

Teste para validação do *Virtual OLT Hardware Abstraction* (VOLTHA) para gerência de 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

² Rede 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 implementation 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) in addition to the control plane.*

Resumo. *O presente artigo descreve 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) adicional ao plano de controle.*

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 [Hakiri and Berthou 2015]. A softwarização foi impulsionada pelo surgimento do paradigma SDN (*Software Defined Networking*) [Casado et al. 2014], cujo conceito consiste na separação entre plano de controle e de dados, que até então eram implementados em conjunto nos equipamentos de rede.

Inicialmente, o conceito de SDN foi aplicado ao domínio de pacotes em centro de dados, entretanto, o conceito vem se expandindo aos domínios óptico e sem fio, nas redes de comunicações das operadoras de serviços. Assim, é possível que um controlador gerencie não apenas roteadores e *switches*, no domínio de pacotes, como também 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*).

Considerando tal expansão e programabilidade das redes ópticas, esse trabalho tem como objetivo investigar e validar a arquitetura proposta no domínio SD-PON (*Software Defined Passive Optical Networks*), com ênfase ao componente VOLTHA (*Virtual Optical Line Termination Hardware Abstraction*), responsável por abstrair a topologia física da rede X-PON (*Passive Optical Network*)¹. Para tal, a pilha de software da arquitetura SD-PON é implementada em conjunto com a ferramenta BBSim (*BroadBand*

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

Simulator) para emular os dispositivos da rede óptica compatíveis com OpenOLT. Então, os resultados obtidos no estudo serão considerados durante a adoção da pilha de software implementada no testbed construído no contexto do projeto OpenRAN@Brasil.

Além dessa seção introdutória, este artigo segue apresentando na Seção 2, a arquitetura do VOLTHA. Na Seção 3 é detalhada a metodologia para os testes com o VOLTHA. Os resultados dos testes e do *deployment* são elucidados na Seção 4. Por fim, na Seção 5 apresenta-se as conclusões e possíveis trabalhos futuros.

2. 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 equipamentos advindos de vários fornecedores [Open Network Foundation 2023]. Na Figura 1, é apresentada a arquitetura do VOLTHA dividida em duas camadas, Infraestrutura e Pilha de softwares [Open Network Foundation 2023]. Cada retângulo representa os componentes da arquitetura, comumente implementados em *clusters kubernetes*.

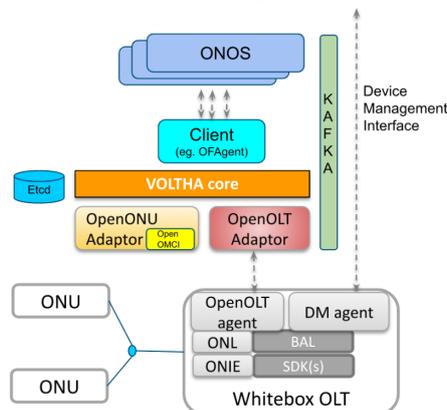


Figura 1. Diagrama da arquitetura do VOLTHA [Open Network Foundation 2023].

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

Já na Pilha de software, 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, fazendo também 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. 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 camada de controle, destacam-se as aplicações do ONOS [VOLTHA 2023] que estão intrínsecas ao VOLTHA. São apresentadas as que serão exploradas nas Seções 3 e 4: 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; 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.

3. Metodologia para os testes com o VOLTHA

Os componentes da arquitetura do VOLTHA e do sistema de virtualização de redes PON são instanciados em clusters *kubernetes*. A comunicação com o ambiente ocorre através dos protocolos *gRPC* para comunicação com o VOLTHA; *OpenFlow* para comunicação entre VOLTHA e ONOS; e *HTTP* para execução de comandos no ONOS proveniente de um orquestrador ou operador externo, consumindo as APIs REST disponíveis. Todo esse conjunto foi explorado na implementação e interação com o ambiente descrito a seguir.

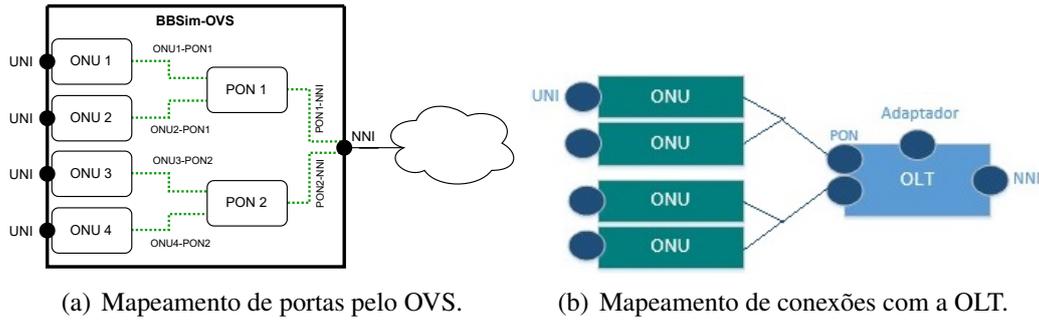
3.1. Ambiente de experimentação

Visando emular um dispositivo compatível com *OpenOLT*, por conseguinte, OLTs, portas PON, ONUs, UNIs (*User Network Interface*) e RGs (*Residential Gateway*), foi usado o BBSim da ONF, que faz parte do projeto VOLTHA. 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, apresentado sucintamente neste trabalho.

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 portas UNIs são mapeadas e adicionadas às interfaces *Ethernet* de entrada e 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. Dessa forma, o tráfego recebido via cliente percorre o circuito ONU-OLT e alcança a rede externa. O mapeamento de portas é representado na Figura 2(a).

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



(a) Mapeamento de portas pelo OVS. (b) Mapeamento de conexões com a OLT.

Figura 2. Arquitetura representativa da topologia simulada no BBSim da rede FTTx com 4 ONUs.

alteração de estado, funções de análise do ambiente são desencadeadas e as configurações são refletidas no OVS alterando, conseqüentemente, o ambiente de experimentação.

3.2. 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 topologia da rede PON utilizada neste trabalho possui 4 ONUs e 1 OLT, como mostra o esquema na Figura 2(b). 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. Os dispositivos da topologia virtualizada criados são listados na Figura 3.

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	
bcecc86-0809-4723-ab22-6b5706818ad2	brcm_openomci_onu	false	cb1971bf-f872-4003-8503-4fe741c3756e	BBSM000a0102	ENABLED	ACTIVE	REACHABLE	omci-flows-pushed
ffdc194f-6f67-4528-8e0d-7763932Aa633	brcm_openomci_onu	false	cb1971bf-f872-4003-8503-4fe741c3756e	BBSM000a0002	ENABLED	ACTIVE	REACHABLE	omci-flows-pushed
f6ba0c14-353f-4fa8-911b-7efdd2631e75	brcm_openomci_onu	false	cb1971bf-f872-4003-8503-4fe741c3756e	BBSM000a0001	ENABLED	ACTIVE	REACHABLE	omci-flows-pushed
0eadd01e-95e7-4eae-833f-bc68130771a9	brcm_openomci_onu	false	cb1971bf-f872-4003-8503-4fe741c3756e	BBSM000a0101	ENABLED	ACTIVE	REACHABLE	omci-flows-pushed

Figura 3. Topologia virtualizada da rede FTTx.

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 4, observam-se os pontos de terminais autenticados pelo controlador e disponíveis para a ativação de clientes.

```

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 4. Pontos de terminais autenticados para operação de serviços.

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. A aplicação das configurações é feita via REST a partir de um arquivo com formato *.json* enviado e armazenado no banco de dados do controlador.

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

4. Resultados

Após a ativação do serviço e provisionamento do assinante, obtém-se os dados conforme mostrado na Figura 5(a). 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.

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 foram abstraídas e representadas no esquema da Figura 2(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 5(b)

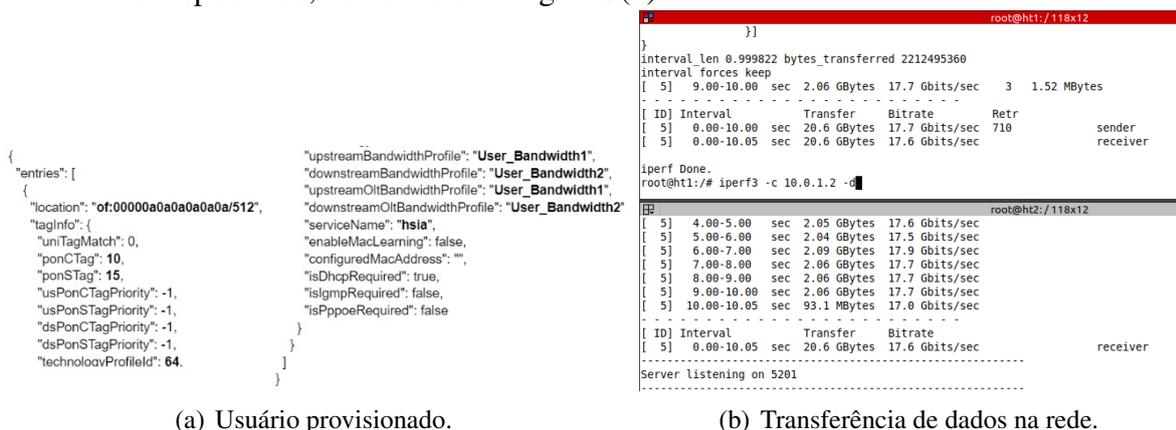


Figura 5. Respostas após a implementação do plano de controle e dados, por meio do ONOS e OVS, respectivamente.

Dado o processo de provisionamento do assinante, o fluxo direcionado para a execução da camada de gerência do sistema é mostrado na Figura 6, sendo representados (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.

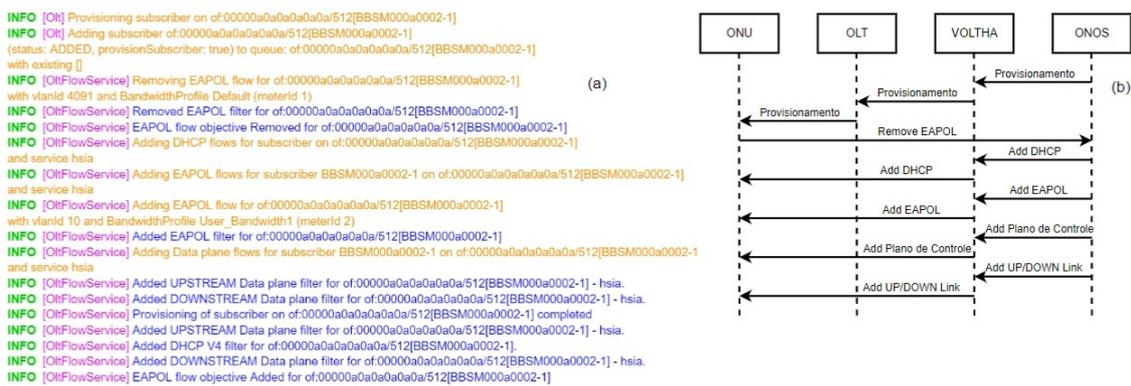


Figura 6. Fluxo de provisionamento da camada de controle. (a) Saída da CLI do ONOS, (b) Fluxograma representativo dos processos apresentados em (a).

5. Conclusões e trabalhos futuros

Por meio do presente trabalho, estabeleceu-se uma rede FTTx com 4 possíveis usuários a serem provisionados. Para isso, 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. Portanto, o estudo demonstrou a viabilidade da pilha de software utilizada pela arquitetura do VOLTHA no contexto SD-PON, mostrando-se capaz de compor o ambiente de testbed do projeto OpenRAN@Brasil.

Portanto, há possíveis trabalhos futuros que incluem atividades tal como o desenvolvimento e testes de controle e orquestração para integrar diferentes domínios tecnológicos, como 5G, FTTx, WDM; e consolidação de sistema OSS/BSS para o uso de padrões abertos e desagregados.

Referências

- Casado, M., Foster, N., and Guha, A. (2014). Abstractions for software-defined networks. *Commun. ACM*, 57(10):86–95.
- Hakiri, A. and Berthou, P. (2015). Leveraging sdn for the 5g networks: Trends, prospects, and challenges. *Software Defined Mobile Networks (SDMN) Beyond LTE Network Architecture*, pages 61–80.
- Open Network Foundation (2023). Voltha. disponível em: <https://opennetworking.org/voltha/>. acessado em : 07/02/2023.
- VOLTHA (2023). Voltha documentation. disponível em: <https://docs.voltha.org/master/index.html> . acessado em : 18/02/2023.