aplicação, utilizou-se o WildFly 18.

[4]Projeto ONOS: `https://wiki.onosproject.org/display/ONOS/Downloads`.
[5]SoapUI: `https://www.soapui.org/`
[6]Postman:`https://www.postman.com/`

3.3. Modelos OpenConfig do ODTN usados no emulador e protocolo NETCONF (interface *southbound*)

O protocolo NETCONF é utilizado pelo ONOS para configurar os equipamentos programáveis, que no caso deste trabalho são representados pelos *transponders* ópticos Cassini emulados, detalhados na Seção 3.5. Tais equipamentos são modelados através de YANG e os modelos que representam o *transponder* Cassini são apresentados na Figura 4:

OpenConfig	openconfig-yang-types	openconfig-platform-types	openconfig-transport-types
	openconfig-types	openconfig-platform-transceiver	openconfig-transport-line-common
	openconfig-extensions	openconfig-platform-linecard	openconfig-transport-line-protection
	openconfig-alarm-types	openconfig-platform-port	openconfig-interfaces
	openconfig-terminal-device	openconfig-platform	openconfig-if-ethernet
IETF	ietf-interfaces		
	ietf-yang-types		
IANA	iana-if-type		

Figura 4. Modelos YANG do emulador ODTN Cassini.

3.4. Plataforma SDN Multicamada

A plataforma SDN Multicamada foi fruto de um convênio de cooperação técnica e científica da RNP com a empresa Huawei e financiado com recursos da Lei de Informática, com objetivo de estudar e desenvolver uma solução de orquestração multicamada (i.e., utilizando as camadas de pacote e óptica) através do paradigma SDN. Para prova de conceito (PoC) da solução construída, abordou-se o desenvolvimento de uma arquitetura para o transporte de dados, denominada Science DMZ, que permitiu a transferência confiável de grandes quantidades de dados científicos a 100 Gbps disco-a-disco. Além disso, este projeto contribuiu com disponibilização de um testbed SDN multicamada (óptico e L2/L3) e um sistema de emulação de redes com as tecnologias do testbed, para experimentação remota tanto pela comunidade acadêmica brasileira, quanto indústria. Na Figura 5, tem-se o ecossistema de tecnologias utilizadas tanto no testbed, quanto no emulador.

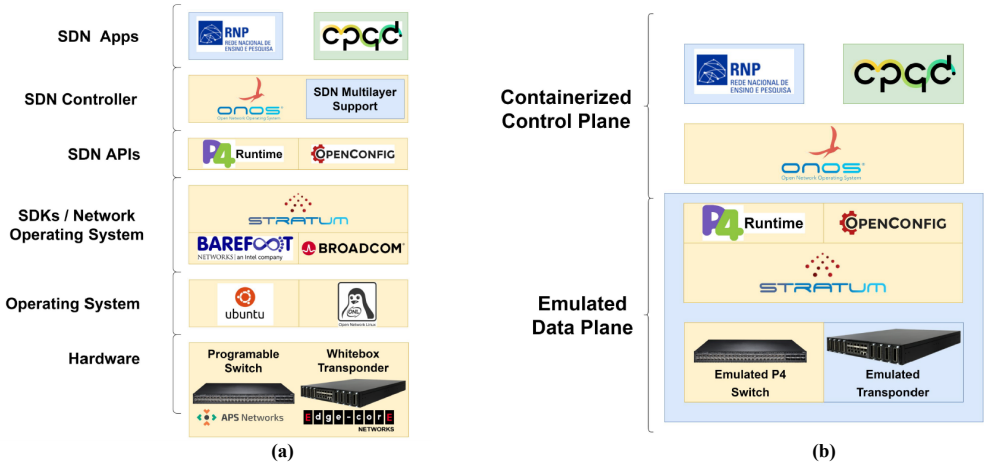

Figura 5. Ecossistema de tecnologias da Plataforma SDN Multicamada. Em (a) tem-se o ecossistema do testbed e na (b) o ecossistema do emulador.

Na Figura 5 (a), observa-se as tecnologias disponíveis para utilização em experimentos multicamada. Boa parte de seu hardware e software são abertos e disponíveis para o uso de qualquer usuário sinalizado pelo retângulo amarelo mais abaixo. Diferente, em partes, do ecossistema do testbed, o ecossistema do emulador, Figura 5 (b), apesar de utilizar esses padrões abertos (ex. P4, OpenConfig e Stratum) necessitou de desenvolvimento adicional para incorporar essas tecnologias e permitir a execução de experimentos no computador local. A camada de aplicação em ambos os ecossistemas são iguais, no ponto de vista da visualização de recursos do experimento, permitindo assim que a construção de aplicações seja executada tanto no ambiente de testbed, quanto emulador.

3.4.1. Emulador CNetLab

Como parte da plataforma SDN Multicamada, o CNetLab é um emulador de redes que possibilita trabalhar tanto com redes modernas (ex. openflow, p4 ou óptica), quanto com redes tradicionais (eg. TCP/IP). Sua arquitetura utiliza a conteinerização de sistema completo através do Docker, de forma, que ele isole os nós em *containers* e os interconecte através de enlaces ou túneis virtuais. A adoção de *container* oferece uma vantagem em relação a outros emuladores, que é o total isolamento de recursos. Além disso, durante o experimento emulado é permitida a utilização de uma diversidade de tipos de sistemas operacionais, protocolos, serviços de rede e aplicações. Através do *container* é possível emular uma infraestrutura completa de rede com hosts, servidores, switches, roteadores ou *transponders*, independentemente se são compatíveis à SDN ou outras arquiteturas. Na figura 6 (a), tem-se a arquitetura do CNetLab que foi desenvolvida para Plataforma SDN Multicamada e atualmente está dividida em quatro camadas: camada de usuário, camada de operação, camada de infraestrutura e repositório de imagens.

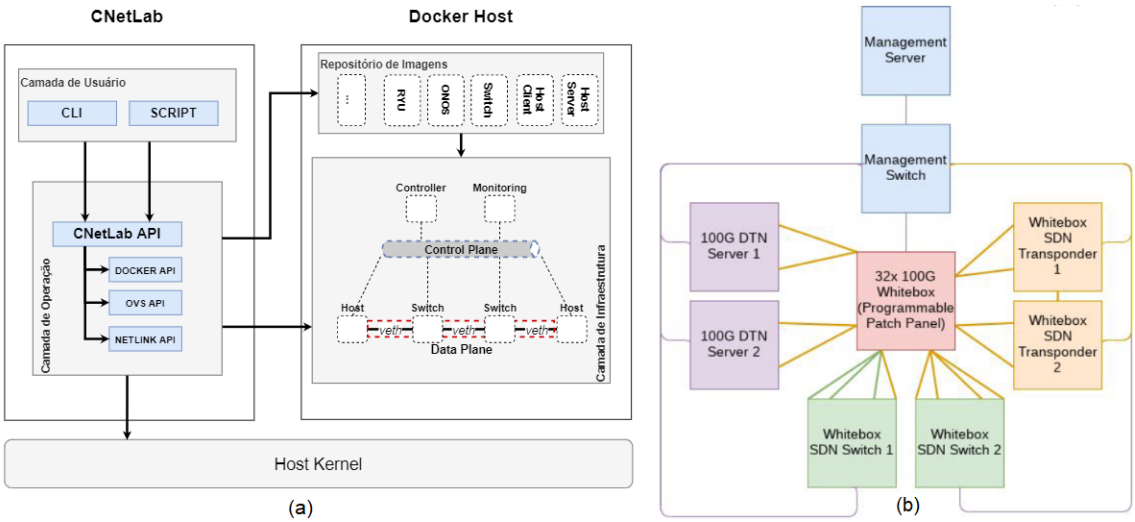

Figura 6. (a) Arquitetura do CNetLab da Plataforma SDN Multicamada e (b) Diagrama Topológico do Testbed SDN Multicamadas.

O CNetLab utilizou, para emulação de *transponders* ópticos, o emulador de *transponder* Cassini desenvolvido pelo grupo ODTN do ONF. A partir da implementação do ONF foi desenvolvido um plano de dados para o *transponder*. O detalhamento do desenvolvimento está descrito em 3.5. A seguir há uma rápida descrição das camadas:

- Camada do Usuário: interação do usuário às funcionalidades do emulador de duas maneiras - CLI e scripts Python utilizando API;
- Camada de Operação: bibliotecas para alocação de recurso no ambiente virtual, construção dos enlaces, acesso a outras funcionalidades internas do *container*, manipulação de elementos do experimento e extensão de novas funcionalidades e modelos utilizados pelo emulador;
- Camada de Infraestrutura: possui os recursos alocados pelo emulador durante a construção da topologia. Basicamente, ela é uma camada de software que contém todos os elementos lógicos de uma emulação. Possui dois canais independentes de conectividade com elementos da infraestrutura - um canal de controle e outro de dados;
- Repositório de imagens: comporta todos os *containers* que representam os elementos da emulação, como: switches, roteadores, servidores e clientes.

3.4.2. Testbed SDN Multicamada

O Testbed SDN Multicamada consiste em um laboratório óptico de experimentação remota, com soluções de DTN (*Data Transfer Node*), de capacidade de transferência até 100 Gbps. Atualmente, ela é composta por dois sistemas finais DTNs, dois switches *whitebox* (L2/L3), dois *transponders* (L1) e um switch *whitebox* funcionando como *patch panel* programável, além dos equipamentos de gerenciamento da plataforma, como mostrado na Figura 6 (b). Essa infraestrutura foi instalada em um rack disposto em um dos data centers da RNP. A implantação e customização de um portal e de um orquestrador permitem a realização de experimentos nessa plataforma por usuários remotos. Para atingir a velocidade de 100 Gbps de transferência, os sistemas finais precisam ser customizados através da escolha adequada dos componentes de hardware dos servidores, configurações do sistema operacional, protocolos de transporte confiáveis utilizados, dentre outros.

Nesse contexto, ainda existe o sistema de orquestração que permite a integração e automação de serviços e ferramentas, possibilitando a interação com a infraestrutura existente e o ambiente de aplicativos. Esse software é responsável pela instanciação desses recursos mantendo o isolamento entre eles. Além disso, o orquestrador é capaz de obter as informações sobre a reserva do laboratório realizadas pelo usuário e, também, é responsável por interagir com o controlador SDN e com os componentes de software que controlam os recursos computacionais necessários à transferência de dados.

3.4.3. Ciclo de Vida dos Experimentos

O ciclo de vida dos experimentos permite organizar e gerenciar a utilização dos recursos do Testbed SDN Multicamada entre os pesquisadores e seus experimentos. Atualmente, esse ciclo é criado através do portal de experimentação. O portal é a interface através da qual o usuário do testbed (também chamado de experimentador) solicita a reserva dos recursos computacionais do laboratório e faz as configurações necessárias à realização do experimento. O experimentador pode visualizar o status do laboratório, status dos equipamentos, agenda de reservas do laboratório e projetos que estão sendo realizados. Após isso, é necessário criar um projeto. O projeto agrupa as reservas e usuários que

podem acessar os recursos no momento em que a reserva ficar ativa. Todos os usuários que são membros de um projeto podem criar reservas, editar a reserva e modificar as configurações da infraestrutura que serão aplicadas no momento do aprovisionamento.

Uma outra questão, em relação ao ciclo de vida dos experimentos é a utilização do emulador, pois, devido à escassez da possibilidade em se manter muitos experimentos em paralelo, esse elemento da plataforma SDN multicamada é oferecido como o primeiro passo desse ciclo, onde o experimentador realizará seu primeiro setup do experimento e a primeira série de desenvolvimento de aplicações e teste que serão experimentados posteriormente no testbed. O objetivo disso é otimizar o uso dos recursos e tempo de utilização do testbed, pois tudo pode ser executado no emulador e posteriormente aplicado ao testbed apenas para coleta de resultados. Atualmente, esse passo é opcional, mas, futuramente, o emulador será um passo obrigatório nesse ciclo de vida dos experimentos, chamado de modelo *digital twins*, que permitirá integrar emulador e testbed de forma que esse experimento passe inicialmente por uma validação no emulador e depois seja encaminhado para execução no testbed e fique nesse ciclo contínuo de desenvolvimento e teste.

3.5. Detalhamento do Emulador ODTN/ONF Cassini e Extensão da RNP para o Plano de Dados

Open and Disaggregated Transport Network (ODTN) é um projeto que permite construir interconexões utilizando equipamentos ópticos desagregados, com padrões comuns e software de código aberto e permite que sejam usados *transponders* de diferentes fabricantes. Utiliza o controlador SDN ONOS, que descobre de forma automática e transparente os componentes desagregados e controla a rede de transporte como um todo. No caso específico deste trabalho, foi usado o emulador de *transponder* Cassini do ODTN que está disponível para uso pela comunidade.

As funcionalidades que podem ser executadas pelo controlador ONOS nos emuladores ODTN Cassini são: Descoberta da topologia dos Cassini, leitura de portas/status; Configuração de um circuito *line side* entre dois Cassini em duas portas ópticas disponíveis; Configuração de uma porta cliente e associação da mesma a um circuito *line side* para a formação de um circuito fim a fim; Remoção de circuito fim a fim pré-configurado.

Para emular o plano de dados entre os Cassini, a RNP desenvolveu uma extensão ao emulador do ODTN, utilizando OpenVSwitch (OVS), que permite mapear as portas dos Cassini em portas Ethernet no OVS. O plano de dados foi desenvolvido considerando a leitura das configurações do Cassini emulado na base SYSREPO[7] (base de dados de estado operacional e configuração baseada em YANG para aplicações Unix/Linux) e o mapeamento das configurações de portas e frequências em um switch virtual, representado na Figura 7 (a). Tais configurações permitem o envio de pacotes entre os Cassini no ambiente.

O emulador Cassini possui dois tipos de portas, as portas elétricas, denominadas (*client side*) e as portas ópticas, denominadas (*line side*), responsáveis por ligar os *hosts* ao Cassini e os Cassini entre si, respectivamente. Então, as portas elétricas do emulador Cassini 101, 102, 103, etc, são mapeadas e adicionadas às interfaces Ethernet nomeadas

[7]Projeto Sysrepo https://github.com/sysrepo/sysrepo.

como xe1/1, xe2/1, etc no OpenVSwitch. Analogamente, as portas ópticas 201, 202, 203, etc, são adicionadas às interfaces Ethernet nomeadas como oe1/1, oe2/1, etc. O mapeamento de portas também é representado na Figura 7 (b).

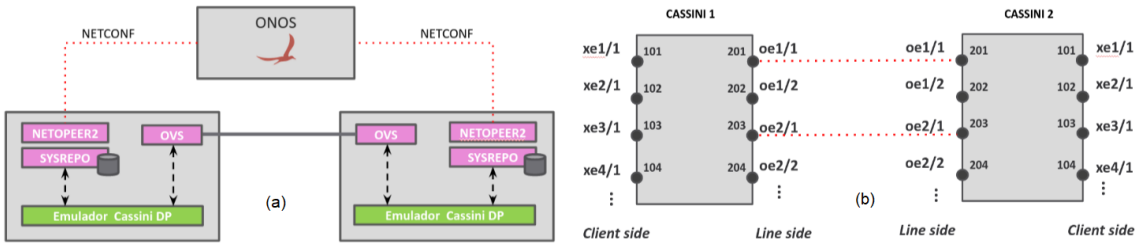

Figura 7. (a) Extensão do emulador Cassini do ODTN com plano de dados e (b) Mapeamento de portas do Cassini.

Além do mapeamento de portas, a extensão implementa um circuito óptico fim a fim, emulando as associações entre as portas elétricas e ópticas e os circuitos de linha entre as portas ópticas de Cassini distintos. Por se tratar de comunicação entre componentes ópticos, os links entre os Cassini se distinguem por meio de frequências. Para realizar a comunicação no plano de dados do emulador, são criadas VLANs (*Virtual Local Area Networks*) utilizando QinQ. Estas VLANs são responsáveis por encapsular os pacotes na porta cliente e adicionar *tags* que são capazes de referenciar a frequência óptica utilizada. As frequências utilizadas nos emuladores Cassinis são convertidas em VLANs, identificadas com o mesmo número da frequência configurada no circuito óptico. Com isso, o tráfego *untagged* recebido via cliente sai do plano de dados do Cassini emulado como *tagged* em portas *trunk*.

Para que o plano de dados sempre reflita o status atual do Cassini emulado, a extensão verifica constantemente o SYSREPO, analisando portas, status e frequências. Em caso de alguma alteração, as configurações são refletidas no OVS (*Open Virtual Switch*) e, consequentemente, alteradas no ambiente de experimentação.

3.6. Ambiente de testes SDN do CPQD e interconexão com Laboratório SDN Multicamadas da RNP

O CPQD, com o apoio da RNP, configurou um ambiente de testes em suas dependências no laboratório de SDN, contendo um servidor Dell PowerEdge R710. Uma máquina virtual de 80 GB de disco, 4 cores de CPU e 16 GB de RAM foi criada com sistema operacional Linux Ubuntu 20.04 LTS e foram iniciados os seguintes *docker containers*: um para a aplicação TAPI, um para o controlador ONOS 2.5.1 e dois emuladores ODTN Cassini com extensão para plano de dados.

O CPQD se conecta à rede IPÊ da RNP através da rede metropolitana de Campinas (Redecomep). Através destas conexões é possível interconectar o laboratório SDN do CPQD com o laboratório SDN Multicamadas da RNP, localizado no POP-RJ no Rio de Janeiro, através da criação de VPN entre as localidades.

4. Resultados

Esta seção apresenta uma visão geral do uso da aplicação *northbound*, descrevendo as funcionalidades de descoberta de topologia, descoberta dos *transponders* ópticos e respectivas portas cliente e de linha e a criação de um circuito fim a fim.

Para a criação de um circuito fim a fim são necessários dois passos: (1) criação de um circuito de linha (*line side*) e (2) associação de portas cliente (*client side*) a um circuito de linha previamente criado.

Para exemplificar, inicialmente não é possível verificar a conectividade entre os terminais ht1 e ht2 ligados aos *transponders* emulados A e B, respectivamente, conforme mostrado na Figura 8, já que ainda não há o circuito óptico criado.

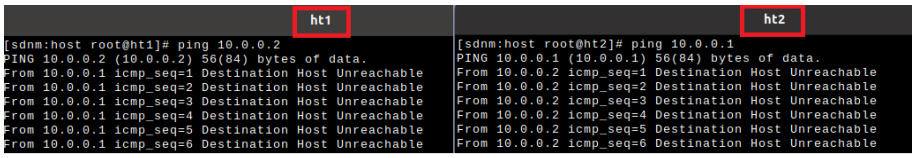

Figura 8. Teste inicial de conectividade entre os *transponders*.

A Figura 9 (a) apresenta a visualização da tela da aplicação *northbound*. Uma vez informado o endereço IP e a porta do controlador ONOS, ao clicar no botão *Load*, a aplicação recebe do controlador informações sobre os dispositivos emulados na aba *Devices*, bem como o detalhes da topologia e das portas dos *transponders* na aba *Links*.

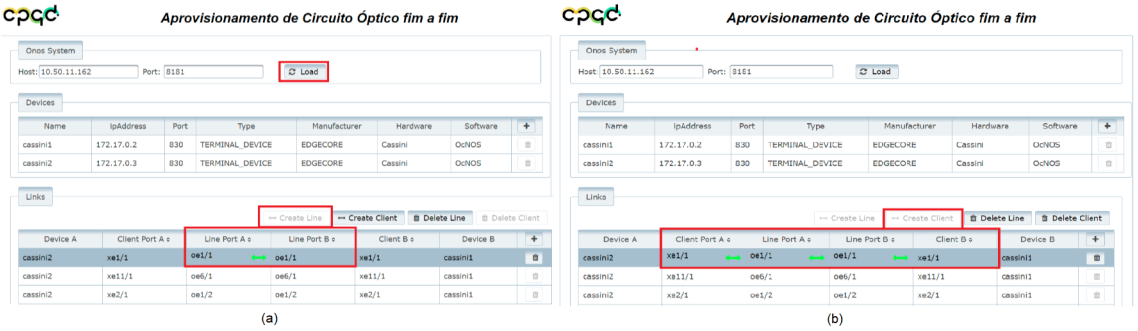

Figura 9. (a) Criação de circuito de linha (*line side*) e (b) Criação de circuito de linha. (*client side*)

Ao selecionar uma linha do subitem Links e clicar no botão *Create Line*, a aplicação envia para o controlador um pedido de estabelecimento de circuito óptico entre os Cassini via TAPI/RESTCONF. O controlador realiza as configurações no plano de controle, que podem ser visualizadas na linha destacada em vermelho na Figura 9 (a), aprovisionando um circuito óptico de linha entre as portas ópticas oe1/1 dos dois *transponders* emulados.

A Figura 9 (b) apresenta o segundo passo para a criação do circuito óptico fim a fim. Ao clicar no botão *Create Client*, o controlador aloca as primeiras portas cliente disponíveis em cada um dos *transponders* e associa as mesmas ao circuito de linha previamente criado. O circuito fim a fim está destacado na Figura 9 (b) no quadro em vermelho.

Realizado os passos de criação do circuito óptico na aplicação *northbound*, o plano de dados do Cassini emulado realiza o mapeamento da frequência utilizada pelo Cassini para uma VLAN. A porta de linha é adicionada à VLAN recém criada, estabelecendo a comunicação entre as interfaces lógicas. Ao identificar novos eventos ou requisições, as alterações são aplicadas no plano de dados e modificadas no repositório, que resultam na alteração da frequência e consequentemente, na modificação da VLAN. Após a

configuração do circuito fim a fim, tanto no plano de controle quanto no plano de dados do emulador, é possível realizar a verificação da conectividade entre os *transponders*, conforme mostrado na Figura 10.

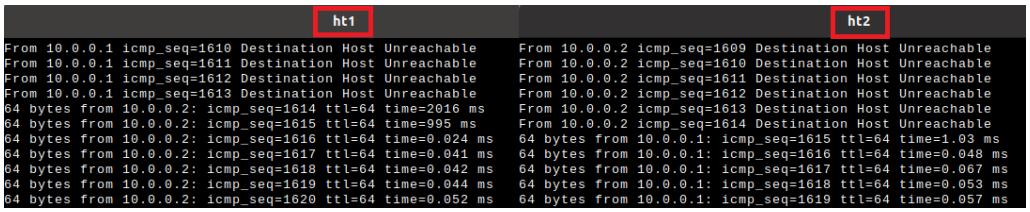

Figura 10. Verificação de conectividade entre *transponders* no plano de dados.

5. Vantagens e desvantagens do uso de emulação e de laboratório físico da Plataforma SDN Multicamadas

No tocante à utilização do emulador de *transponders* do ODTN em conjunto com a extensão desenvolvida pela RNP, a solução apresenta as seguintes vantagens:

- Eficiente para a construção de um ambiente de testes, considerando, tanto o plano de controle, quanto o plano de dados;
- Flexibilidade de criação de topologias além das permitidas pelo ambiente físico do Laboratório SDN Multicamadas;
- Prototipagem rápida de aplicações, de novos elementos de rede e testes de integração à rede;
- Emulação de um ambiente de rede óptica útil como recurso educacional, para práticas de ensino em universidades ou para formação de recursos humanos em áreas como programabilidade e virtualização de rede e desenvolvimento de orquestradores SDN.

Como pontos de desvantagem do uso de ambiente emulado, pode-se indicar:

- O emulador do transponder Cassini possui restrições de funcionalidades que limitam os tipos de testes que podem ser realizados;
- Devido ao emulador rodar em *container*, testes de desempenho reais de rede não são possíveis.

Algumas vantagens do ambiente físico do laboratório SDN Multicamadas são:

- Ambiente com hardware real e sistema operacional *Stratum*, onde se pode realizar testes com protocolos de padrão aberto como Netconf, gNMI, gRPC e gNOI;
- Possibilidade de testes de desempenho entre equipamentos, podendo-se simular grandes distâncias através de bobinas de fibra óptica de diferentes quilometragens;
- Uso das funcionalidades reais dos equipamentos, em que pode-se averiguar a maturidade do desenvolvimento pelos fabricantes nas soluções abertas para programabilidade.

Quanto às desvantagens do uso do ambiente físico, podem ser pontuados:

- Necessidade de interação com técnicos do POP-RJ e RNP caso haja algum problema como falta de energia, indisponibilidade dos equipamentos ou outro tipo de problema de acesso aos elementos de forma remota.

6. Conclusões e trabalhos futuros

Este artigo apresentou o desenvolvimento de uma aplicação *northbound* comunicando-se com o controlador ONOS via TAPI e RESTCONF para realizar o aprovisionamento de circuitos ópticos fim a fim.

A aplicação permitiu a descoberta de topologia, a descoberta das portas de linha e de cliente e a realização o aprovisionamento de um circuito fim a fim. A criação do circuito fim a fim pôde ser verificada no plano de dados, monitorando-se a conectividade entre dois terminais instalados em cada um dos emuladores de *transponder*. Em trabalhos futuros, a aplicação pode ser evoluída considerando novas funcionalidades, como a monitoração de status administrativo dos dispositivos ópticos e switches Stratum, configurações e outras funcionalidades via RESTCONF.

As funcionalidades de aprovisionamento podem ser evoluídas considerando a possibilidade de escolha de quais portas de cliente o usuário quer utilizar no aprovisionamento, assim como quais portas cliente associar a um circuito de linha pré-estabelecido. Para a escolha das portas cliente seria necessário realizar alterações no algoritmo interno do ONOS, que faz a escolha obtendo-se sempre a primeira porta cliente disponível no *transponder* óptico, não permitindo a escolha pela aplicação *northbound*.

Além da evolução da aplicação *northbound*, considera-se como trabalho futuro a utilização da mesma em um ambiente físico de rede óptica, podendo, para isso, serem utilizados diversos testbeds disponíveis para pesquisa e experimentação, sendo primeiro o Testbed SDN Multicamadas da RNP. Este testbed contém dois *transponders* ópticos físicos Cassini como os emulados e testados no CPQD. Desta forma, será possível validar as funcionalidades da aplicação já desenvolvidas e explorar outras funcionalidades do equipamento, já que as funções configuráveis dos emuladores são limitadas.

Referências

Bierman, A., Björklund, M., and Watsen, K. (2017). RESTCONF Protocol. RFC 8040.

Kreutz, D., Ramos, F. M. V., Verissimo, P., Rothenberg, C. E., Azodolmolky, S., and Uhlig, S. (2014). Software-Defined Networking: A Comprehensive Survey.

NG-SDN (2021). Next generationsoftware defined network. disponivel em: https://opennetworking.org/ng-sdn/. acessado em : 11/03/2021.

ODTN (2021). Open disaggregated transport network. disponível em: https://opennetworking.org/odtn/. acessado em : 11/03/2021.

ONF-TAPI (2021). Onf open transport api (tapi). disponível em: https://github.com/opennetworkingfoundation/tapi. acessado em : 11/03/2021.

ONF-TR147 (2020). TAPI v2.1.3 Reference - Implementation Agreement - TR-547 - Version 1.0. Technical report, ONF. https://opennetworking.org/wp-content/uploads/2020/08/TR-547-TAPI-v2.1.3-Reference-Implementation-Agreement-1.pdf.

Tomkos, I., Palkopoulou, E., and Angelou, M. (2012). A survey of recent developments on flexible/elastic optical networking. In *2012 14th International Conference on Transparent Optical Networks (ICTON)*, pages 1–6.