



## **OpenVirteX no ambiente FIBRE**

RF – Relatório Final

Autor: Nilon Guedes Duarte

Data: 18/01/2016

## 1. Introdução

Cada ilha do FIBRE tem um conjunto de dispositivos com o protocolo OpenFlow, para que mais de um experimento seja realizado simultaneamente é necessário uma camada intermediária. Atualmente o software utilizado é o FlowVisor, um controlador que atua como um *proxy* transparente entre os switches OpenFlow e os controladores dos usuários.

O FlowVisor apresenta problemas de estabilidade, o que motivou esse projeto, uma alternativa para o FlowVisor é o OpenVirteX, um controlador que atua como um virtualizador de rede, podendo criar múltiplas redes virtuais sobre a rede física, cada rede virtual tem seu próprio controlador.

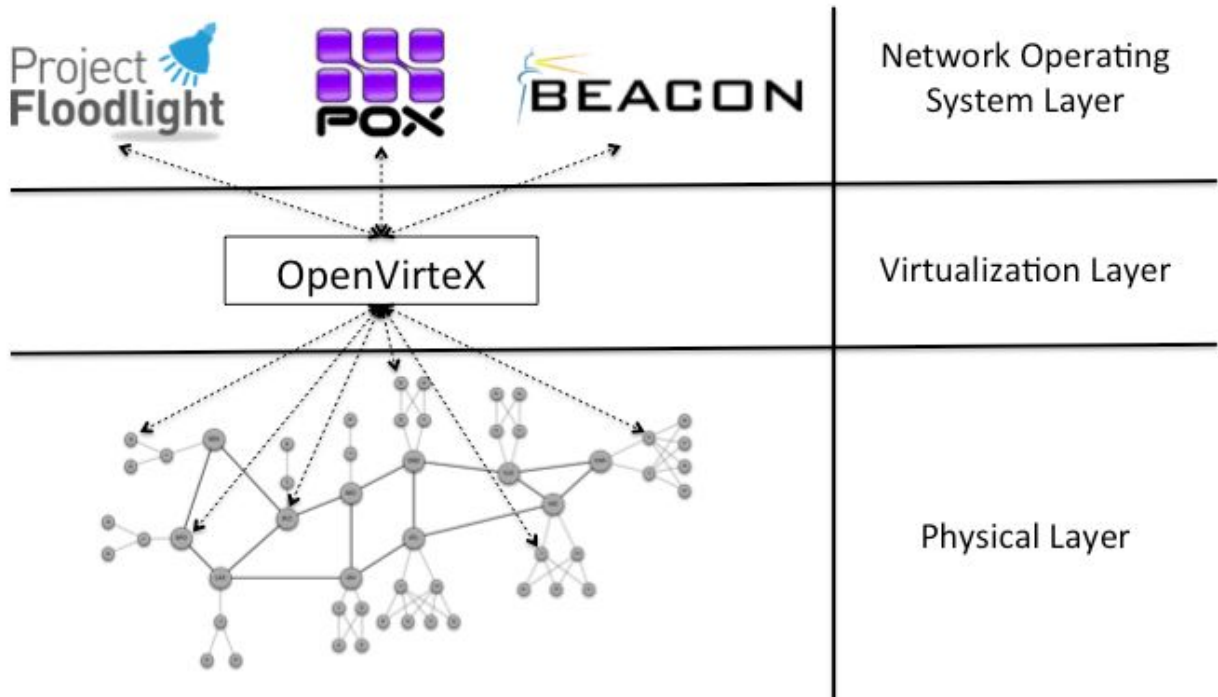


Figura 1: OpenVirteX na camada de virtualização, entre a rede física e os controladores.

## 2. Ambiente de testes

A rede utilizada para o projeto foi a ilha da UFRJ e principalmente o emulador de rede Mininet.

Mininet é um emulador de rede que cria uma rede de *hosts*, *switches* e *links*, proporcionando melhor controle e flexibilidade para o estudo e testes.

Para o controle da rede OpenFlow foi utilizado majoritariamente o controlador Floodlight.

O principal ambiente utilizado foi a máquina virtual disponibilizada no tutorial do próprio OpenVirteX, que vem com o Mininet, Floodlight e OpenVirteX instalados no mesmo sistema. Os testes na ilha da UFRJ foram feitos com o OpenVirteX e o controlador de rede OpenFlow instalados em máquinas virtuais distintas.

### 3. Lições aprendidas

#### 3.1. Visão Geral do Sistema

O OpenVirteX é um virtualizador de rede que trabalha com o conceito de *tenant*, cada *tenant* tem a ilusão de quem possui toda a rede a sua disposição e garante o isolamento do tráfego.

Como virtualizador o OpenVirteX permite a criação de redes virtuais isoladas com topologias determinadas pelo usuário, cada rede virtual pode usar seu próprio sistema operacional de rede e permite alterar a rede virtual durante a execução.

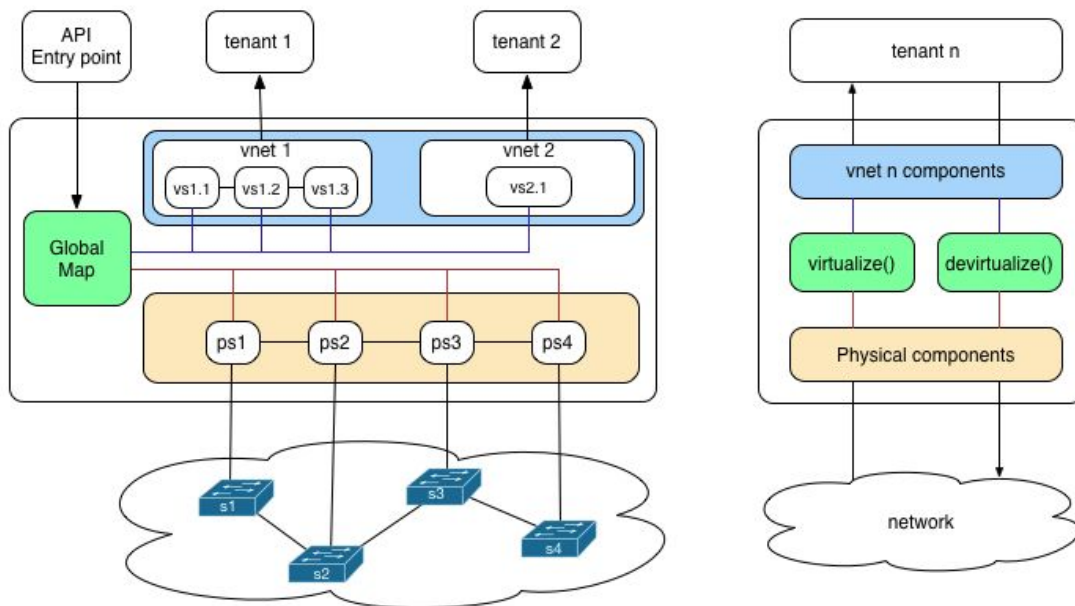


Figura 2: Em amarelo está representada a rede física, em azul a rede virtuais de cada *tenant* e em verde o sistema que faz a conexão lógica entre a rede virtual e a rede física.

Na esquerda da Figura 2 temos duas redes virtuais vnet 1 e vnet 2 com seus *tenants*, cada rede virtual é logicamente separada e cada *tenant* tem apenas a visão da sua topologia virtual. O Global Map é o responsável por receber as chamadas de configuração e estabelecer o mapeamento entre a rede física e as redes virtuais. Na direita está ilustrado a separação física e virtual feita pela virtualização/desvirtualização do Global Map.

A virtualização é a ação lógica que acontece quando uma mensagem OpenFlow sai da rede física para ser entregue ao controlador da rede virtual e a desvirtualização acontece no processo inverso.

Para cada rede virtual o OpenVirteX cria vários mapeamentos, para esse projeto é importante destacar o mapeamento de *link* virtual para *link* físico e o mapeamento de *flow*. No mapeamento de *link* é possível identificar quais *links* físicos são utilizados pelo *link* virtual e o mapeamento de *flow* é feito entre todos os pares de *hosts* e o entre todos os *hosts* e o endereço de *broadcast*, cada um recebendo um identificador de *flow*.

### 3.2. Utilização

O OpenVirteX possui uma API para controle e monitoramento utilizando o protocolo JSON-RPC 2.0. As chamadas para criação da rede virtual e seus parâmetros são as seguintes:

- createNetwork <protocol> <controller\_urls> <ip\_network> <ip\_mask>
  - Cria uma rede virtual com protocolo e endereço do controlador especificado
  - Endereço da rede IP e máscara são exigidos mas não estão implementados
  - Retorna o identificador do *tenant* da rede que é usado em todas as demais chamadas
- createSwitch [options] <tenant\_id> <physical\_dpids>
  - Cria um *switch* virtual para o *tenant* especificado
  - Retorna o identificador do *switch* virtual
- createPort <tenant\_id> <physical\_dpids> <physical\_port>
  - Cria uma porta virtual no *switch* virtual correspondente ao *switch* físico especificado
  - Retorna o número da porta virtual
- connectLink <tenant\_id> <src\_virtual\_dpids> <src\_virtual\_port> <dst\_virtual\_dpids> <dst\_virtual\_port> (routing mode and the number of backup routes)
  - Cria um *link* entre os dois *switches* virtuais especificados
- connectHost <tenant\_id> <virtual\_dpids> <virtual\_port> <host\_mac>
  - Conecta um *host* em um *switch* virtual

### 3.3. Virtualização de Endereço IP

A colisão entre os *tenants* é evitada criando para cada *host* um endereço virtual (único dentro da rede virtual) e um endereço físico (único dentro da rede física). A tradução entre os endereços garante que o controlador de cada *tenant* possa lidar com os fluxos seguindo seu próprio esquema de endereçamento, pois na criação da rede virtual é possível que o endereço IP dos *hosts* se sobreponham entre em diferentes redes virtuais, isto é, dois *hosts* podem ter o mesmo endereço IP se estiverem em redes virtuais distintas. Também garante que os *switches* possam diferenciar o tráfego de diferentes *tenants*.

Por padrão, o primeiro octeto do endereço IP físico é usado para codificar o identificador do *tenant*.

### 3.4. Codificação de Endereço MAC

Os endereços MAC da fonte e do destino são utilizados pelo OpenVirteX para codificar informações. O endereço MAC é formado pelo ON.Lab's OUI (A4:23:05) e 3 bytes para codificar as seguintes informações:

- Identificador do *Tenant* - Primeiro *byte* da fonte
  - Isola os *tenants* em um *link* virtual
- Identificador do *Link* virtual - Últimos 2 *bytes* da fonte e meio *byte* do destino
  - Permite identificar a qual *link* virtual a mensagem pertence, pois *links* virtuais distintos podem estar utilizando o mesmo *link* físico
- Identificador do *Flow* - Últimos 2 *bytes* e meio do destino
  - Permite identificar o fluxo que entrou em um *link* virtual e recuperar ao sair do *link* virtual.

### 3.5. Loop de Eventos - O primeiro problema

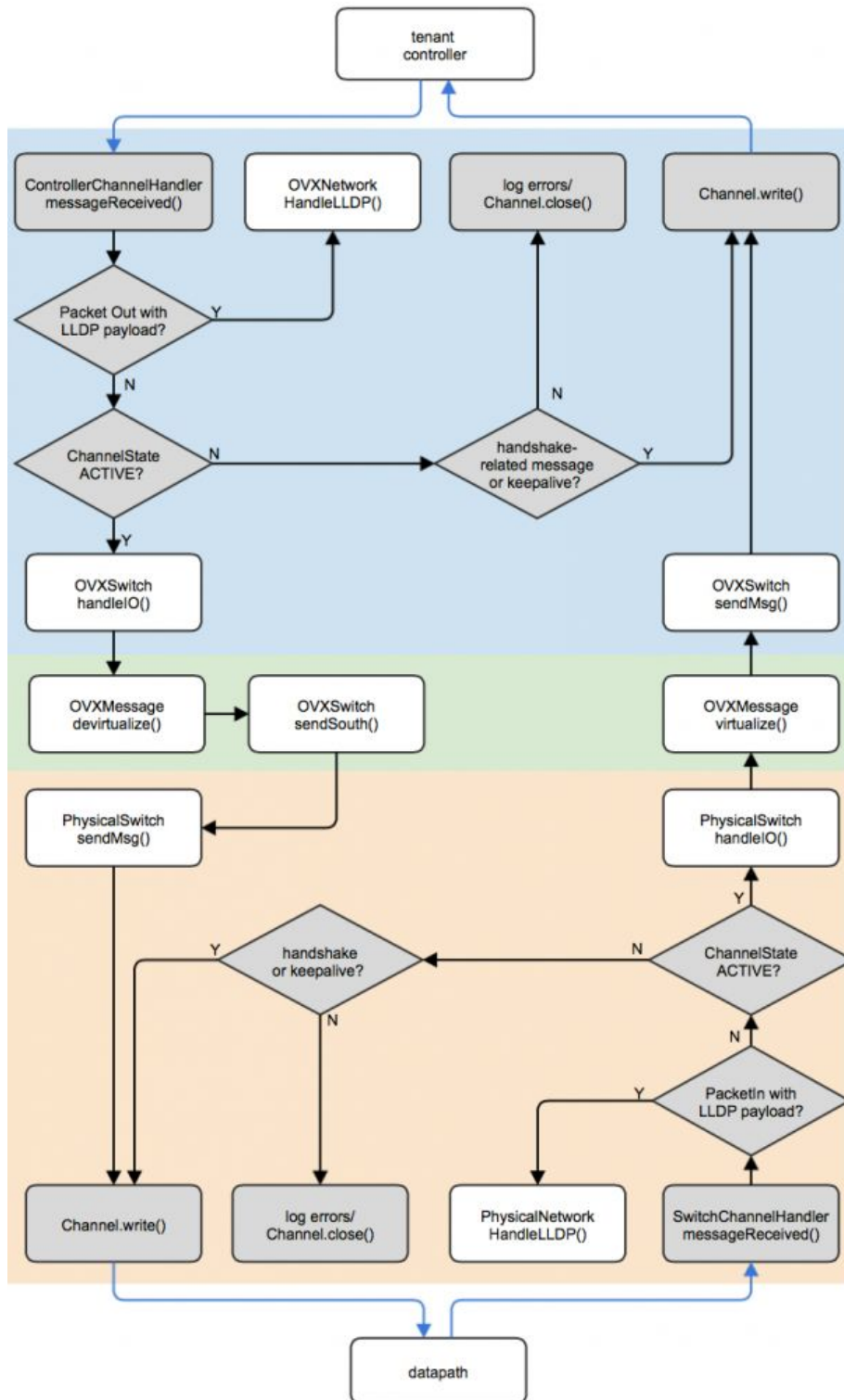


Figura 3: Em azul os componentes virtuais, em verde os componentes do Global Map e em laranja os componentes da rede física. As setas em azul representam os canais OpenFlow.

O loop de eventos é importante para entender como o OpenVirteX trata o LLDP (*Link Layer Discovery Protocol*), aqui está o primeiro problema para a utilização do OpenVirteX no ambiente federado do FIBRE, o OpenVirteX trata o LLDP vindo dos *switches* ele mesmo (bloco *PhysicalNetworkHandleLLDP()*), sem passar para o controlador, isso causa conflito quando duas redes utilizando OpenVirteX estão conectadas, pois os pacotes LLDP enviados pelos *switches* de uma rede serão recebidos na outra rede, exemplo:

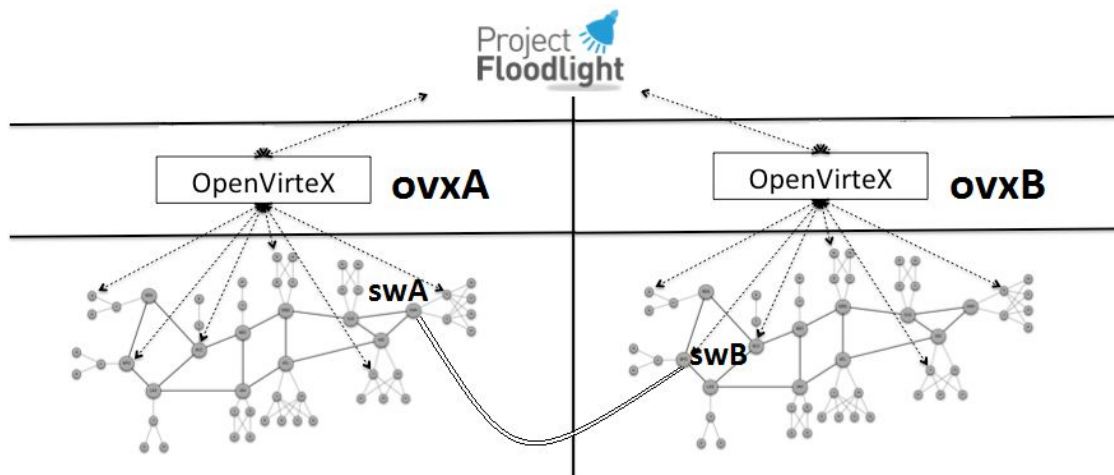


Figura 4: Ilustração de duas redes utilizando OpenVirteX

- Duas redes utilizando OpenVirteX, ovxA e ovxB
- As rede são conectas entre si pelos *switches*, swA da rede ovxA e swB da rede ovxB
- swA envia LLDP para swB
- swB encaminha o LLDP ao ovxB
- ovxB reconhece o pacote como um pacote LLDP
- ovxB não encontra swA conectado nele, entende como um erro no processamento da mensagem e desconecta o *switch* swB

Essa cadeia de eventos se repete, pois o *switch* swB se conecta novamente ao OpenVirteX.

Mesmo que o *switch* não fosse desconectado e o *link* entre os *switches* swA e swB fossem identificados pelos OpenVirteX ovxA e ovxB jamais um *link* entre os dois *switches* poderia ser criado, pois para criar o *link* é necessário que ambos os *switches* do *link* estejam criados na rede virtual, como os *switches* estão conectados em OpenVirteX diferentes, não é possível criar o *link*.

### 3.6. Outros problemas

- O identificador de *flow* é um único dentro da rede virtual, criado pelo OpenVirteX entre a fonte e o destino, no caso do destino estar em uma rede diferente da rede da fonte não seria possível identificar o *flow*.
- O identificador de *tenant* é único em cada sistema, quando o pacote passasse de um sistema para outro, o identificador do *tenant* pode não mais corresponder ao mesmo controlador.
- A virtualização de endereço IP cria um IP único dentro da rede física, a comunicação entre as redes pode causar conflito nos endereços.
- Toda a camada de isolamento e abstração usada pelo OpenVirteX para que cada *tenant* tenha a ilusão de que tem toda a rede a sua disposição impossibilita o funcionamento em federação.
- O controlador POX não estava funcionando com o OpenVirteX em meus testes.
- O desenvolvimento está parado, a última atualização foi em 05/01/2015 e todos da equipe estão trabalhando em outros projetos.

### 3.7. Interface Web

Um projeto para utilização do OpenVirteX por uma interface web utilizando a API JSON-RPC foi iniciado, o primeiro protótipo se mostrou funcional mas o desenvolvimento foi interrompido para que fosse feito um estudo mais aprofundado das limitações e dos problemas da implantação em federação.

O protótipo criava a rede, *switches*, portas, *links* e conectava *hosts* de acordo com as escolhas e parâmetros do usuário, a limitação era que a rede virtual apenas poderia ser um subconjunto da rede física, não por uma limitação tecnológica, apenas para simplificar esse primeiro protótipo.

## Welcome to OpenVirtex Control

### Hosts options on next screen

Controlle IP address (dotted)\*

Protocol\*

Controller Port \*

Network host's IP Adress range\*

Hosts IP Mask\*

Choose Switches\*  
 00:00:00:00:00:00:03:00  00:00:00:00:00:00:04:00  00:00:00:00:00:00:02:00  00:00:00:00:00:00:01:00

Choose Switch ports\*  
 00:00:00:00:00:00:03:00 Port 3  00:00:00:00:00:00:04:00 Port 3  00:00:00:00:00:00:02:00 Port 3  00:00:00:00:00:00:01:00 Port 3

Choose links\*  
 00:00:00:00:00:00:01:00/5 - 00:00:00:00:00:00:04:00/3  00:00:00:00:00:00:01:00/3 - 00:00:00:00:00:00:00:00/3

Figura 5: Primeira tela do protótipo da interface web para criar redes virtuais utilizando OpenVirteX

#### 4. Conclusão e avaliação

Inicialmente o OpenVirteX se mostrou muito interessante e promissor, com o conceito de virtualização poderoso e simples, porém, com a impossibilidade da implantação em federação se tornou inviável para o testbed. Para instalação isoladamente em cada ilha ainda se mostra interessante, pode ser instalado sobre o FlowVisor.

Para a implantação em todo o testbed seria necessário um esforço enorme de desenvolvimento para que o OpenVirteX se tornasse um sistema distribuído.

#### 5. Referências Bibliográficas

- OpenVirteX  
[ovx.onlab.us](http://ovx.onlab.us)
- Projeto OpenVirteX no GitHub  
[github.com/opennetworkinglab/OpenVirteX](https://github.com/opennetworkinglab/OpenVirteX)
- OpenFlow Switch Specification -  
[www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.0.0.pdf](http://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.0.0.pdf)
- Flowgrammable  
[flowgrammable.org/sdn/openflow/](http://flowgrammable.org/sdn/openflow/)