

Implementação e testes do *Virtual OLT Hardware Abstraction* (VOLTHA) em redes SD-PON

Luciano Martins¹, Luis G. M. Riveros¹, Carine Mineto¹, Michelle S. P. Facina¹,
Gustavo C. Lima¹, Lucas B. Oliveira², Fernando N. N. Farias²

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

²Rede Nacional de Ensino e Pesquisa (RNP) – Campinas, SP – Brasil

{lmartins, lriveros, cmineto, mfacina, gcorrea}@cpqd.com.br

{lucas.oliveira, fernando.farias}@rnp.br

Abstract. *This article describes the testbed used in the OpenRAN@Brasil project to check the functionalities and resources of VOLTHA (Virtual Optical Line Termination Hardware Abstraction) when controlling SD-PON (Software Defined Passive Optical Networks). Components that will compose the experimental environment are described, as well as the topology of each one of the sites in the Centro de Pesquisa e Desenvolvimento em Telecomunicações (CPQD) and Rede Nacional de Ensino e Pesquisa (RNP), organizations that are developing the project in partnership. Advantages and disadvantages related to the employment of the testbed are presented, and the implementation is made to emulate an FTTx network (Fiber to the “x”), composed of OLT (Open Line Terminal) and ONU (Optical Network Unit), with simulations for the control plane and considerations about a data plane developed with OVS (OpenVSwitch) complementary to the control plane. Through this emulated environment, it was possible to carry out a deployment and perform tests with several characteristics similar to those that will be implemented in the ideal testbed.*

Resumo. *O presente artigo descreve o testbed usado no projeto OpenRAN@Brasil para verificar a atuação do VOLTHA (Virtual Optical Line Termination Hardware Abstraction) ao controlar SD-PON (Redes Ópticas Passivas Definidas por Software). São descritos os componentes que irão compor o ambiente experimental, bem como a topologia de cada um dos sites no Centro de Pesquisa e Desenvolvimento em Telecomunicações (CPQD) e Rede Nacional de Ensino e Pesquisa (RNP), organizações que estão desenvolvendo o projeto em parceria. São apresentadas também vantagens e desvantagens relacionadas ao emprego do testbed, e a implementação feita para emular uma rede FTTx (Fiber to the “x”), composta por OLT (Open Line Terminal) e ONU (Optical Network Unit), com simulações para o plano de controle e considerações sobre um plano de dados desenvolvido com OVS (OpenVSwitch) complementar ao plano de controle. Através desse ambiente emulado, foi possível realizar um deployment e testes com características similares às que serão implementadas no testbed definitivo.*

1. Introdução

Nos últimos anos, é notável que as infraestruturas de rede tem se desenvolvido seguindo a tendência de softwarização em ambientes de nuvem, trazendo enormes benefícios e

desafios [Jakovčić et al. 2019]. Tal softwarização facilita a programabilidade dos elementos de rede assim como a virtualização dos seus recursos, permitindo, por exemplo, a alocação dinâmica e o particionamento da rede em fatias logicamente isoladas (*slicing*). A softwarização foi impulsionada pelo surgimento do paradigma SDN (*Software Defined Network*) [Lara et al. 2014].

Dessa forma, a inteligência da rede passou a ser centralizada no controlador SDN que, de posse de estatísticas de tráfego e informações de topologia constantemente coletadas, define as regras de encaminhamento de pacotes a serem usadas pelos equipamentos, de acordo com as políticas definidas pelos diferentes serviços de rede. Inicialmente, o conceito de SDN foi aplicado ao domínio de pacotes em ambiente de *datacenter*, sendo a primeira interface proposta e padronizada para a programabilidade do plano de dados dos equipamentos, o protocolo OpenFlow [McKeown et al. 2008].

Recentemente, o conceito de SDN vem também sendo aplicado aos domínios óptico e sem fio nas redes de comunicações das prestadoras de serviços, permitindo que um controlador gerencie elementos da rede óptica tais como transponders, comutadores, amplificadores e OLTs (*Open Line Terminals*), além de elementos em redes sem fio, como nas redes baseadas na arquitetura open RAN (*Open Radio Access Networks*). Mas para que isso seja possível, os equipamentos devem ser programáveis, permitindo que suas configurações sejam alteradas dinamicamente através de uma determinada interface.

O projeto OpenRAN@Brasil, desenvolvido em parceria entre CPQD (Centro de Pesquisa e Desenvolvimento em Telecomunicações) e RNP (Rede Nacional de Ensino e Pesquisa) com apoio de universidades como UFPA (Universidade Federal do Pará) e UNICAMP (Universidade Estadual de Campinas) desde o final de 2021, tem dentre suas metas a construção de um *testbed* com diversos domínios tecnológicos, sendo eles: Rede sem fio 5G (OpenRAN), Rede Óptica Passiva Definida por Software (SD-PON), Rede de transporte óptica DWDM (*Dense Wavelength-Division Multiplexing* softwarizada, rede de dados com programabilidade em P4 (*Programming Protocol-independent Packet Processors*), usando infraestrutura de nuvem como *kubernetes* e orquestração multidomínios.

Dentre os múltiplos domínios tecnológicos sendo investigados, testados e desenvolvidos no projeto, este artigo enfatiza o SD-PON, apresentando o *deployment* realizado em um ambiente temporário do *testbed*. Testes e resultados neste ambiente serão replicados no ambiente do *testbed* definitivo. É válido ressaltar a grande importância de existir *testbeds* disponíveis para o meio científico e industrial com tecnologias de ponta, para que soluções que englobam conceitos de orquestração, desagregação, SDN, programabilidade, controle e padrões abertos (*open source*), sejam experimentadas e desenvolvidas, sem trazer ônus a ambientes de produção.

Este artigo está organizado da seguinte forma: A Seção 2 se atém a uma descrição do *testbed*, tanto do ambiente definitivo quanto do ambiente temporário. Na Seção 3, é apresentada a arquitetura do VOLTHA (*Virtual Optical Line Termination Hardware Abstraction*), que abstrai a topologia física com tecnologia X-PON¹. A Seção 4 detalha a metodologia para os testes com o VOLTHA. Na Seção 5, são apresentados os resultados do *deployment* e dos testes já realizados. Por fim, a Seção 6 apresenta as conclusões e possíveis trabalhos futuros.

¹“X” pode ser GPON, XGS-PON, EPON, Combo PON.

2. Testbed do projeto OpenRAN@Brasil

O *testbed* definitivo do projeto OpenRAN@Brasil, previsto estar disponível para a comunidade no terceiro trimestre de 2023, cuja topologia é apresentada na Figura 1, contém 2 sites principais: o POP-RJ (Ponto de Presença da RNP) no Rio de Janeiro (Site A) e o CPQD em Campinas (Site B). São mostrados também os elementos que compõem cada um deles.

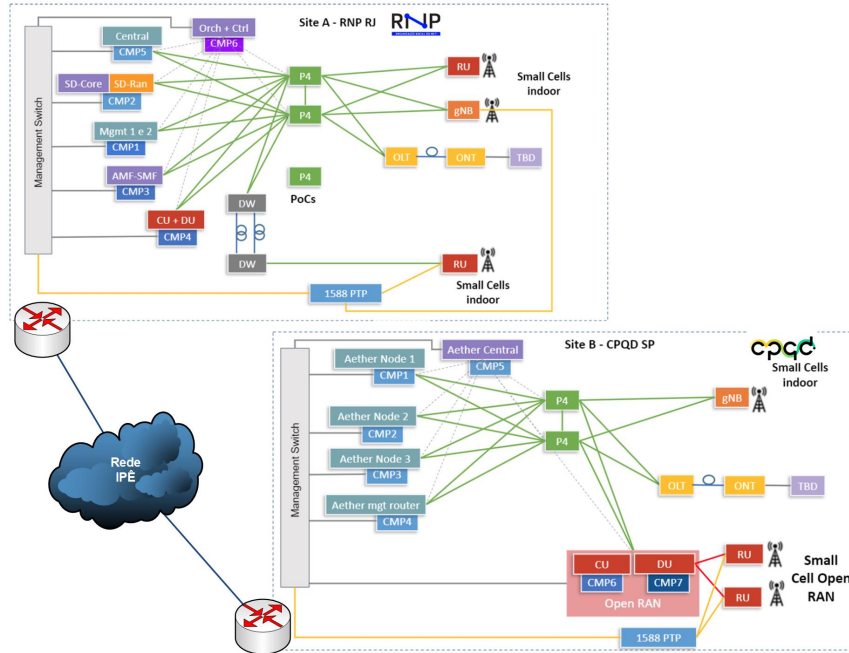


Figura 1. Topologia do *testbed* do projeto OpenRAN@Brasil que será disponibilizado para a comunidade no terceiro trimestre de 2023.

A lista de equipamentos que farão parte do *testbed* é apresentada na Figura 2, onde constam detalhes como os tipos, *labels*, funções dos equipamentos e quantidades que estão sendo adquiridas para cada *site*.

Equipamentos		Nós por Destino	
Tipo	Funções	RNP	CPqD
Compute Tipo 01	Bastion, Aether Central Node, SmartEdge RNP	3	2
Compute Tipo 02	CU+DU (RNP), DU (CPqD)	2	2
Compute Tipo 03	Aether Compute Server	1	1
Compute Tipo 04	CU (CPqD)	0	1
Compute Tipo 05	Orquestração RNP	1	1
Switches P4	UPF	2	2
RU	Antena OpenRAN	3	3
Sw Gerencia	Switch Gerencia	1	1
OLT Combo	OLT GPON e XGS-PON	0	1
OLT XGS-PON	OLT XGS-PON	1	0
ONT GPON	ONT GPON 4x1Gb, 2xPOT e WiFi	0	4
ONT XGS-PON	ONT XGSPON 1x10Gb, 4x1Gb, 2xPOT e WiFi	4	4
GrandMaster	Sincronismo de Relógio	1	1
Cassini	DWDM Programável	2	0

Figura 2. Lista de equipamentos do *testbed*.

Basicamente, ambos os sites possuem um conjunto de 6 a 7 servidores, categorizados em 5 tipos de configurações de hardware específicas. Cada *site* inicialmente terá um

conjunto de 3 antenas *indoor* baseadas na arquitetura OpenRAN. No domínio de pacotes, tem-se a aplicação de uma topologia *leaf-spine*, sendo que apenas a camada de *leafs* é formada de *switches* P4. No domínio óptico, há dois tipos de tecnologias disponíveis: uma baseada em FTTx (*Fiber to the x*), como a XGS-PON e GPON, e a outra baseada em WDM (*Wavelength Division Multiplexing*) com a utilização de *transponders* ópticos. Em que, o GPON conecta tanto terminais de usuários às ONUs (*Optical Network Unit*), quanto na infraestrutura 5G servindo como *backhaul*. Os sites serão interligados pela rede IPê da RNP através de enlaces de alta velocidade (10 Gbps) e VPN (*Virtual Private Network*) Layer 2.

Um ambiente temporário “pré-testbed” foi implementado, focado no SD-PON, para que seu *deployment* fosse experimentado e um *setup* fosse preparado para o momento em que servidores, OLTs e ONUs estivessem presencialmente nos *sites*. Este ambiente será detalhado na próxima subseção.

2.1. Ambiente *pré-testbed* do projeto para o domínio tecnológico SD-PON

Para viabilizar o *deployment* e testes de validação em um ambiente emulado, foi empregada a topologia apresentada na Figura 3. Nela é apresentado um servidor “VOLTHA” em cada um dos sites, onde são testados todos os recursos relacionados ao SD-PON e VOLTHA, cuja arquitetura será detalhada na Seção 3. O servidor “VOLTHA” é implementado em uma VM (*Virtual Machine*) com tecnologia de *cloud computing*, mais especificamente *kubernetes*, e a topologia PON, que será implementada com OLTs e ONUs programáveis na versão definitiva do *testbed*, foi implementada em caráter temporário com o BBSim (*BroadBand Simulator*), detalhado na Subseção 4.1. Nota-se também na Figura 3 um “*Jump Host*” que apoia a interconexão entre os sites através da criação de VPNs. Por fim, existe o *firewall* usado para a segurança de acesso aos *sites*.

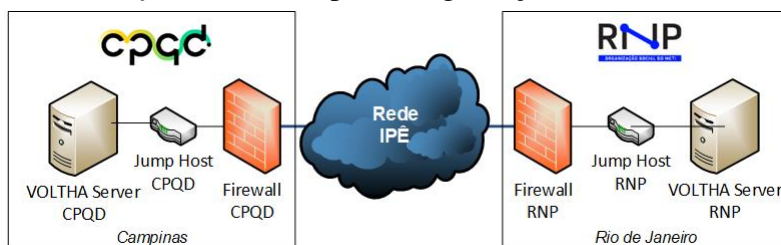


Figura 3. Topologia do ambiente *pré-testbed* para viabilização do *deployment* e testes de validação em um ambiente emulado.

2.2. Vantagens e desvantagens do *testbed* definitivo

As principais vantagens do ambiente físico, que será disponibilizado no *testbed* definitivo do projeto OpenRAN@Brasil, são:

- Disponibilizar um ambiente com *hardwares* reais (*whiteboxes*) para a comunidade nacional (científica, indústria, operadoras) fazer experimentações;
- Proporcionar um ambiente para realizar provas de conceito das tecnologias às operadoras, onde suas redes de produção podem ser construídas com os mesmos fabricantes usados no *testbed*, eventualmente levando a redução de custos;
- Ambiente multidomínio tecnológico baseado em *cloud computing* com controle e orquestração entre os domínios 5G (OpenRAN, RIC, etc.), SD-PON (XGS-PON e XG-PON) e DWDM;

- Possibilidade de testes de aplicações inovadoras que demandem alta largura de banda e baixa latência, tais como realidade virtual e aumentada, vídeo de altíssima definição e controle de drones.

Quanto às desvantagens do uso do *testbed*, podem-se pontuar:

- Necessidade de interação com técnicos do POP-RJ (RNP) e CPQD caso haja algum problema, como, por exemplo, falta de energia, indisponibilidade dos equipamentos ou outro tipo de problema de acesso aos elementos de forma remota;
- Na primeira fase do projeto OpenRAN@Brasil, fisicamente, o *testbed* estará localizado em apenas 2 sites: CPQD em Campinas e POP-RJ no Rio de Janeiro, podendo ser um limitante dependendo do tipo de testes a serem executados por órgãos nacionais (ou até mesmo internacionais). Porém, em uma possível Fase 3 do projeto há o planejamento de expansão do *testbed* para outras localidades do Brasil.

2.3. Vantagens e desvantagens do *pré-testbed*

As principais vantagens do ambiente físico usado no *testbed* “temporário” do projeto OpenRAN@Brasil são:

- Ambiente simples e com bom custo-benefício, com muitas similaridades funcionais ao ambiente definitivo, proporcionando uma experiência prévia das tecnologias que serão usadas na implantação do *testbed*;
- Ambiente que proporciona a criação de procedimentos para o *deployment* e operação dos domínios tecnológicos que serão alvo no ambiente de *testbed* definitivo.

Quanto às desvantagens do uso do *testbed* temporário, podemos citar:

- Recursos físicos com limitações em termos de CPU, memória e HD;
- Funcionalidades que existem em equipamentos reais não podem ser reproduzidas integralmente através de emuladores;
- Ambiente temporário não é preparado para ser disponibilizado para a comunidade realizar testes.

3. Arquitetura do VOLTHA

VOLTHA é um projeto de código aberto para criar uma abstração de *hardware* para equipamentos de acesso de banda larga. Atualmente, ele suporta o princípio de desagregação com vários fornecedores [Open Network Foundation 2023]. Sumariamente, este projeto é composto por duas camadas principais: 1 - Infraestrutura, onde são armazenados os dados, mensagens e o controlador de SDN; 2 - Pilhas, que incluem o VOLTHA Core, adaptadores e o agente *openflow*. Além disso, há uma camada opcional, que gerencia os dispositivos, por exemplo uma atualização de software por APIs (*Application Programming Interfaces*). Na Figura 4, é apresentado o diagrama da arquitetura do VOLTHA [Open Network Foundation 2023], em que cada retângulo representa os contêineres dos componentes, comumente implementados em *clusters kubernetes*. Representando a camada de infraestrutura estão o controlador ONOS (*Open Network Operating System*), o Kafka e o Etcd; enquanto que VOLTHA core, OpenOLT, OpenONU Adaptor e Whitebox OLT representam as pilhas do VOLTHA.

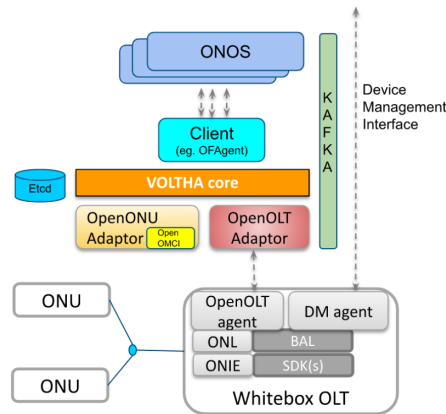


Figura 4. Diagrama da arquitetura do VOLTHA [Open Network Foundation 2023].

Ao analisar a camada de infraestrutura, tem-se o ONOS como controlador SDN responsável por gerenciar o chaveamento abstrato entre ONUs e OLTs, bem como as regras de encaminhamento de tráfego e tratamento de falhas; para isso, existe um sistema de armazenamento de dados próprio. Para os demais componentes do VOLTHA, o *cluster* Etcd cumpre esse papel de armazenamento de informações. Além disso, o *cluster* kafka é o responsável por externalizar os eventos demandados pelo operador OSS/BSS (*Operation Support System / Business Support System*).

Por outro lado, na camada da pilha [VOLTHA 2023a], tem-se o VOLTHA *core*, que recebe as solicitações do ONOS e as encaminha ao adaptador correspondente, além de manipular os registros e configurações advindos destes e os armazena no *cluster* etcd, ele também faz a abstração dos pares OLT e ONU como chaves lógicas. Já o adaptador OpenOLT realiza a importação dos modelos para estabelecer a comunicação entre a OLT física e o VOLTHA, permitindo assim a coleta de informações, eventos e *status*, bem como a abstração da tecnologia instalada na OLT, por exemplo GPON, XGS-PON, EPON. Assim também, o adaptador OpenONU interage com a ONU via OMCI (*Optical network termination Management and Control Interface*), enviando comandos de descoberta de equipamentos, *upload* de MIB (*Management Information Base*), configuração de ME (*Mobile Equipment*). Por fim, o *openflow agent* (*ofAgent*) estabelece a conexão entre o ONOS e o VOLTHA *core*, de modo a traduzir os eventos advindos do VOLTHA e as instruções do ONOS em chamadas *OpenFlow* e *gRPC*.

Na Figura 4, há também os blocos que compõem a arquitetura na *whitebox* do VOLTHA tal como OpenOLT *agent*, o qual fornece uma interface de gerenciamento e controle para as OLTs baseada em *gRPC*. Integrados a ele, podem estar o BAL (*Broadband Adaptation Layer*) que é um *software* da Broadcom responsável por promover a interface entre os chipsets de interfaces proprietárias nas OLTs, bem como eventuais SDKs (*Software Development Kit*) que facilitam o desenvolvimento de futuras novas aplicações. O agente DM (*Device Management*) permite a gerência de aspectos alheios ao VOLTHA a um operador OSS/BSS, que pode aproveitar para enviar informações pelo barramento kafka e se comunicar com a OLT física, o ONL (*Open Network Linux*) representando a camada de *software* para as próximas gerações de redes com hardware aberto para a interoperabilidade e ONIE (*Open Network Install Environment*) com o intuito de viabilizar a compatibilidade e automação dos softwares de operação de rede. Ambos fazem parte do projeto de computação aberta.

Na camada de controle, destacam-se as aplicações do ONOS [VOLTHA 2023b] que estão intrínsecas ao VOLTHA. São elas: a AAA (*Authentication, Authorization, and Accounting*) com o papel de servidor NAS (*Network Attached Storage*) e autenticador RADIUS (*Remote Authentication Dial in User Service*) das portas, que mantêm o estado delas durante o processo de autenticação com máquinas de estados; dhcp12relay é um agente de retransmissão DHCP da camada L2, o qual marca duas vezes os pacotes DHCP enviados (*discover/request*) ao servidor DHCP, recupera os valores das tags para serem usados no Sadis, e coloca tags de prioridade nos pacotes DHCP recebidos do servidor (OFFER/ACK); Mac Learning examina os pacotes DHCP do tipo *discover/request* e mantém o endereço MAC temporariamente conforme as informações obtidas nestes pacotes; a aplicação OLT é responsável por configurar os fluxos requeridos pela gerência de uma OLT física conforme respostas do VOLTHA; OLT *Topology* representa a topologia da rede com os equipamentos que estão conectados e ativos; Sadis (*Subscriber Access Device Information Service*) é um serviço opcional que fornece uma ponte para a infraestrutura do cliente a fim de consultar/guardar as informações do assinante/acesso e disponibilizá-las para outros serviços dentro do ONOS, além do Kafka e Multicast.

4. Metodologia para os testes com o VOLTHA

O ambiente de implementação do VOLTHA é construído em *cloud* na arquitetura de *kubernetes*. Nesse modelo, as funcionalidades do sistema de virtualização de redes PON são distribuídas em um *cluster* com camadas de acordo com a estrutura do VOLTHA. O acesso às camadas do cluster são definidas pelas interfaces *northbound* e *southbound*. Tanto a camada de controle quanto a pilha do VOLTHA possuem ambas as interfaces.

Dentre os protocolos para comunicação, podem-se utilizar gRPC para a conexão ao VOLTHA, e openflow para a conexão com o controlador. A CLI (*Command-Line Interface*) é outra forma de acesso ao sistema e também ao controlador. A interface *southbound* do controlador é responsável por conectá-lo à camada do VOLTHA utilizando protocolo openflow. No nível mais elevado, encontra-se a interface *northbound* do ONOS, responsável por receber e executar requisições para o controlador proveniente de um orquestrador ou operador externo, com uso de CLI ou utilizando API REST.

NAMESPACE	NAME	READY	STATUS	RESTARTS	AGE
infra	bbsim-sadis-server-7547584ff6-pjjlm	1/1	Running	0	20d
infra	elasticsearch-master-0	1/1	Running	0	20d
infra	kafkacat-7845b87866-gundj	1/1	Running	8 (137m ago)	8d
infra	voltha-infra-etcd-0	1/1	Running	0	20d
infra	voltha-infra-fluentd-elasticsearch-fjhpp	1/1	Running	0	20d
infra	voltha-infra-freeradius-859fb6c4c9-ss5mb	1/1	Running	0	20d
infra	voltha-infra-kafka-0	1/1	Running	0	20d
infra	voltha-infra-kibana-668df56fdf-nmslz	1/1	Running	0	20d
infra	voltha-infra-onos-classic-0	1/1	Running	0	20d
infra	voltha-infra-onos-classic-onos-config-loader-7b578bfc4-b7g6m	1/1	Running	0	20d
infra	voltha-infra-voltha-tracing-jaeger-646c659d6d-t5wqc	1/1	Running	0	20d
infra	voltha-infra-zookeeper-0	1/1	Running	0	20d
kube-flannel	kube-flannel-ds-j99v8	1/1	Running	6 (108d ago)	219d
kube-system	coredns-6d4b75cb6d-shdk	1/1	Running	4 (111d ago)	219d
kube-system	coredns-6d4b75cb6d-x6n6b	1/1	Running	4 (111d ago)	219d
kube-system	etcd-oran-seba	1/1	Running	4 (111d ago)	219d
kube-system	kube-apiserver-oran-seba	1/1	Running	7 (104d ago)	219d
kube-system	kube-controller-manager-oran-seba	1/1	Running	166 (99d ago)	219d
kube-system	kube-proxy-xlxtp	1/1	Running	4 (111d ago)	219d
kube-system	kube-scheduler-oran-seba	1/1	Running	163 (99d ago)	219d
voltha	bbsim0-69485c9f5f-ql598	1/1	Running	0	20d
voltha	voltha-voltha-adapter-openolt-75d5fc78dc-j5ptt	1/1	Running	0	20d
voltha	voltha-voltha-adapter-openonu-59bd5c7dc7-6sfff	1/1	Running	1 (20d ago)	20d
voltha	voltha-voltha-ofagent-6cc7899f94-8fXk8	1/1	Running	0	20d
voltha	voltha-voltha-rw-core-784dfbc4-4jzdz	1/1	Running	0	20d

Figura 5. Cluster *kubernetes* do VOLTHA.

A Figura 5 mostra uma saída do comando “*kubectl*”, listando os *Pods* do cluster da infraestrutura implementada do VOLTHA em *kubernetes*. Dadas as interfaces da infraestrutura de controle do SD-PON, neste trabalho será apresentado o uso de REST para

o gerenciamento da rede, bem como a CLI do VOLTHA para o controle da infraestrutura implementada.

4.1. Ambiente de experimentação

Uma vez que a topologia da rede ainda não possui os equipamentos físicos para o uso, uma alternativa foi o uso do simulador de banda larga, o BBSim (*BroadBand Simulator*) da ONF que faz parte do projeto VOLTHA. Esse simulador permite emular um dispositivo compatível com OpenOLT, por conseguinte, OLTs, portas PON, ONUs, UNIs (*User Network Interface*) e RGs (*Residential Gateway*). A partir da abstração dos equipamentos é possível simular aplicações de gerenciamento, possibilitando uma execução virtualizada de operação. Entretanto, o simulador apresenta somente o plano de controle da rede, sem o plano de dados, e com isso não abrange o tráfego de pacotes entre dispositivos. Então, a partir da implementação do BBSim, foi desenvolvido um plano de dados para o simulador pela equipe do OpenRAN@Brasil da RNP, o que possibilita uma visão completa do sistema emulado com plano de controle e dados, tal qual em um sistema real.

Com esse fim, a emulação do plano de dados se deu com a implementação de um *switch* virtual no núcleo do BBSim, o OVS (OpenVSwitch). Assim, por meio do OVS é possível mapear os componentes emulados em elementos virtuais, como pontes e portas, bem como em portas Ethernet. Para controlar o ciclo de vida dos dispositivos, o BBSim faz uso de máquina de estados, que a cada operação, desencadeia ações responsáveis por mapear o estado dos dispositivos em configurações nas pontes, portas e VLANs. Tais configurações permitem o envio de pacotes entre os usuários e os serviços.

A OLT simulada no BBSim possui dois tipos de portas, as portas PON e a porta NNI (*Network-Network Interface*), responsáveis por se conectarem às ONUs e à rede, respectivamente. Além disso, as ONUs também possuem dois tipos de portas, as portas UNI, e a porta de comunicação com a OLT. Então, no OVS, as interfaces Ethernet de entrada são adicionadas às portas UNIs, enquanto que a interface Ethernet de saída é adicionada a porta NNI. O circuito fim a fim é implementado emulando as associações entre as portas UNI, PON e NNI, representado na Figura 6. Dessa forma, o tráfego recebido via cliente percorre o circuito ONU-OLT e alcança a rede externa.

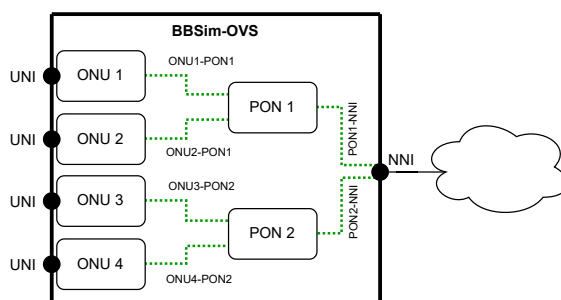


Figura 6. Mapeamento de portas do BBSim.

Para que o plano de dados sempre reflita o *status* atual do plano de controle, a máquina de estado implementada no BBSim tem papel fundamental. Após qualquer alteração em seu estado, ela desencadeia funções para analisar o ambiente. Em caso de alteração, as configurações são refletidas no OVS e, conseqüentemente, alteradas no ambiente de experimentação.

4.2. Implementação no ambiente emulado

O procedimento para implementação da infraestrutura emulada do sistema de rede FTTx é descrito nos passos a seguir:

1. Acesso à CLI do VOLTHA (voltctl);
2. Verificação dos adaptadores OpenOLT e OpenOnu conectados aos equipamentos;
3. Criação da topologia virtualizada com os parâmetros da OLT;
4. Ativação da OLT e das ONUs;
5. Listagem e verificação da topologia empregada e conexão dos equipamentos.

A lista de dispositivos da topologia virtualizada é mostrada na Figura 7, onde se visualiza as informações como identificação dos equipamentos, tipo, *serial number*, *status* administrativo e de conexão.

ID	TYPE	ROOT	PARENTID	SERIALNUMBER	ADMINSTATE	OPERSTATUS	CONNECTSTATUS	REASON
cb1971bf-f872-4003-8503-4fe741c3756e	openolt	true	7243964a-157e-4213-94c3-cbf8c6ced4b9	BBSIM_OLT_10	ENABLED	ACTIVE	REACHABLE	
bcecc86-d008-4723-ab22-6b57b8418ad2	brcm_openomci_onu	false	cb1971bf-f872-4003-8503-4fe741c3756e	BBSM000a0102	ENABLED	ACTIVE	REACHABLE	omci-flows-pushed
ffdc194f-6f67-452a-8e0d-77639324a633	brcm_openomci_onu	false	cb1971bf-f872-4003-8503-4fe741c3756e	BBSM000a0002	ENABLED	ACTIVE	REACHABLE	omci-flows-pushed
f6ba0c14-353f-4fa8-911b-7ef0d2631675	brcm_openomci_onu	false	cb1971bf-f872-4003-8503-4fe741c3756e	BBSM000a0001	ENABLED	ACTIVE	REACHABLE	omci-flows-pushed
oedd801e-95e7-4eae-833f-bc68130771a9	brcm_openomci_onu	false	cb1971bf-f872-4003-8503-4fe741c3756e	BBSM000a0101	ENABLED	ACTIVE	REACHABLE	omci-flows-pushed

Figura 7. Topologia virtualizada da rede FTTx.

Na Figura 8 é mostrada a topologia da rede PON criada no simulador com uma arquitetura de 4 ONUs e 1 OLT. Essa arquitetura é criada com duas portas PON provenientes da OLT que se conectam aos divisores de sinais alcançando cada porta de interface das ONUs, em que, cada ONU possui uma porta UNI na interface de usuário totalizando 4 portas UNI para operação. No outro lado da OLT está uma porta NNI para conexão externa da rede e a porta de gerenciamento para conexão com adaptadores de OLT virtualizados.

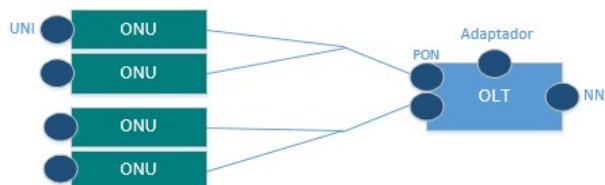


Figura 8. Arquitetura representativa da topologia simulada de uma rede FTTx com 4 ONUs e uma OLT.

Após a implementação dos equipamentos na estrutura do VOLTHA, a visualização é feita a partir da camada do controlador. A topologia é identificada pelo controlador ONOS e assume-se o estado de autenticação para a operação dos equipamentos. A partir da API “*aaa*”, é possível obter as portas que estão autenticadas e disponíveis para a ativação de serviços na camada óptica. O número de portas de autenticação não necessariamente é igual ao número de equipamentos da topologia, sendo que, alguns podem estar desativados, ocupados ou indisponíveis, somente identificando os usuários autenticados para operação. Na Figura 9, observam-se os pontos de terminais autenticados pelo controlador e disponíveis para a ativação de clientes.

Diante das etapas de mecanismos de gerência, realiza-se as configurações de serviço para atribuição aos clientes. Desse modo, a API “*network*” do sistema do controlador é responsável por alterar e aplicar as configurações de serviço para a infraestrutura implementada do FTTx. Entre os principais parâmetros estão: o nome do serviço, a tag do cliente, a tag do serviço, o nome da ONU, endereço MAC dos equipamentos, o perfil

```

karaf@root > aaa-users
of:00000a0a0a0a0a0a/256: AUTHORIZED_STATE, last-changed=8d2h ago, mac=2E:0A:00:01:00:00, subid=BBSM000a0001-1, username=user
of:00000a0a0a0a0a0a/512: AUTHORIZED_STATE, last-changed=8d1h ago, mac=2E:0A:00:02:00:00, subid=BBSM000a0002-1, username=user
of:00000a0a0a0a0a0a/65792: AUTHORIZED_STATE, last-changed=7d3h ago, mac=2E:0A:01:01:00:00, subid=BBSM000a0101-1, username=user
of:00000a0a0a0a0a0a/66048: AUTHORIZED_STATE, last-changed=20d8h ago, mac=2E:0A:01:02:00:00, subid=BBSM000a0102-1, username=user
karaf@root >

```

Figura 9. Pontos de terminais autenticados para operação de serviços.

de tecnologia, os perfis de largura de banda de *upstream* e *dowstream*, dentre outros. A aplicação das configurações é feita via REST e utilizando um arquivo contendo os dados necessários do tipo dicionário com formato *.json*, que são enviados e armazenados no banco de dados do controlador.

A partir das configurações fornecidas pelo operador, o sistema de gerenciamento apresenta o estado pré-aprovisionamento de um cliente. Através da API “*olt*”, pode-se escolher o ponto de terminal a ser aplicado o aprovisionamento de um determinado cliente de acordo com as configurações desejadas para o serviço. Com o comando REST recebido, o fluxo do aprovisionamento de um cliente é realizado, tornando o serviço habilitado para o ponto de terminal selecionado. O *status* da conexão passa de autenticado para ativado, permanecendo até que uma alteração seja realizada por parte do operador. Alterações das configurações e remoção do aprovisionamento podem ser aplicadas em um ou vários clientes sem comprometer a operação do sistema.

5. Resultados

No plano de controle, a configuração da camada de serviço e o aprovisionamento de um usuário para conexão do acesso aos serviços digitais são as etapas sistêmicas do gerenciamento da rede SD-PON. Adicionar um assinante, aplicar as configurações de operação e ativar as requisições do sistema foram realizados seguindo um fluxo de implementação de assinante em FTTx.

A partir da ativação do serviço e aprovisionamento do assinante, os dados são mostrados na Figura 10. O controlador armazena um perfil de assinante, que contém informações a respeito dos parâmetros de serviço. Dentre os principais parâmetros presentes estão: o ponto de terminal do usuário, que identifica a porta de conexão da ONU com o endereço da OLT; os valores das tags de cliente e do serviço na conexão com as portas PON; o identificador do perfil de tecnologia, que determina as condições de operação das interfaces das ONUs; o perfil de largura de banda de *uplink* e *downlink* do plano de gerência e do plano de dados do assinante, constituído pelos valores dos parâmetros de taxa de informação e o nome do serviço em operação.

```

{
  "entries": [
    {
      "location": "of:00000a0a0a0a0a0a/512",
      "tagInfo": {
        "uniTagMatch": 0,
        "ponCTag": 10,
        "ponSTag": 15,
        "usPonCTagPriority": -1,
        "usPonSTagPriority": -1,
        "dsPonCTagPriority": -1,
        "dsPonSTagPriority": -1,
        "technologyProfileId": 64,
        "upstreamBandwidthProfile": "User_Bandwidth1",
        "downstreamBandwidthProfile": "User_Bandwidth2",
        "upstreamOltBandwidthProfile": "User_Bandwidth1",
        "downstreamOltBandwidthProfile": "User_Bandwidth2",
        "serviceName": "hsia",
        "enableMacLearning": false,
        "configuredMacAddress": "",
        "isDhcpRequired": true,
        "isIcmpRequired": false,
        "isPppoeRequired": false
      }
    }
  ]
}

```

Figura 10. Usuário provisionado.

Dado o processo de provisionamento do assinante, o fluxo direcionado para a execução da camada de gerência do sistema é mostrado na Figura 11, sendo representado (a) a tela de saída da CLI do ONOS e (b) o fluxograma que descreve de forma visual os passos retratados em (a). Então, para a implementação do fluxo de serviço, o controlador inicia identificando o ponto de terminal do assinante. Diante disso, algumas configurações são removidas da ONU, como a VLAN padronizada do cliente e são adicionadas novas configurações dos protocolos DHCP (*Dynamic Host Configuration Protocol*) e EAPOL (*Extensible Authentication Protocol over LAN*), seguindo para a aplicação do plano de controle com os parâmetros do serviço. Para completar o processo, os perfis da largura de banda para o plano de dados são aplicados, provendo ao assinante o serviço orquestrado.

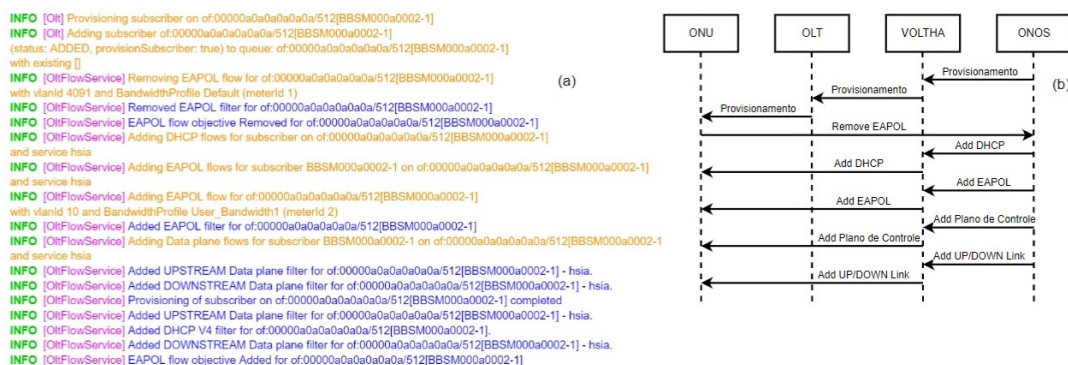
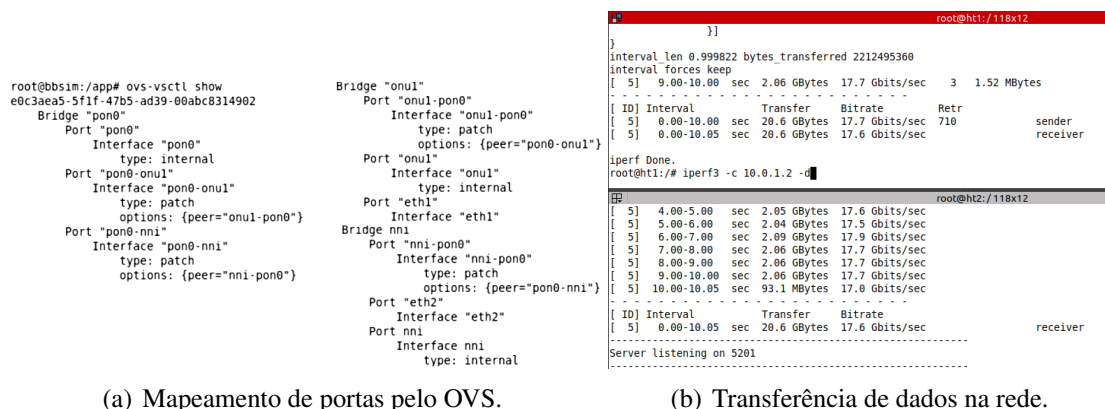


Figura 11. Fluxo de provisionamento da camada de controle.

Enquanto isso, os parâmetros recebidos são utilizados pelo OVS, no plano de dados, para configurar as portas, tipo e capacidade de tráfego entre os enlaces. A complexidade das configurações e operações são abstraídas e mostradas na Figura 12(a). As configurações geradas a partir da interação entre controlador SDN, VOLTHA, máquina de estados do BBSim e OVS proporcionam a transferência de dados entre os dispositivos, como visto na Figura 12(b)



(a) Mapeamento de portas pelo OVS.

(b) Transferência de dados na rede.

Figura 12. Esquema de representação do plano de dados com OVS.

6. Conclusões e trabalhos futuros

Por meio do presente trabalho, nós apresentamos a visão geral da topologia do *test-bed* do projeto OpenRAN@Brasil, o que permitiu ter uma noção dos equipamentos que

compõem tal ambiente experimental, tanto em ambiente de simulação, quanto no ambiente definitivo que está sendo construído e será disponibilizado no terceiro trimestre de 2023. Dessa forma, estabeleceu-se uma rede FTTx com 4 possíveis usuários a serem provisionados. Para isso, foi apresentada uma visão inicial dos *testbeds* temporários e definitivos que serão instalados no CPQD e RNP, bem como as vantagens e desvantagens em utilizá-los. Também foi apresentada uma breve introdução da arquitetura do VOLTHA, seus componentes internos e aplicativos, que permitem gerenciar o plano de controle, para interagir diretamente com OLTs. Ressaltando, que neste caso as OLTs foram emuladas pelo BBSim, tanto na camada de controle, quanto na camada de dados que foi implementada pela equipe do projeto.

Nesse contexto, há melhorias a serem propostas ao *testbed* do SD-PON, como o desenvolvimento de uma camada de CO (*Central Office*), que é responsável por gerenciar e controlar toda a infraestrutura do VOLTHA em uma camada de mediação conhecida como sistema de gerenciamento do elemento. Esse recurso permite o gerenciamento do ponto de vista dos elementos e a integração com os sistemas de rede das operadoras de telecomunicações.

Portanto, vislumbrando o grande potencial de uso do *testbed* implementado pelo CPQD em parceria com a RNP, há possíveis trabalhos futuros que incluem atividades tal como o desenvolvimento e testes de controle e orquestração para a integração de diferentes domínios tecnológicos, como 5G, FTTx, WDM; interação com sites internacionais que dispõem de *testbeds* já instalados; instalação de novos *sites* nacionais com *testbeds* integrados testando as tecnologias investigadas; e consolidação de sistema OSS/BSS para o uso de padrões abertos e desagregados.

Referências

- Jakovčić, A., Barać, V., and Blažević, B. (2019). Critical infrastructure solutions for upcoming developments in telecom and data industries. In *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 451–456. doi : 10.23919/MIPRO.2019.8756988.
- Lara, A., Kolasani, A., and Ramamurthy, B. (2014). Network Innovation using OpenFlow: A Survey. *IEEE Communications Surveys Tutorials*, 16(1):493–512. doi : 10.1109/SURV.2013.081313.00105.
- McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: enabling innovation in campus networks. *ACM SIGCOMM computer communication review*, 38(2):69–74.
- Open Network Foundation (2023). Voltha. disponível em: <https://opennetworking.org/voltha/>. acessado em : 07/02/2023.
- VOLTHA (2023a). Voltha architecture overview. disponível em: https://docs.voltha.org/master/overview/architecture_overview.html. acessado em : 18/02/2023.
- VOLTHA (2023b). Voltha documentation. disponível em: <https://docs.voltha.org/master/index.html> . acessado em : 18/02/2023.