

Laboratório de Roteamento BGP para Ambiente Educacional no testbed Fibre

Gracieth Valenzuela¹, Leonardo Amaral²

¹Instituto Federal de Mato Grosso do Sul (IFMS)
Caixa Postal 79.200-000 – Aquidauana – MS – Brazil.

²Instituto Federal de São Paulo (IFSP)
Caixa Postal 13.069-901 – Campinas – SP, Brazil.

gracieth.valenzuela@ifms.edu.br, contato@leonardoamaral.com.br

Abstract. *This article describes the implementation of an laboratory of dynamic routing for the purpose of helping Computer Network teachers to have, in classes, an environment closer to reality for students experimentals and demonstrations. The used tools were FIBRE testbed infrastructure - an education project composed by 15 test places where complex scenarios can be developed in an environment close to production ones - and the Border Gateway Protocol (BGP) - the most used protocol in different Autonomous Systems connection.*

Resumo. *Este artigo visa descrever a implementação de um laboratório educacional de roteamento dinâmico com finalidade de auxiliar os professores das disciplinas de Redes de Computadores a ter um ambiente mais próximo da realidade para demonstração e operação pelos alunos. As ferramentas utilizadas foram a Infraestrutura do testbed FIBRE - uma estrutura educacional composta por 15 locais de testes onde cenários mais complexos podem ser desenvolvidos em ambiente similar ao de produção - e o protocolo de roteamento BGP - protocolo mais utilizado para conexões entre Sistemas Autônomos distintos.*

1. Introdução

No Brasil, segundo [LACNIC 2018], houve um expressivo aumento no registro anual de ASNs no Brasil nos últimos 10 anos. Este aumento indica um forte crescimento das redes brasileiras e abre espaço para uma nova demanda de conhecimentos do profissional de redes para trabalhar com sistemas autônomos. Parte deste trabalho está na conexão entre provedores de trânsito e pontos de troca de tráfego (IX) com os provedores de última milha - largamente difundidos no Brasil nos últimos anos, de acordo com [ABRANET 2018].

Esta demanda requer que novos laboratórios mais condizentes com a realidade sejam um recuso indispensável no ensino superior tecnológico. Desta forma, um laboratório utilizando o sistema BGP Quagga/FRR em conjunto com uma rede Layer2 moldavel no sistema Fibre foi desenvolvida e será apresentada neste artigo.

2. Testbed Fibre

O Testbed Fibre teve inicio com o objetivo de implantar uma nova instalação experimental para pesquisas futuras na Internet no Brasil. O testbed Fibre foi originalmente composto por 13 ilhas de experimentação localizados em 3 paises da Europa e no Brasil.

No Brasil estavam localizadas 10 ilhas, após o encerramento do projeto em outubro de 2014, as instituições brasileiras que eram membros do projeto assumiram a infraestrutura da FIBRE para oferecer o testbed como serviço. Foram propostos novas políticas, novo modelo de governança, regras de admissão e processos de operação de serviços. No entanto, a marca e o nome do testbed permaneceram os mesmos, porém com o seguinte significado: Future Internet Brazilian Environment for Experimentation [Fibre 2018].

Atualmente, funciona como um laboratório virtual de grande escala com acesso disponível para estudantes e pesquisadores testarem novas aplicações e modelos de arquitetura de rede, a infraestrutura FIBRE está composta por 11 locais de testes que também são denominados de ilhas. Cada ilha possui um conjunto de dispositivos de rede para suportar experimentos. Os equipamentos são conectados por uma rede de sobreposição no backbone RNP composta por duas camadas separadas de rede: o de controle e o de experimento. Esta rede é denominada de FIBREnet. O modelo de governança permite que cada instituição participante tenha autonomia sobre seus recursos locais e podem usar recursos de outras ilhas para configurar experimentos de rede.

Segundo [Fibre 2018], não é necessário hospedar uma ilha experimental para usar o testbed. O testbed FIBRE está aberta para ser usada por qualquer pesquisador, professor ou aluno de qualquer instituição. E os objetivos do FIBRE consistem em:

- Construção de uma plataforma de testes da Internet no Brasil, federada com outras iniciativas mundiais de teste;
- Fornecer uma plataforma de grande porte para promover a pesquisa da Internet no Futuro no Brasil;
- Incentivar professores a utilizar o testbed para as aulas em redes para promover uma nova geração de pesquisadores.

Sendo este último o objetivo do laboratório versátil para roteamento proposto neste artigo, que consiste em configurar e apresentar um ambiente para que os professores possam utilizar o testbed para as suas aulas.

3. Um laboratório versátil para roteamento

3.1. Requisitos

- Acesso ao controlador OCF (OFELIA Control monitoring Framework) hospedado pelo FIBRE;
- Ilha com imagem Debian Jessie, com versão de kernel maior ou igual à 3.10 conforme a documentação em [Inc 2018b, Docker].

3.2. Infraestrutura utilizada

O testbed FIBRE: O testbed FIBRE [Fibre 2018] utiliza uma implementação flexível de rede - o OpenFlow. Este recurso permite que diversas redes em Layer 2/OSI possam ser implementadas e mapeadas para as máquinas virtuais de um projeto, permitindo diversas possibilidades de simulações de redes físicas. Desta forma, possível criar os cenários físicos existentes e aplicar neles o modelo de roteamento desejado. Para a demonstração das capacidades deste laboratório, implementaremos um cenário simples de conexão entre dois ISPs distintos.

3.3. Softwares utilizados

Docker: O Docker [Inc 2018a] é um software que permite isolar softwares em seus próprios espaços operacionais, permitindo que softwares com requisitos de ambiente diferentes possam rodar em um mesmo host sem necessidade de virtualização.

Docker Compose: O Docker Compose é um sistema de provisionamento de ambientes docker com as imagens prontas e parametros já pré determinados. O conjunto Docker + Docker Composer permite que independente do local de execução, ao executar as instâncias, elas estarão tal como um softrouter pronto e com a configuração que o usuário desejar. Todas as aplicações utilizadas neste projeto utilizam deste recurso para fácil replicabilidade, melhoria de entendimento e independência de estrutura de virtualização.

RYU: O sistema RYU [Community 2017] é um controlador de OpenFlow programável. Dele, usaremos neste artigo um simples switch L2, porém as imagens utilizadas permitem configurar qualquer estrutura de L2 nas interfaces interligadas com OpenFlow.

Quagga/FRR: O Quagga é um tradicional software de roteamento dinâmico que pode ser executado em qualquer sistema Linux. O FRRouting - ou FRR - é um *fork* do Quagga que visa modernizar o Quagga para algumas extensões modernas. Não há diferenças em qual dos dois será utilizado pois são retrocompatíveis em termos de configurações.

3.4. Projeto da implementação

Os pontos mais centrais do projeto são os seguintes:

- A estrutura de controle do Layer 2/OSI e a interligação das máquinas virtuais
- O conteúdo dos containers e como os mesmos irão se comportar

A distinção de um laboratório convencional está em permitir modelagens de L2/L3 OSI distintas e com padrões reais de comportamento, mudanças de comportamento em tempo real com o sistema rodando - controlado pelo OpenFlow e mudanças no roteamento através do console de configuração. Os containers permitem fácil reconfiguração caso necessário sem a obrigatoriedade de reinstalação ou danos ao sistema, economizando tempo e aumentando a capacidade de virtualização.

O experimento descrito neste artigo consiste em um pequeno experimento BGP entre dois AS distintos - neste caso, um eBGP. O fluxo esperado desta demonstração é que um bloco alocado para um AS consiga se comunicar com um outro bloco alocado para um outro AS - simulando um Peering entre 2 ISPs. Este experimento, uma vez colocado, permite adicionar novos roteadores, ou VLANs com imensa facilidade, podendo escalar rapidamente durante as aulas caso o professor deseje. Há um objetivo claro em manter cada AS em um virtualizador diferente, que consiste em demonstrar os efeitos da comunicação real entre redes, medições de latência, jitter, performance, convergencia de rotas, etc.

Na Figura 1, descrevemos a topologia proposta do experimento para o eBGP. Os equipamentos desenvolvidos como containers Docker estão em vermelho e equipamentos externos ou fora de containers estão em azul.

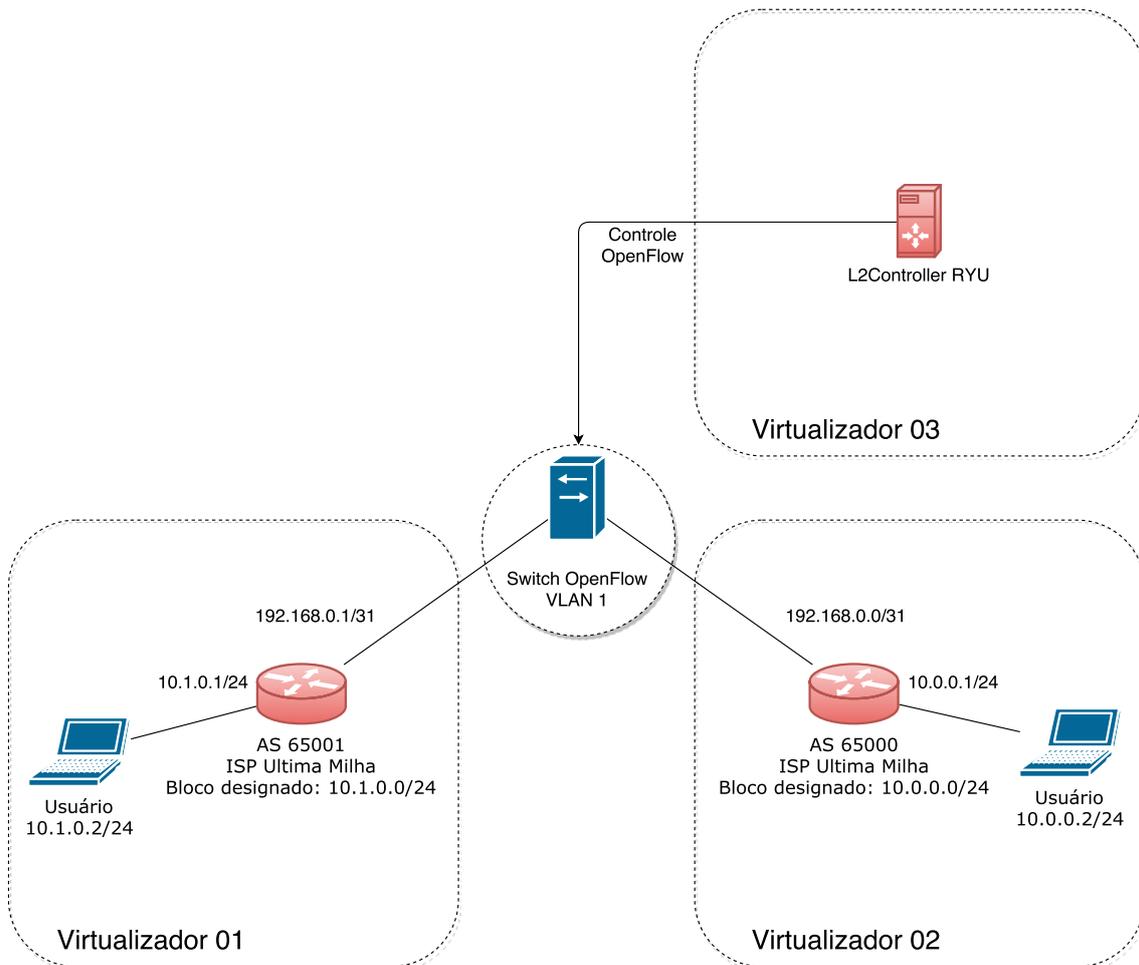


Figura 1. Representação lógica da estrutura de demonstração.

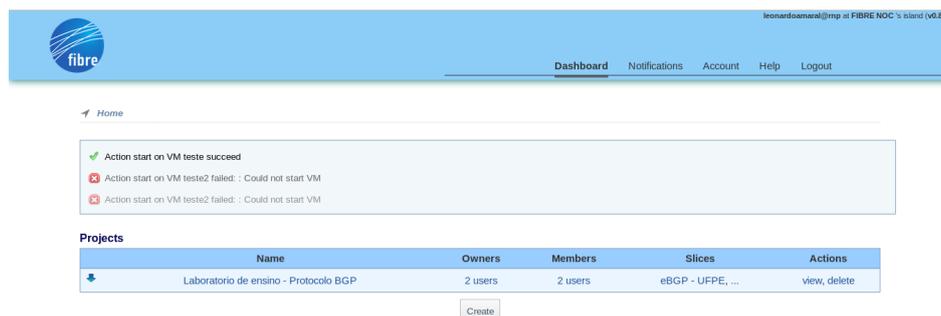
4. Sistema de Implementação

A descrição da implementação se divide em duas etapas, descritas a seguir.

4.1. Passos no sistema OCF

A configuração no sistema OCF se dá da seguinte forma:

1. Solicitar a criação do Projeto, conforme imagem abaixo:



2. Adicionar ao menos um agregado OpenFlow e um Agregado Virtualization.

Neste caso temos dois agregados e dois slices. Para adicionar um slice, basta clicar em “*Create Slice*” e preencher os campos:

Aggregates

Name	Type	Location	Description	Size	Managers	Status	Actions
UFPE OpenFlow	OpenFlow Aggregate	Recife-BR	TBD	53	fibre		remove
UFPE Virtualization	Virtualization Aggregate	Recife-BR	TBD	69	fibre		remove
UFBA Openflow	OpenFlow Aggregate	Salvador - BA	TBD	24	fibre		remove
UFBA Virtualization	Virtualization Aggregate	Salvador - BA	Virtualization Aggregate	167	fibre		remove

Slices

Name	Description	Size	Owner	Reserved?	Actions
eBGP - UFPE	eBGP - UFPE	0	leonardoamaral@mp		view, delete
eBGP - UFBA	eBGP - UFBA	0	leonardoamaral@mp		view, delete

Ao terminar o preenchimento, deverá ser salvo clicando em “*Save*”:

Create Slice in Project Laboratorio de ensino - Protocolo BGP

Name:

Description:

3. Cada slice deve ter ao menos um agregado OpenFlow e um Agregado Virtual.

Para adicionar um agregado ao slice, basta clicar em “*Add an Agregate Manager to the current slice*”:

Slice eBGP - UFPE in Project Laboratorio de ensino - Protocolo BGP

Slice status	Description	Management
eBGP - UFPE		<input type="button" value="Start Slice"/> <input type="button" value="Stop Slice"/>

Physical topology

Tip: Move cursor over the icons to get extra information...

Nothing to show yet. Add Aggregates to the slice.

Slice AMs and resource details

4. Para seleccionar o slice, basta clicar em Select e ao finalizar a adição dos Slices, clicar em “Done”:

Add an Aggregate for Slice eBGP - UFPE

Name	Type	Location	Description	Size	Managers	Status	Actions
UFPE OpenFlow	OpenFlow Aggregate	Recife-BR	TBD	53	fibre	✓	Select
UFPE Virtualization	Virtualization Aggregate	Recife-BR	TBD	68	fibre	✓	Select
UFBA Openflow	OpenFlow Aggregate	Salvador - BA	TBD	24	fibre	✓	Select
UFBA Virtualization	Virtualization Aggregate	Salvador - BA	Virtualization Aggregate	167	fibre	✓	Select

5. Com os agregados adicionados, precisamos adicionar as máquinas virtuais. O procedimento será ilustrado com a criação do controlador OpenFlow, devendo o mesmo procedimento ser seguido em todas as criações de máquinas.

- (a) Para adicionar a máquina virtual, basta clicar em “Create VM” na sessão “Computational resources” após escolher o servidor que irá hospedar a máquina:

The screenshot shows the 'Computational resources' section for the 'Virt. Aggregate: UFPE Virtualization'. It displays a table with columns: Server Name, Virt. Tech., Operating System, CPU, Memory, Disc, and Update. The table lists four servers: UFPE_whitebox1A, UFPE_whitebox2, UFPE_whitebox1B, and UFPE_Pronto, all using XEN technology and GNU/Linux Debian (7.0) OS. Below the table, there is an 'Actions' section with a dropdown menu set to 'UFPE_whitebox1A' and a 'Create VM' button. A 'Remove from slice' section with a 'Remove AM' button is also visible.

- (b) A próxima tela irá apresentar o nome do servidor que está sendo criada a VM.

Os campos a serem preenchidos são o “Name” e o “Memory” após escolher o servidor que irá hospedar a máquina. O campo “Disc Image” deve sempre ser ajustado para “Debian Jessie” ou mais novo:

Create a new Virtual Machine in server UFPE_whitebox1A

The screenshot shows the 'Create a new Virtual Machine' form. It includes the following fields: 'Disc Image' (set to 'Debian Jessie'), 'Name' (set to 'ControladorL2'), 'Memory' (set to '128'), 'HD Setup Type' (set to 'File Image'), and 'Virtualization Setup Type' (set to 'Paravirtualization'). At the bottom, there are 'Done' and 'Cancel' buttons.

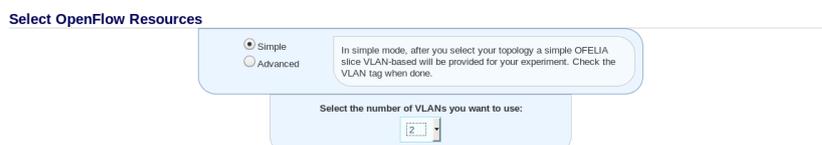
6. Após a configuração da máquina, teremos o seguinte resultado:

O procedimento de criação de máquinas está concluído e as novas máquinas deverão ser criadas tal como demonstrado no passo anterior.

7. Na configuração do OpenFlow, faremos a configuração do “*OpenFlow controller*” e das “*Actions*”

8. (a) Em “*OpenFlow controller*” ao clicar no botão “*Update controller*”, selecionaremos a máquina “*ControladorL2*” na sessão “*Controller IP (from current slice)*” com o campo “*Protocol*” em “*tcp*” e o campo “*Controller port*” com o padrão “*6653*” clicando em “*Save*” ao finalizar:

- (b) Em “Actions” ao clicar no botão “Define flowspace (virtual topology)”, deverá ser selecionado o modo “Simple” com duas VLANs:



Em “OpenFlow Aggregate Nome da agregação” a seleção deve se dar entre outras duas VMs para que as VLANs possam comunicar de um virtualizador ao outro:

Datapath ID	Connections (Port and Remote Port)
00:00:0c:c4:7a:e3:d3:74	<input type="checkbox"/> Port 65534
	<input type="checkbox"/> Port 5 Connected to datapath 67:8c:08:9e:01:62:d6:81 at port 5.
	<input type="checkbox"/> Port 1
	<input type="checkbox"/> Port 4 Connected to datapath 00:00:0c:c4:7a:e3:d3:d6 at port 2.
	<input type="checkbox"/> Port 6 Connected to VMs hosted at UFPE_whitebox2 to the VM's interface eth1
	<input type="checkbox"/> Port 7
	<input type="checkbox"/> Port 2 Connected to datapath 00:00:0c:c4:7a:e3:d3:d7 at port 2.
00:00:0c:c4:7a:e3:d3:d6	<input type="checkbox"/> Port 3
	<input type="checkbox"/> Port 65534
	<input type="checkbox"/> Port 4 Connected to datapath 00:00:0c:c4:7a:e3:d3:d7 at port 4.
	<input type="checkbox"/> Port 1 Connected to datapath 67:8c:08:9e:01:62:d6:81 at port 3.
00:00:0c:c4:7a:e3:d3:d7	<input type="checkbox"/> Port 2 Connected to datapath 00:00:0c:c4:7a:e3:d3:74 at port 4.
	<input checked="" type="checkbox"/> Port 3 Connected to VMs hosted at UFPE_whitebox1B to the VM's interface eth1
	<input type="checkbox"/> Port 1 Connected to datapath 67:8c:08:9e:01:62:d6:81 at port 1.
	<input type="checkbox"/> Port 2 Connected to datapath 00:00:0c:c4:7a:e3:d3:74 at port 2.
	<input type="checkbox"/> Port 4 Connected to datapath 00:00:0c:c4:7a:e3:d3:d6 at port 4.

Ao terminar, clicar em Next.

Estes passos elucidam o mecanismo de configuração do OCF. A criação de VMs para o experimento será escopo do próximo tópico.

4.2. Passos em VMs

Existem para este experimento dois tipos de VMs - A Controladora OpenFlow e as VMs de laboratório, que são configuradas de acordo com a necessidade da estrutura a simular.

A instalação do sistema operacional não será abordada neste documento por já vir pronta nas imagens do projeto Fibre. Todos os arquivos necessários para a execução dos laboratórios estão disponíveis no repositório <https://git.rnp.br/fibreTutorials/bgpFibreTutorial/>.

4.2.1. Estrutura de container - Docker

Os comandos necessários para a instalação do Docker estão disponíveis no arquivo “*installDocker.sh*” no repositório GIT do projeto e devem todos serem executados como o usuário *root*.

Cada parte do experimento terá uma pasta própria que será apontada durante a documentação. Nesta pasta estarão - salvo indicado - os scripts “*stopImage.sh*” e “*upgradeImage.sh*”. Estes scripts são usados respectivamente para parar e iniciar/atualizar

um container. Os dados de configurações e arquivos utilizados estarão em suas respectivas pastas.

O composer das imagens - salvo em caso de aviso explícito - são configurados para boot automático, não sendo necessária nenhuma intervenção no sistema para auto inicialização dos experimentos.

4.2.2. Controladora OpenFlow

A controladora OpenFlow é uma máquina muito simples e a única instrução de instalação é instalar o docker e executar o script *"upgradeImage.sh"* disponível na pasta.

4.2.3. Serviço BGP

O sistema de BGP é o padrão de mercado Quagga. Consiste no serviço de roteamento e no serviço de forwarding. O modelo do experimento para cada instância de roteador é ter o serviço BGP exportado via Docker e ouvindo em uma VLAN enquanto há o IP disponível na segunda VLAN - de acordo com o modelo configurado no item 8b, separando a VLAN de sessão e de transporte.

Ou seja, cada roteador tem uma VLAN para as sessões BGP - que é o local que a sessão BGP ouve e outra VLAN que contem o prefixo anunciado em si. Este sistema está configurado para instalar rotas via Zebra.

Este cenário pode ser utilizado com uma única VLAN, unindo a interface de BGP com a interface de forwarding, a depender da proposta educacional.

Na pasta *"bgp/frr-docker-only/configs/bgpd"* há a configuração das instâncias de roteamento que deve ser configurada pelos alunos ou professores dependendo da atividade. Não há outra necessidade de infraestrutura, pois os arquivos de docker-composer estão automatizados para suprir demandas da estrutura.

Para cada entrada *"services"* no arquivo *'bgp/frr-docker-only/frr/docker-compose.yaml'* haverá um protocolo BGP executando. É recomendada a execução de uma única instância.

Para iniciar o BGP, basta executar o *"bgp/frr-docker-only/startImage.sh"*.

5. Considerações Finais

O sistema aqui descrito permite criar roteadores BGP virtuais de forma prática e rápida, permitindo inclusive testes em casos reais envolvendo a comunicação real entre hosts, diferente do que acontece na simulação. O processo de configuração do ambiente do serviço BGP foi simplificado e automatizado à exaustão com o uso de Docker e do Docker-Compose, tornando sua implementação rápida e sem complicações. Um simples teste de conexão pode ser implementado em questão de poucos minutos e é possível com o sistema descrito por este artigo criar topologias inteiras utilizando o Fibre como plataforma real e organica para o ensino de redes de forma mais próxima da realidade que é encontrada em redes em produção.

Referências

- [ABRANET 2018] ABRANET (2018). Pequenos provedores impulsionam o crescimento do mercado de telecom no brasil.
<http://www.abranet.org.br/Noticias/Pequenos-provedores-impulsionam-o-crescimento-do-mercado-de-Telecom-no-Brasil-1514.html>. Data de acesso: 2018-03-31.
- [Community 2017] Community, R. S. F. (2017). Ryu sdn framework.
<https://osrg.github.io/ryu/>.
- [Fibre 2018] Fibre (2018). Fibre history.
<http://fibre.org.br/about/fibres-history/>. Data de acesso: 2018-03-26.
- [Inc 2018a] Inc, D. (2018a). Docker - build, ship and run any app, anywhere.
<https://www.docker.com/>.
- [Inc 2018b] Inc, D. (2018b). Get docker ce for debian — docker documentation.
<https://docs.docker.com/install/linux/docker-ce/debian/>.
- [LACNIC 2018] LACNIC (2018). Alocaciones de asn por país (región lac).
<http://stats.labs.lacnic.net/REGISTRO/asnpercc.html>. Data de acesso: 2018-03-31.