MC833A – Programação de Redes de Computadores

Professor Nelson Fonseca

http://www.lrc.ic.unicamp.br/mc833/

Roteiro

- **Objetivo: entender o funcionamento de um controlador SDN e o protocolo openflow.**

- OpenVSwitch
- Controlador SDN (opendaylight)
- Openflow

Introdução

Elementos em cima de uma rede:

VLAN, Roteamento, Firewall, ACLs e Serviços.

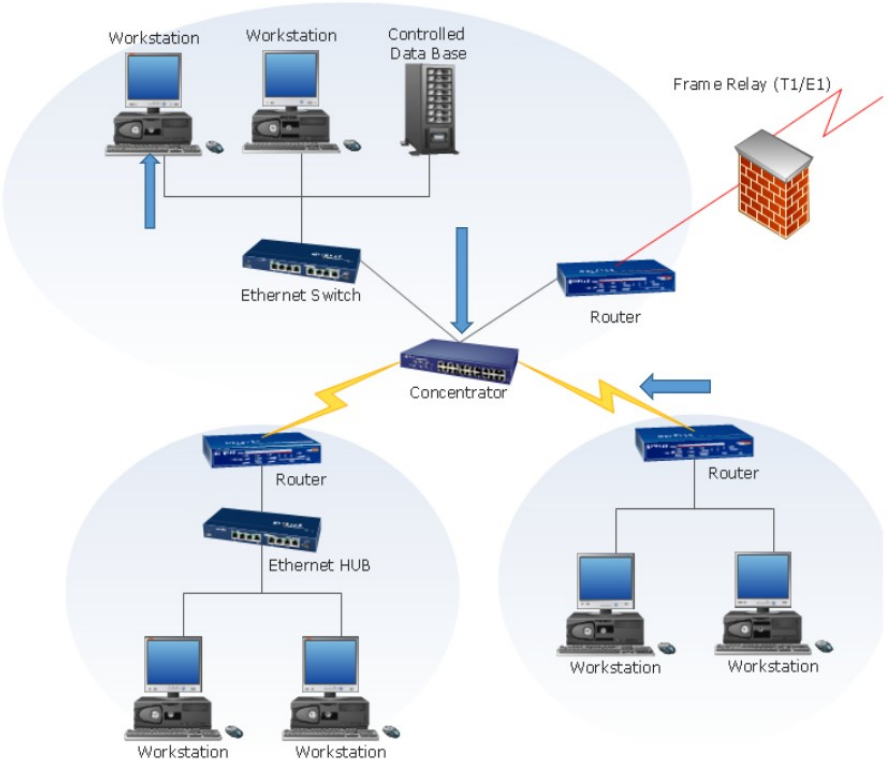


Imagem Slides Prof. Nelson Fonseca (Network Virtualization) ComSoc School

Introdução

Switchs, Routers e Wireless Access Point

Introdução

As redes nas ultimas décadas:

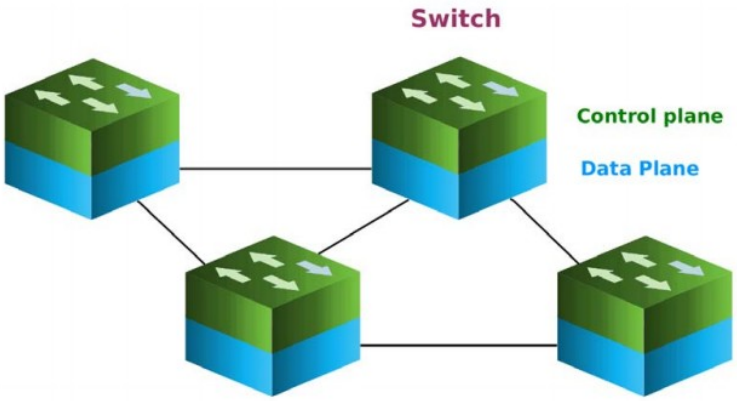
- Modelo TCP/IP sem novas inovações.
- Modelo de tomada de decisão na Internet é descentralizada, sem garantias.
  - Para prover a QoS (Qualidade de Serviço) / QoE (Qualidade da Experiência)
- Equipamentos fechados e sem flexibilidade na camada de controle.
- Novas tecnologias com necessidade de ser testadas (IoT) - ex. e-health
- Compartilhamento de recursos, necessitando de um ambiente ágil (Cloud).

As redes programáveis foi proposta com o objetivo facilitar futuras evoluções nas redes de computadores.

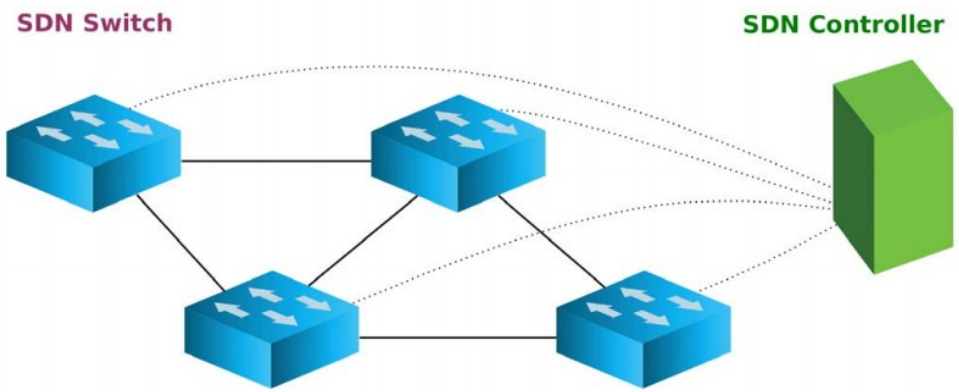
Motivação

- **Mudança dos padrões de tráfego:** servidores distribuídos geograficamente nas nuvens públicas e privadas c/ tráfego extremamente flexível com largura de banda sob demanda.
- **Serviços em nuvens:** atualmente é um recurso muito utilizado e que cresce cada vez mais, onde os usuários esperam ter acesso on-demand para aplicações, infraestrutura e vários recursos.
- **Largura de banda:** estamos na era do “Big Data” que requer processamento paralelo massivo.
- **Complexidade:** dispositivos móveis e outros diversos tipos dispositivos, tornaram a criação de políticas e gerenciamento mais complexos, demorados e manuais.
- **Escalabilidade:** padrão eficaz tráfego que são mais dinâmicos e virtualizados.
- **Dependência de fornecedor:** a tecnologia atual é dependente do fabricante, existe uma falta de padrão e interfaces limitadas a capacidade dos administradores com o seu ambiente de acesso.

Tradicional vs SDN



(a) Traditional architecture



(b) SDN architecture

Imagens do artigo: Distributed SDN Control: Survey, Taxonomy, and Challenges
IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 20, NO. 1, FIRST QUARTER 2018

Definição de SDN

Facilitar a automação das configurações de rede

+

Programar completamente a Rede (> flexibilidade)

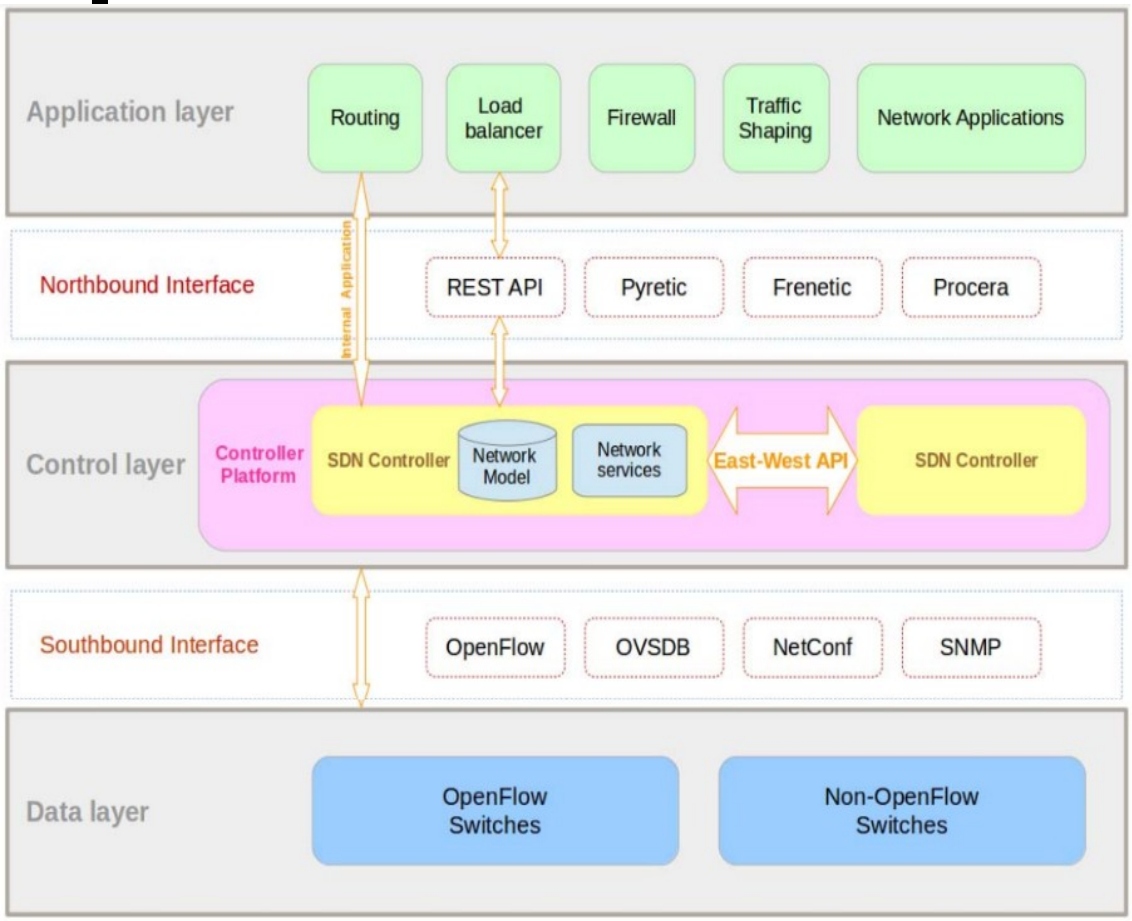
=

Melhorar o gerenciamento

Software Defined Network

Criar aplicações para a camada de controle que conversem e manipulem informações na camada de dados.

Arquitetura SDN



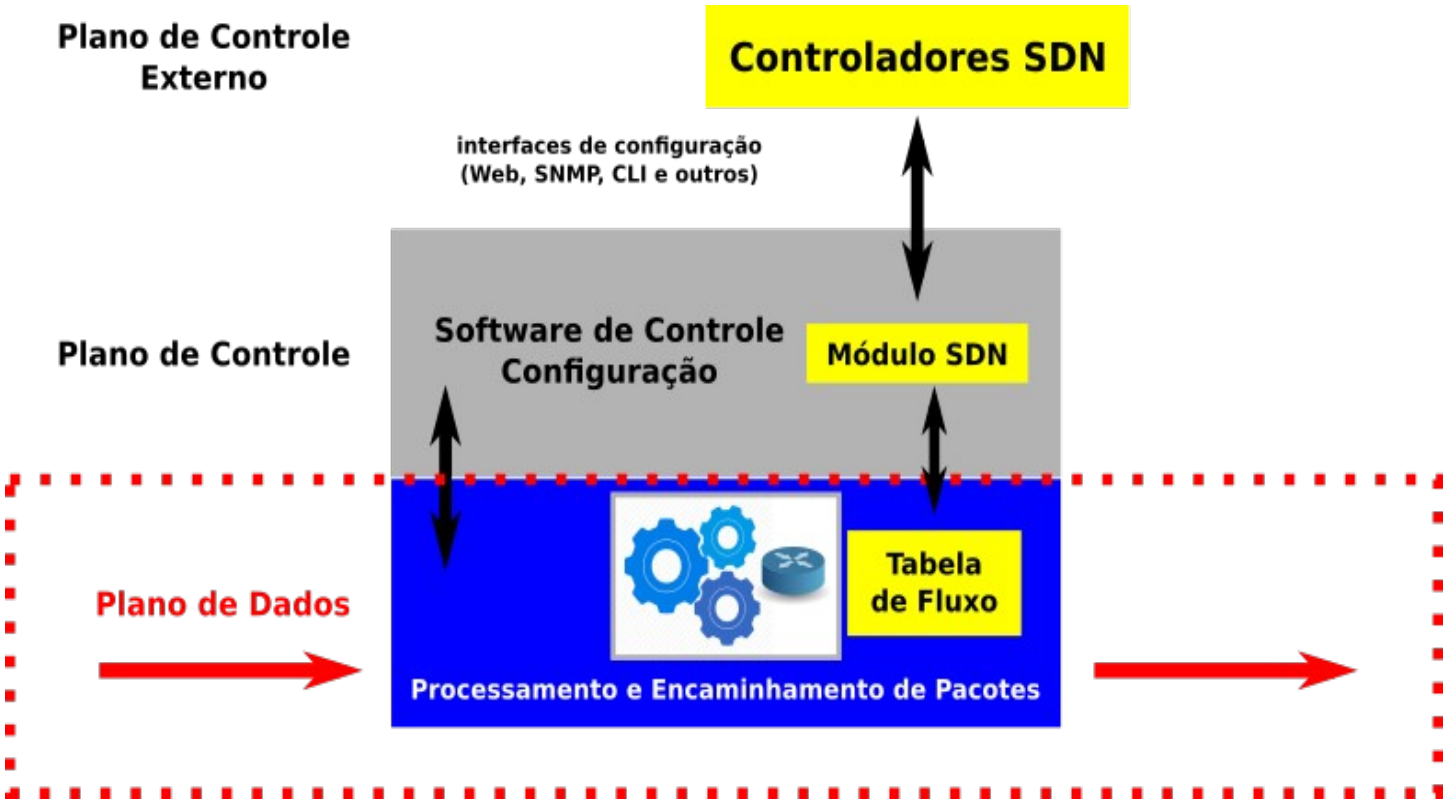
→ **Plano Aplicação**

→ **Plano Controle**

→ **Plano Dados**

Imagens do artigo: Distributed SDN Control: Survey, Taxonomy, and Challenges
IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 20, NO. 1, FIRST QUARTER 2018

Plano de Dados



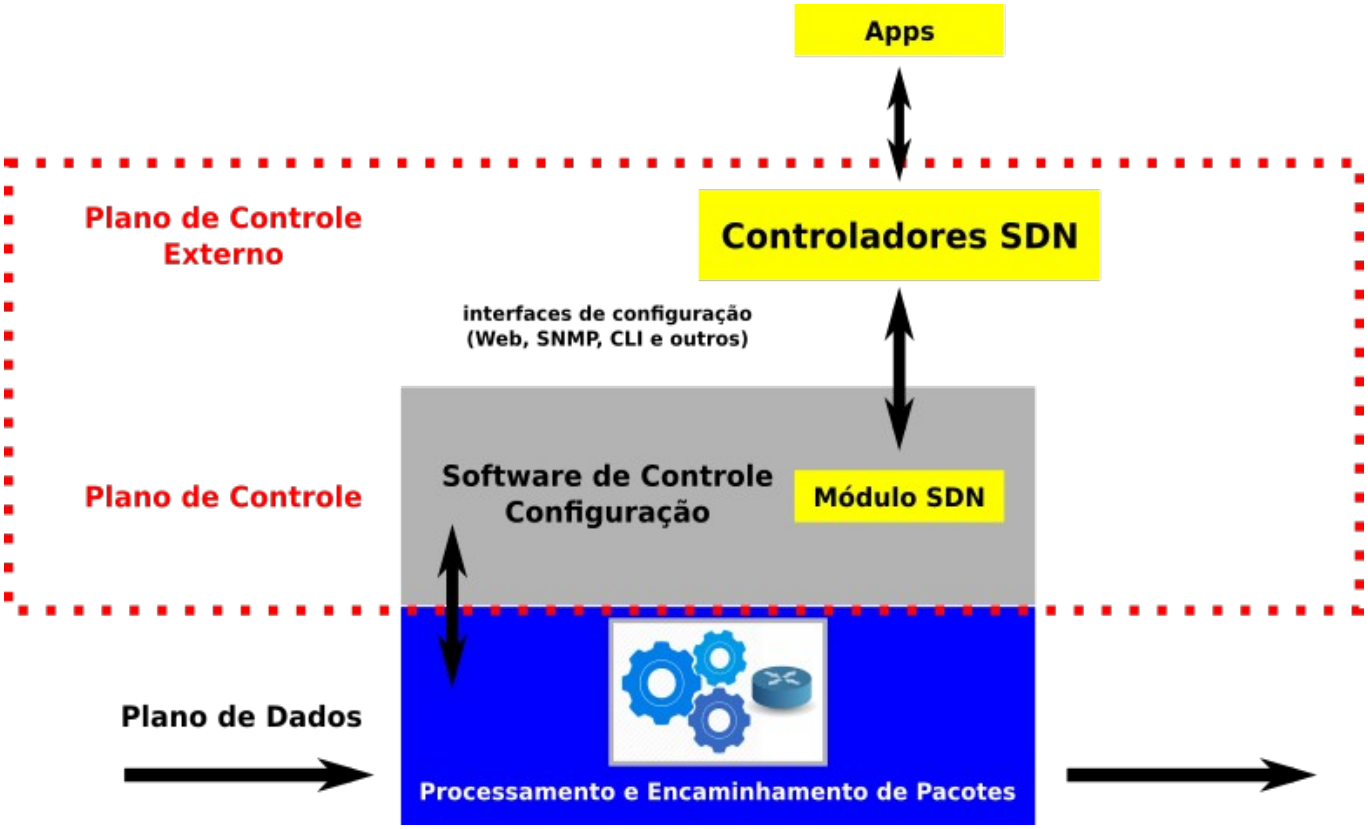
Plano de Dados

- Responsável pelo encaminhamento de pacotes e alguns tipos de processamento.
- Requer acesso remoto do software via interface Southbound de qualquer fornecedor.
- Plano de dados possui as tabelas de fluxos e ações.

Existem diversas implementações para configuração de switches a integrar ao encaminhamento e com o processamento no plano de dados.

- Fabricantes (via web/telnet/API proprietaria) (cada um com sua caracteristica)
- OpenFlow / ForCES
- Netconf
- SNMP
- Outras

Plano de Controle

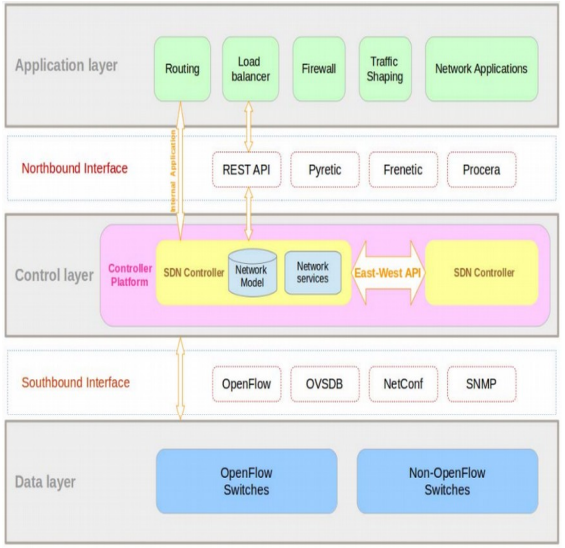


Plano de Controle

O plano de controle tem como objetivo analisar e permitir mover a lógica de tomada de decisão para os **controladores** externos.

E que as aplicações (softwares) possam se comunicar de maneira simples.

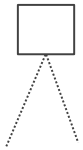
- Baseado nas padronizações em APIs.
- Visão centralizada de todos os equipamentos.
- Análise dos ambientes mais eficiente.



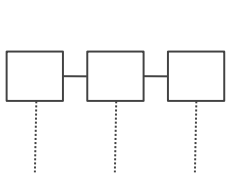
Plano de Controle

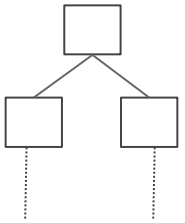
Os controladores SDN são uma parte muito importante, vamos ver algumas características e verificar as possíveis arquiteturas físicas.

- Arquitetura Centralizada.
  - Tendo um único ponto de visão de todo o ambiente.



- Arquitetura Descentralizada. (Plana / Hierarquia)
  - Possui mais de um ponto de processamento e tomada de decisão.





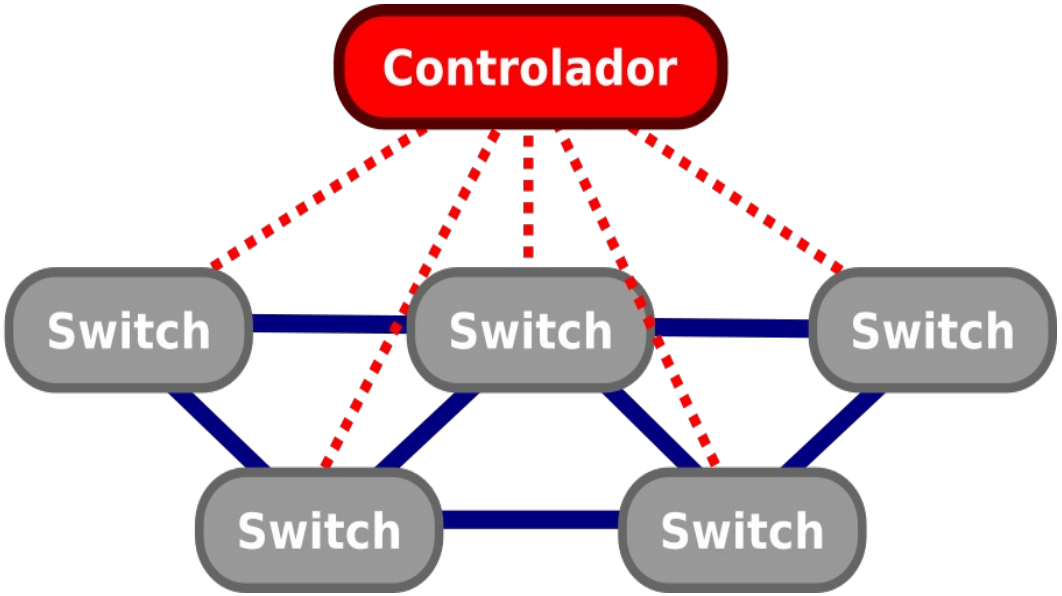
Plano de Controle

- Programa de controles projetados para implementar a lógica e estratégia do controle da rede.
- Aplicativos SDN -> Controladores (APIs) -> Southbound APIs -> HW
- Aplicativos devem definir metas e políticas centralizadas (maneira independente).
- Existem diversas APIs do Northbound:
  - APIs simples e primitivas
    - APIs ad-hoc (proprietarios / fortemente depende ao controlador)
      - NOX em C++ / POX em Python.
    - APIs REST API (serviços WEB) * qualquer linguagem.
      - Floodlight - REST API
  - APIs nivel superior.
    - Linguagens de programação especificas (Frenetic/Procera/Pyretic)
      - Eleva o nivel abstração (dev flexivel para rotinas de controle de rede)

Plano de Controle e Dados

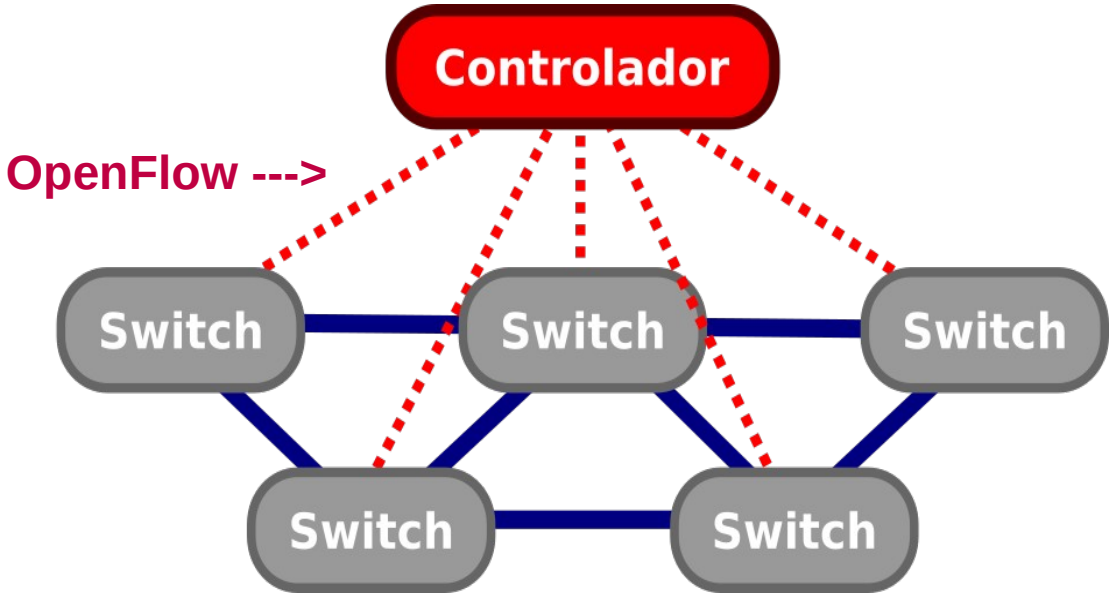
Plano de Controle
(Controle)

Plano de Dados
(Encaminhamento)



OpenFlow

Plano de Controle
(Controle)



Plano de Dados
(Encaminhamento)

OpenFlow

OpenFlow criado originalmente na universidade de Stanford, e é um protocolo de padrão aberto, que surgiu da necessidade dos pesquisadores executarem protocolos experimentais na rede acadêmica.

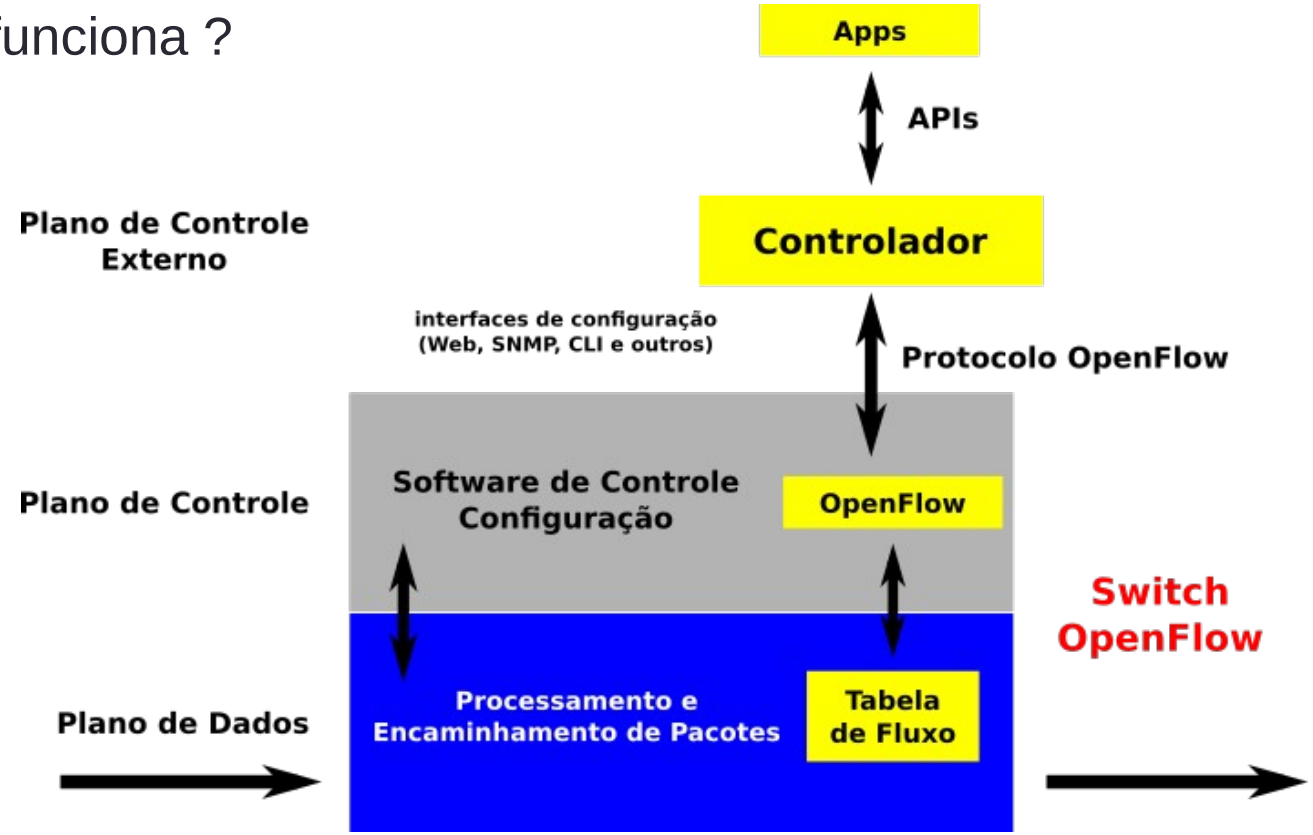
Onde um controlador se comunicava com os dispositivos de rede em uma arquitetura SDN, substituindo todo o plano de controle, e definindo toda a ação que deve ser feita dentro do equipamento.

OpenFlow provê meios de controlar os dispositivos de rede (switches OpenFlow) sem a necessidade dos fabricantes exporem o código de seus produtos (Lara et al., 2014).

Alguns modelos com SDN estavam presos nas limitações do plano de controle dos equipamentos.

OpenFlow

Como funciona ?



Através do protocolo OpenFlow, os controladores podem programar as tabelas de fluxo dos switches.

OpenFlow

Composição dos Switches OpenFlow

Baseando-se na versão 1.0.0, a versão mais difundida do OpenFlow, os switches OpenFlow devem possuir/suportar as seguintes características (McKeown et al., 2008):

- **Tabela de Fluxos** (12 campos classificar o fluxo, ação e contadores);
- **Canal Seguro** (até o controlador);
- **Protocolo** OpenFlow (comunicação).

OpenFlow

Tabela de Fluxos (para tomada de decisão):

Entrada da tabela de fluxos

- Campo dos Cabeçalhos (combinação)
- Ações
- Contadores

OpenFlow

Tabela de Fluxos (para tomada de decisão):

- Campo dos Cabeçalhos (combinação)

Ingress port

Ethernet src / dst / type

VLAN id / pri

IP src / dst / proto / ToS

TCP/UDP src / dst

ICMP type / code

OpenFlow

- **Tipos de ações** (cada entrada tabela de fluxos é associado uma ou mais ações).

Encaminhamento

- Obrigatório
  - ALL - Envia o pacote para todas as interfaces, exceto a interface de entrada;
  - CONTROLLER - Encapsula e envia o pacote para o controlador;
  - LOCAL - Envia o pacote para a pilha de rede local;
  - TABLE - Realiza ações na tabela de fluxos;
  - IN_PORT - Envia o pacote para a porta de entrada;
- Opcional
  - NORMAL - Processa o pacote utilizando um encaminhamento tradicional;
  - FLOOD -Inunda o pacote, sem incluir a interface de entrada, levando em consideração o Spanning Tree.

OpenFlow

- Enfileirar (opcional) - Encaminha o pacote através de uma fila relacionada a uma porta;
- Descartar (obrigatória);
- Modificar campo (opcional):
  - Setar Vlan ID
  - Setar Vlan Priority
  - Separar o cabeçalho da Vlan
  - Modificar endereço MAC de origem
  - Modificar endereço MAC de destino
  - Modificar endereço IP de origem
  - Modificar endereço IP de destino
  - Modificar ToS
  - Modificar a porta de transporte de origem
  - Modificar a porta de transporte de destino

OpenFlow

Contadores (Por tabela, fluxo, porta ou fila)

Counter	Bits
Per Table	
Active Entries	32
Packet Lookups	64
Packet Matches	64
Per Flow	
Received Packets	64
Received Bytes	64
Duration (seconds)	32
Duration (nanoseconds)	32
Per Port	
Received Packets	64
Transmitted Packets	64
Received Bytes	64
Transmitted Bytes	64
Receive Drops	64
Transmit Drops	64
Receive Errors	64
Transmit Errors	64
Receive Frame Alignment Errors	64
Receive Overrun Errors	64
Receive CRC Errors	64
Collisions	64
Per Queue	
Transmit Packets	64
Transmit Bytes	64
Transmit Overrun Errors	64

Imagem Teleco Fonte: Foundation (2009) Open Network Foundation. Openflow switch specification, version 1.0.0.

OpenFlow

Canal Seguro (até o controlador);

O protocolo de comunicação entre o switch e o controlador utiliza do protocolo TCP para transporte do protocolo OpenFlow.

Se configura no switch quem será o controlador e que porta ele deverá trabalhar.

uso de SSL/TLS é recomendado com o objetivo de garantir confidencialidade em toda comunicação.

É importante lembrar que o OpenFlow não requer uma conexão física direta entre o switch e o controlador,podendo utilizar redes já em produção para efetuar a comunicação. Por isso, a importância de um canal seguro.

OpenFlow

O protocolo OpenFlow

suporta três tipos diferentes de mensagens:

Controlador-Switch - Geradas pelo controlador para gerenciar e inspecionar o estado de um switch.

Assíncronas - Geradas pelo switch para atualizar o controlador sobre eventos da rede e mudanças no estado do switch.

Simétricas - Podem ser geradas tanto pelo controlador quanto pelo switch. São enviadas sem solicitação.

OpenFlow

Controlador-Switch

Características (Features) - O controlador requisita as características do switch. O switch deve responder com as características suportadas;

Configuração (Configuration) - Usado para configurar ou solicitar configurações do switch;

Modificação de estado (Modify-State) - Usado para adicionar, deletar e modificar a tabela de fluxos e para setar propriedades nas portas do switch;

Leitura de estado (Read-State) - Coleta estatísticas;

Envio de pacote (Send-Packet) - Utilizado para enviar pacotes por uma determinada porta do switch;

Barreira (Barrier) - Usado para garantir que as dependências foram atendidas ou para receber notificações de operações finalizadas;

OpenFlow

Assíncrona

Entrada de pacotes (Packet-In) - Utilizado quando fluxos não classificados entram no switch.

Remoção de fluxo (Flow-Removed) - Mensagem enviada para o controlador, quando um fluxo é removido da tabela. Seja por Idle Timeout, Hard Timeout ou por uma mensagem de modificação da tabela de fluxos que delete a entrada em questão;

Estado da porta (Port-Status) - Mensagem enviada para o controlador sempre que há mudanças nas configurações das portas;

Erro (Error) - Notificações de erros;

OpenFlow

Simétrica

Hello - Mensagens trocadas entre o controlador e o switch quando uma conexão é estabelecida;

Echo - Mensagens usadas para identificação de latência, largura de banda e existência de conectividade;

Vendor - Provêem uma forma padrão para os switches OpenFlow oferecerem funcionalidades adicionais;

OpenFlow

Versões do Protocolo.

Diferentes versões do protocolo OpenFlow estão disponíveis.
A primeira versão foi a 0.2.0 lançada em Maio de 2008 e atualmente está obsoleta.

A versão 1.0.0 lançada em Dezembro de 2009 foi a mais amplamente divulgada e implementada.

Após a versão 1.0.0, foram lançadas as versões 1.1, 1.2, 1.3, 1.4 e 1.5.

O OpenFlow encontra-se em constante evolução e o mesmo é mantido pela ONF (Open Network Foundation).

OpenFlow

Evolução do Protocolo

OpenFlow 1.0 (protocolo mais conhecido)
OpenFlow 1.1
Multiplas tabelas e grupos de tabelas.
Algumas combinações tipos fluxos e ações.
OpenFlow 1.2
Suporte a IPv6.
Possibilidade de conectar um comutador a múltiplos controladores.
• Recuperação rápida de falhas.
• Balanceamento de carga.

OpenFlow

OpenFlow 1.3

- Possibilidade de controlador a taxa de pacotes por fluxo.
- Métricas.
- Tabela de características.

OpenFlow 1.4

Sincronização da tabela de fluxo.

Monitoramento de fluxo.

OpenFlow 1.5

Combinações e ações mais refinadas.

Tabelas de saída.

Melhorias da tabela de grupo / métrica.

Exemplo Controladores

- NOX

Primeiro controlador desenvolvido

C++

- POX

Baseado no NOX

Python

Permite desenvolvimento mais rápido

- OpenDayLight

Java

Roda diretamente da JVM

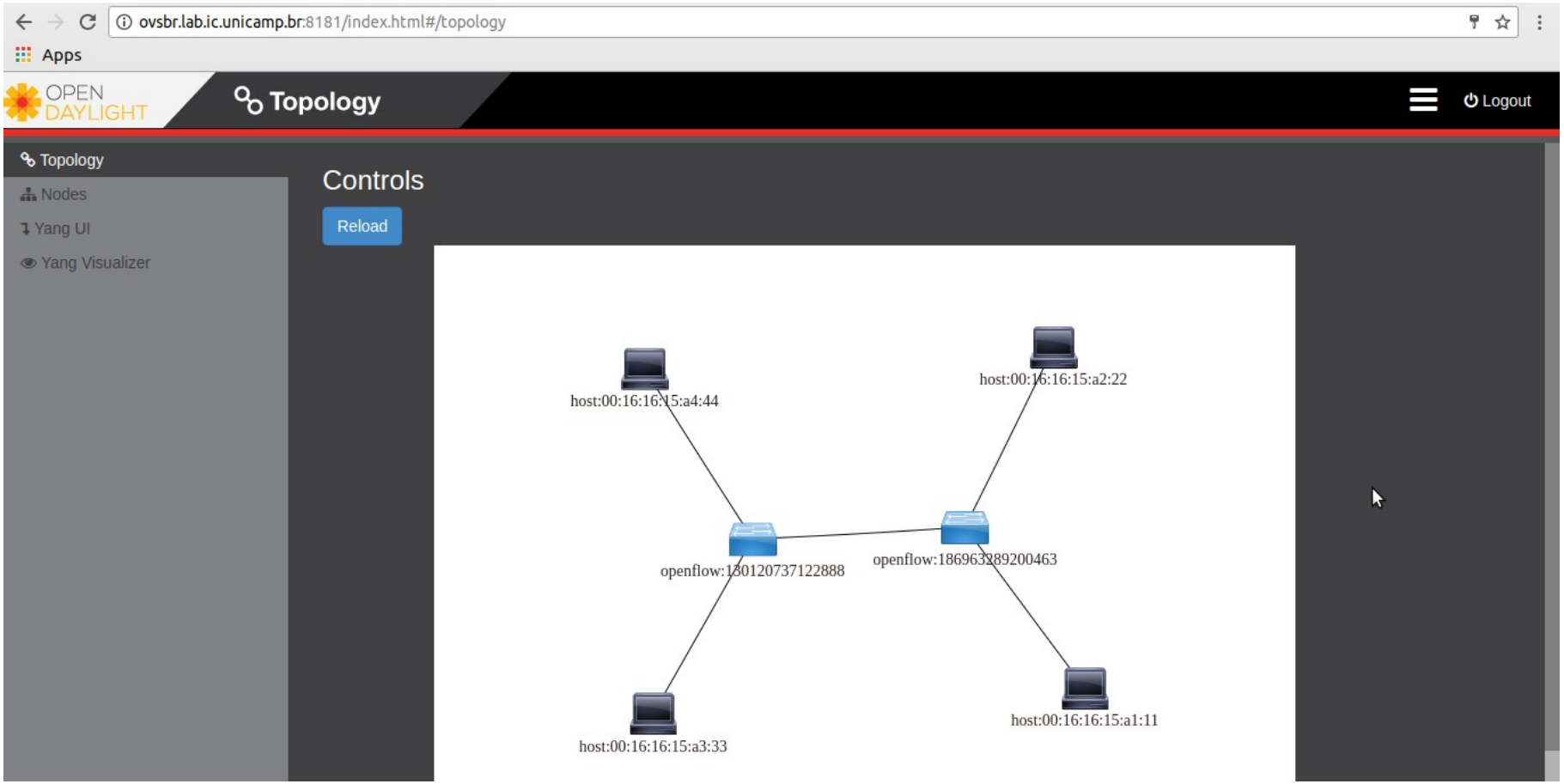
- ONOS

Java

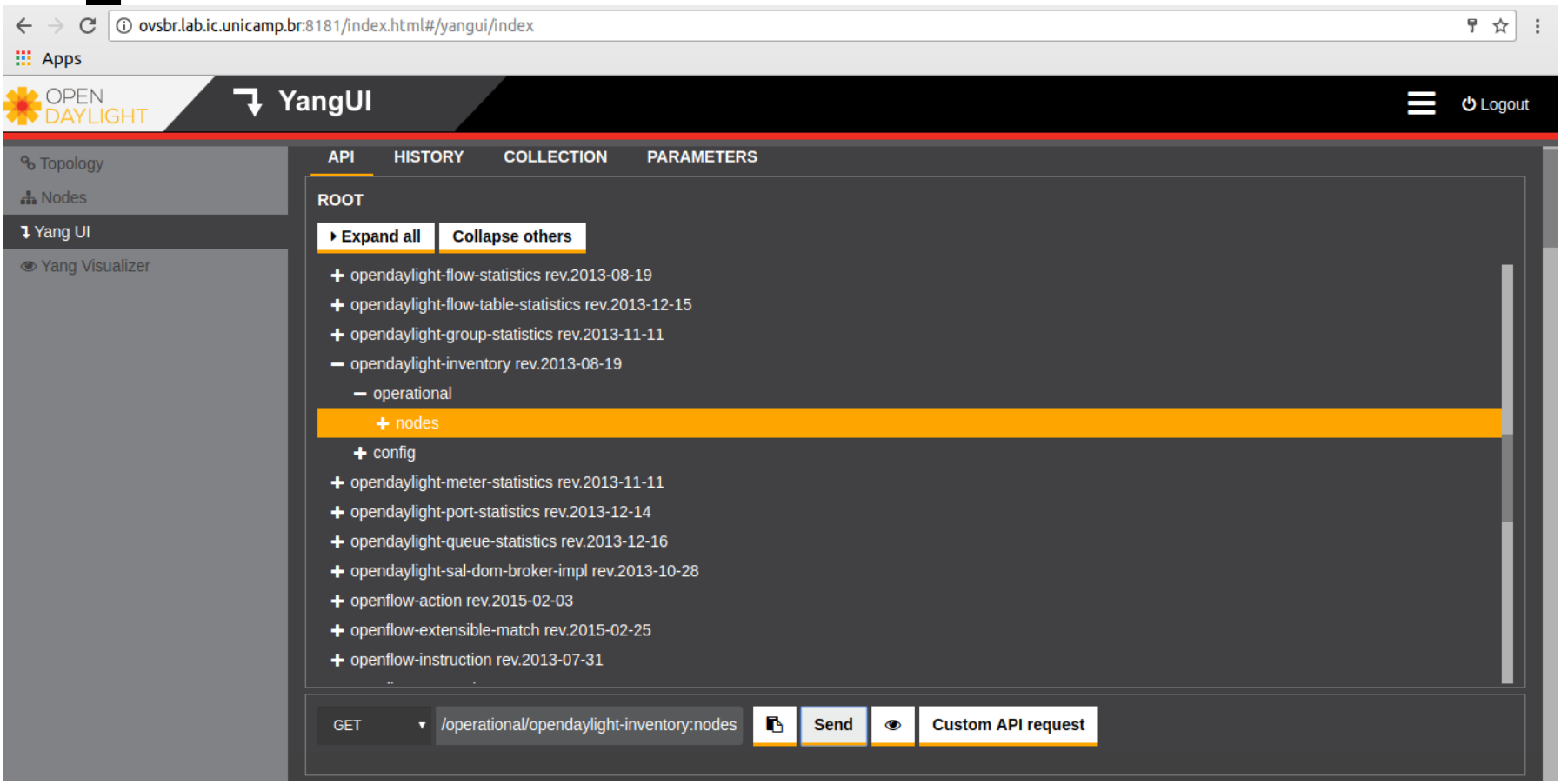
Suporta falhas sem causar interrupções

Próprio conjunto de abstrações

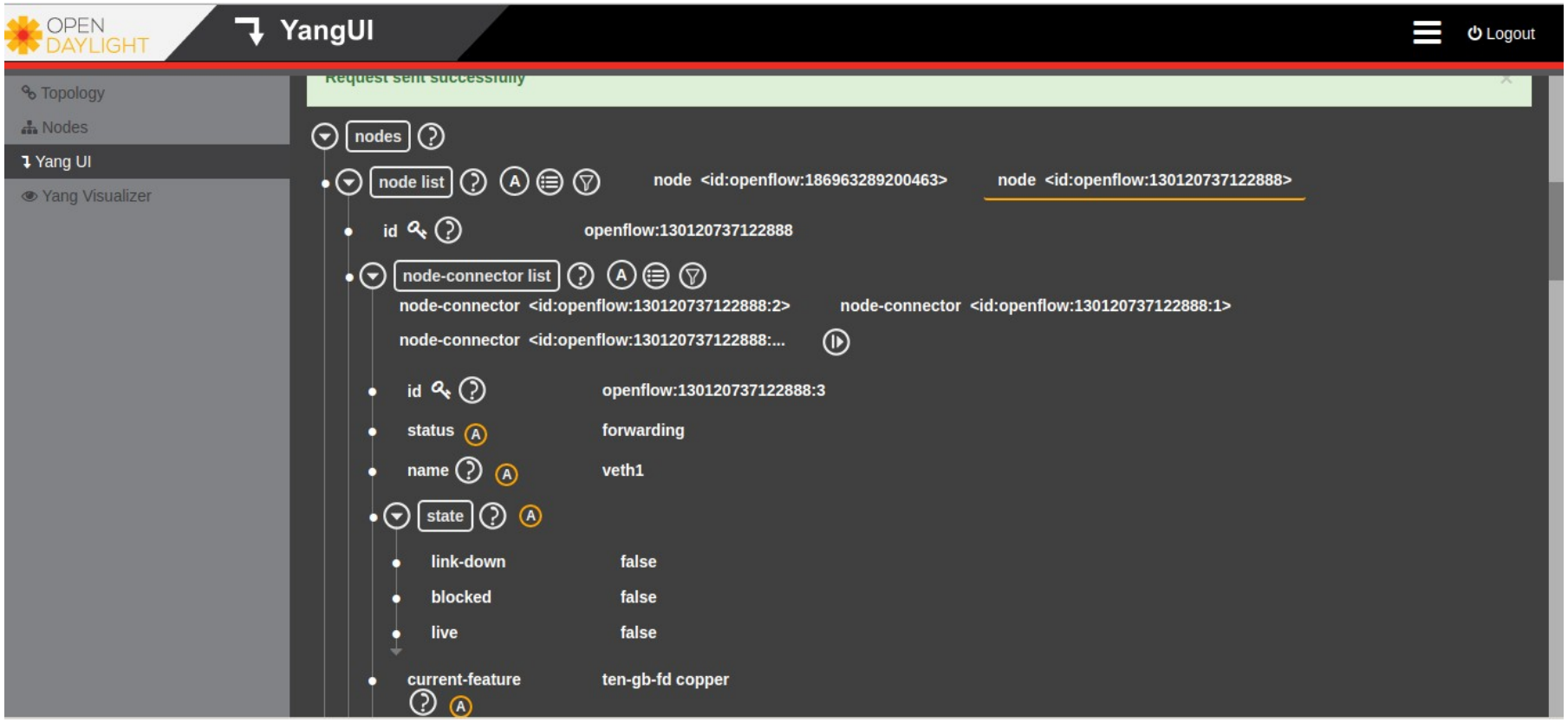
OpenDayLight



OpenDayLight



OpenDayLight



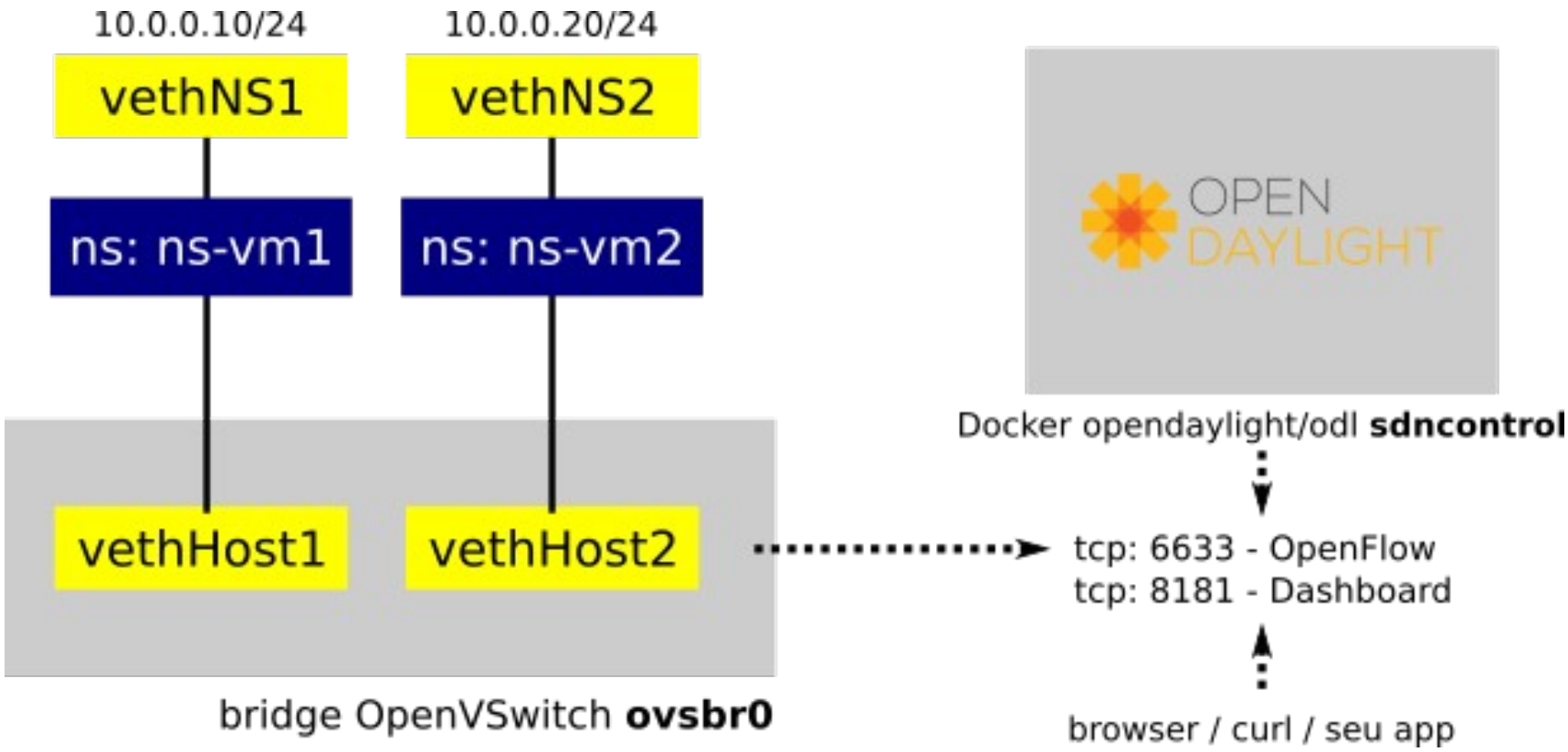
API OpenDayLight

```
$ curl --user "admin":"senha123"
http://ovsbr.lab.ic.unicamp.br:8181/restconf/operational/o
pendaylight-inventory:nodes/node/openflow:186963289200463/
flow-node-inventory:switch-features

{"flow-node-inventory:switch-features":
{"max_tables":254,"capabilities":["flow-node-
inventory:flow-feature-capability-group-stats","flow-node-
inventory:flow-feature-capability-flow-stats","flow-node-
inventory:flow-feature-capability-queue-stats","flow-node-
inventory:flow-feature-capability-port-stats","flow-node-
inventory:flow-feature-capability-table-
stats"],"max_buffers":0}}
```

** GET, POST, PUT, PATCH, DELETE

Atividade


10.0.0.10/24
10.0.0.20/24
vethNS1
vethNS2
ns: ns-vm1
ns: ns-vm2
vethHost1
vethHost2
bridge OpenVSwitch ovsbr0
OPEN
DAYLIGHT
Docker opendaylight/odl sdncontrol
tcp: 6633 - OpenFlow
tcp: 8181 - Dashboard
browser / curl / seu app