



Assis Tiago de Oliveira Filho

assis.tiago@gprt.ufpe.br
atof@cin.ufpe.br

[Topics](#) ▼[News](#)[In Depth](#)[Reviews](#)[Blogs](#) ▼[Opinion](#)[Sha](#)[Data Center](#)[Disaster Recovery](#)[High Performance Computing](#)[Infrastructure Managem](#)[Sustainable IT](#)[Virtualization](#)[Home](#) > [Data Center](#) > [Virtualization](#)**News**

VMware to acquire OpenFlow pioneer Nicira for \$1.26 billion

Nicira helped define the emerging field of software-defined networking

By Josh Jackson

July 23, 2012 05:26 PM ET 

July 23, 2012

B4: Experience with a Globally-Deployed Software Defined WAN

Sushant Jain, Alok Kumar, Subhasree Mandal, Joon Ong, Leon Poutievski, Arjun Singh, Subbaiah Venkata, Jim Wanderer, Junlan Zhou, Min Zhu, Jonathan Zolla, Urs Hölzle, Stephen Stuart and Amin Vahdat

Google, Inc.
b4-sigcomm@google.com

Sigcomm
2013

Google's OpenFlow WAN



Agosto, 2013



Google Lifts Veil On “Andromeda” Virtual Networking

April 2, 2014 by Timothy Prickett Mo



Topic: Cloud

Follow via: RSS

Google launching Andromeda networking stack for Cloud Platform

Summary: Google promises to make network connections for Cloud Platform



By Rachel King for B

Follow @rachelking

Abril, 2014

The Register
Biting the hand that feeds IT

DATA CENTRE SOFTWARE NETWORKS SECURITY BUSINESS HARDWARE SCIENCE BOOTNOTES

Google exposes its Andromeda software-defined networking

Search giant plugs cloud customers into its thrumming network arteries

3 Apr 2014 at 03:22, Jack Clark



SDN é apresentada como o futuro da colaboração acadêmica internacional



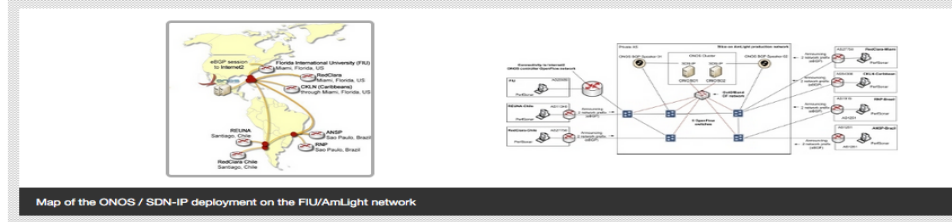
A tecnologia SDN (Redes Definidas por *Software*) foi apontada como o futuro da colaboração acadêmica internacional por Jerônimo Bezerra, da Universidade Internacional da Flórida (FIU, em inglês), no primeiro dia do WRNP 2015, em Vitória. O palestrante atualmente coordena projetos na AmLight, infraestrutura de rede criada para facilitar a troca de tráfego entre

pesquisadores dos Estados Unidos e da América do Sul.

2015

FIU/AmLight Deploys ONOS and SDN-IP Software Live on Next-Generation Research and Education Networks Connecting Brazil, Chile and the Caribbean to the US

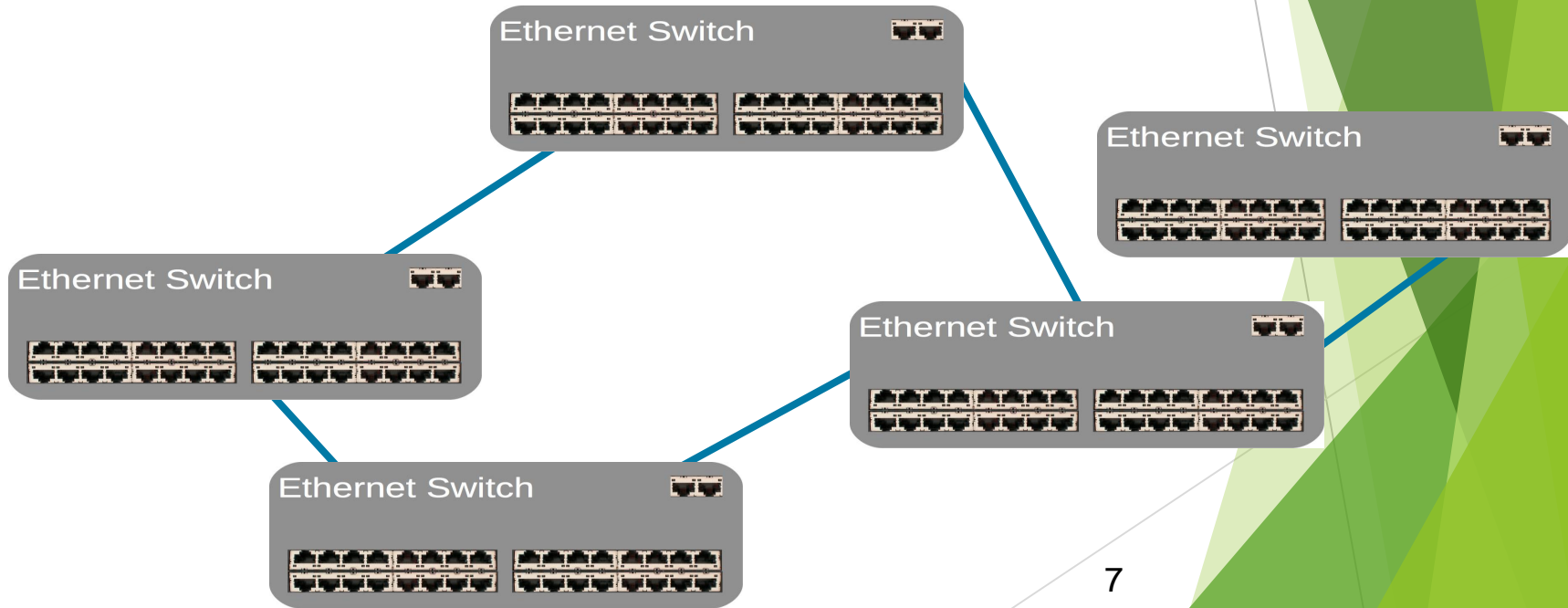
Collaboration among FIU/AmLight, ANSP, RNP, RedClara, REUNA, and KCLN delivers a pure OpenFlow-based Software-Defined Network to accelerate research and education



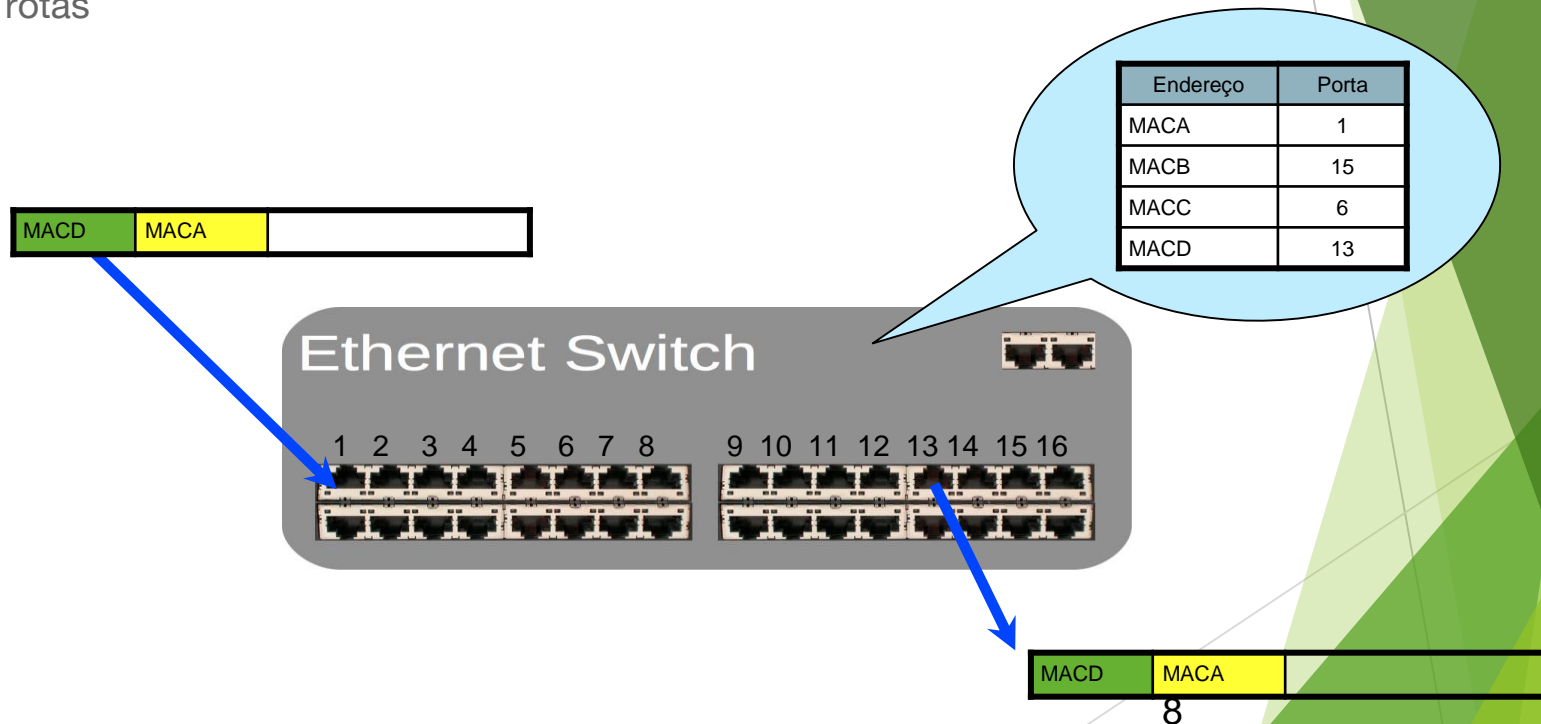
MIAMI, Aug. 20, 2015 /PRNewswire/ -- Florida International University (FIU) and the Open source SDN Network Operating System (ONOS) community today announced they have actively deployed ONOS and its SDN-IP Peering application on Americas Lightpaths (AmLight), creating a software-defined networking (SDN) facility entirely based on OpenFlow. Five Latin American research and

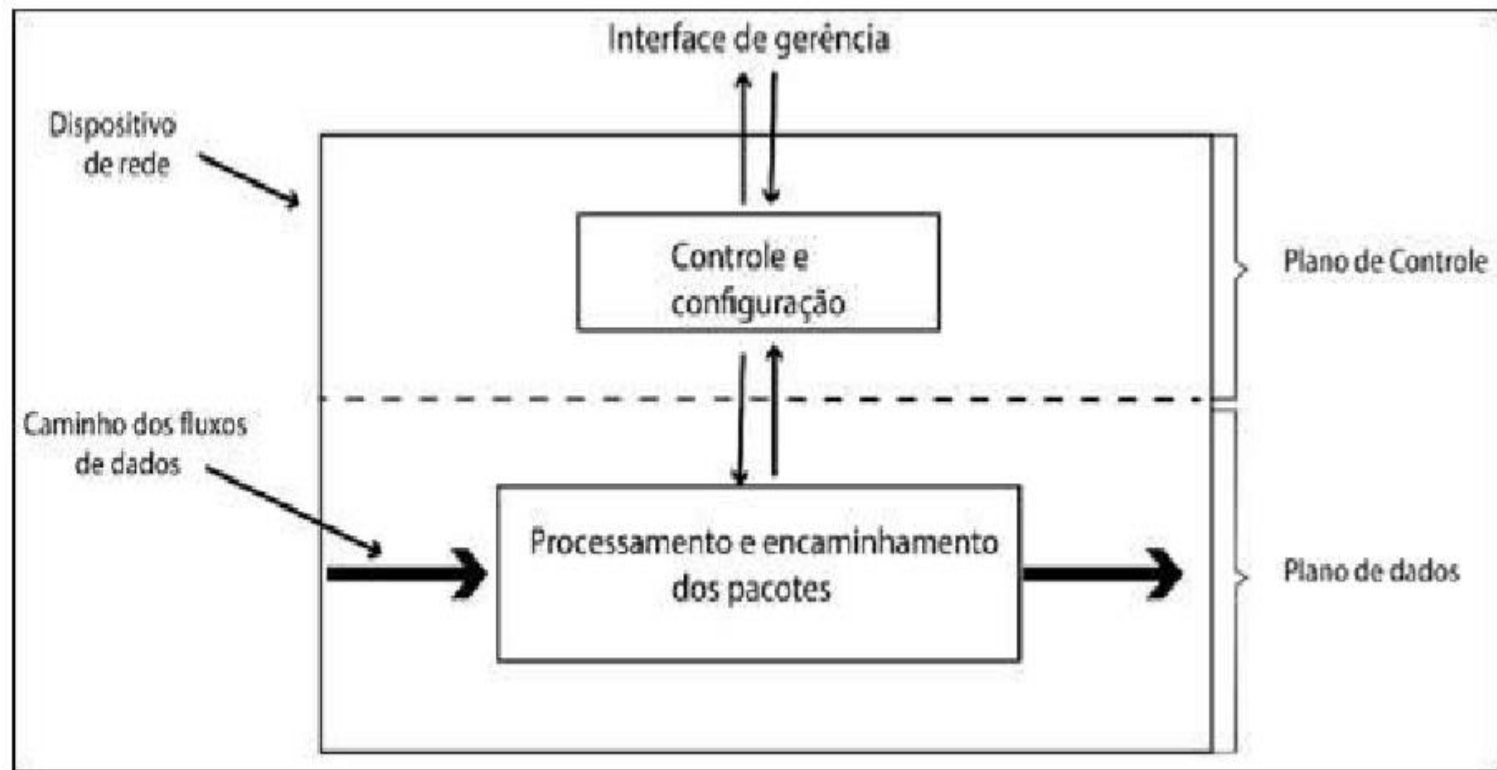


- Introdução
- **Elementos de uma rede tradicional**
- O padrão OpenFlow
- O princípio de Redes Definidas por Software
- Cenários de aplicação
- O controlador SDN
- Exemplos de controladores existentes



Encaminhamento: selecionar um porto de saída baseado no endereço de destino e na tabela de rotas





Plano de dados: o caminho de cada pacote

- pacote chega
- tabela de encaminhamento é consultada – e atualizada
- pacote sai

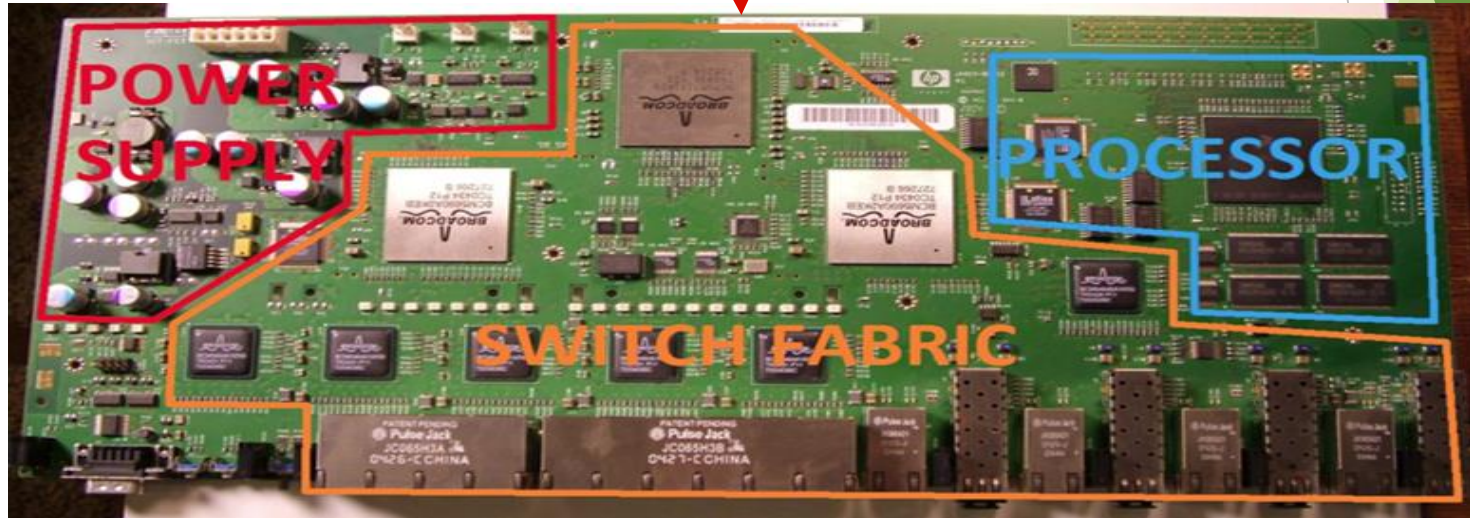
Plano de controle: tudo que acontece “por trás”

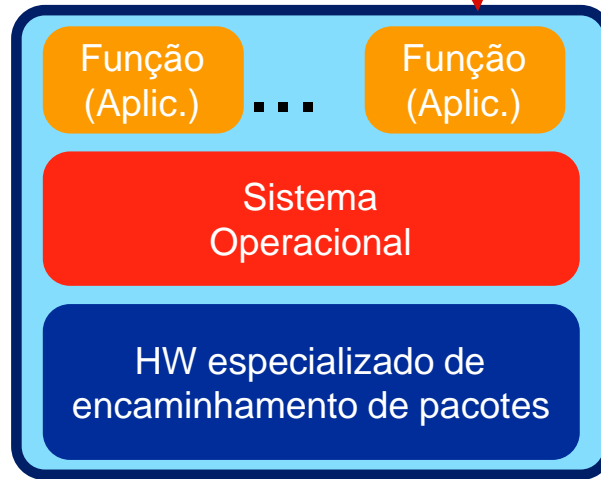
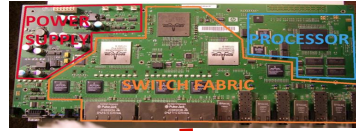
- atualizações da tabela de encaminhamento
- manutenção de estatísticas

Plano de controle sofisticado

Plano de dados otimizado







Muitas funções complexas embutidas na infraestrutura

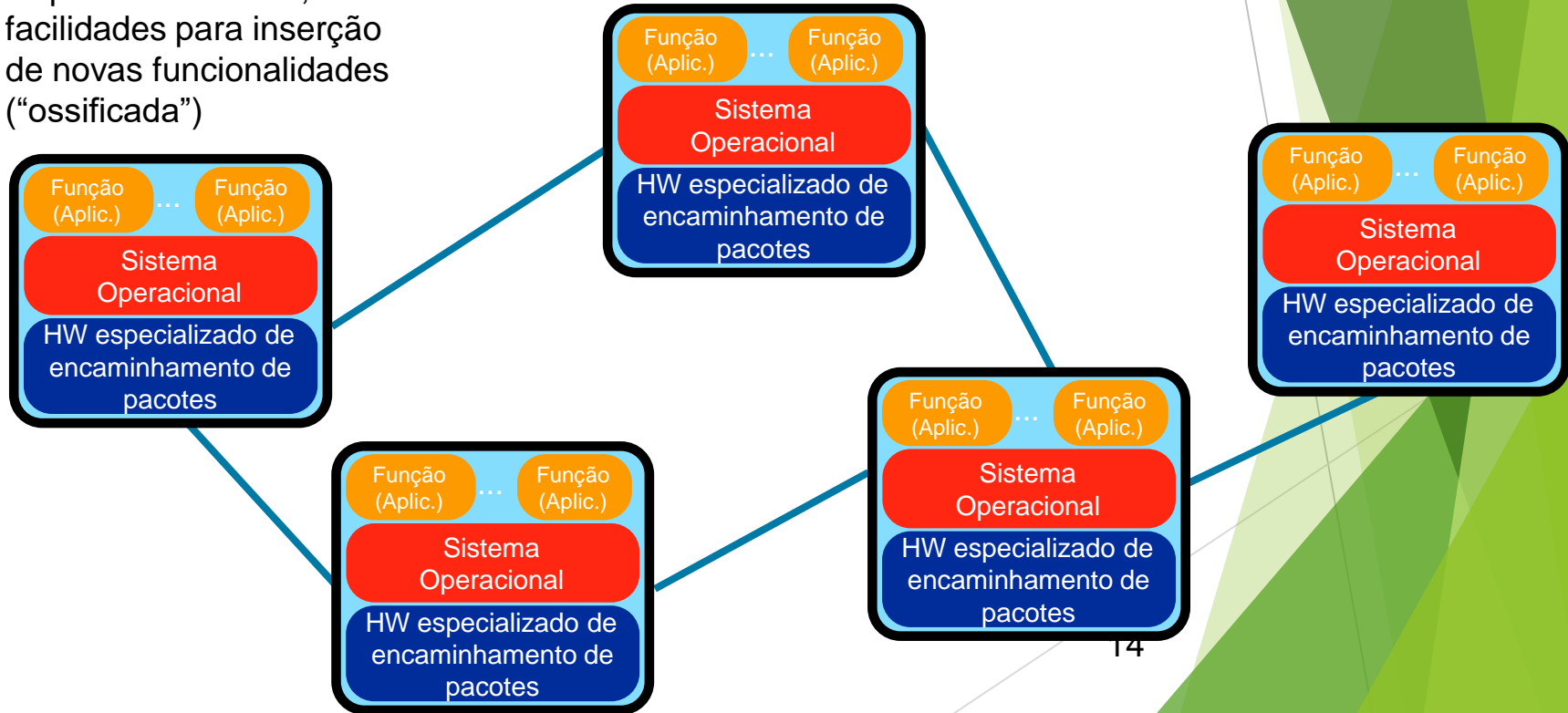


Gerenciamento, roteamento, firewall, controle de acesso, VPN, etc.

Milhões de linhas de código

Bilhões de portas lógicas

Arquitetura fechada, sem facilidades para inserção de novas funcionalidades (“ossificada”)





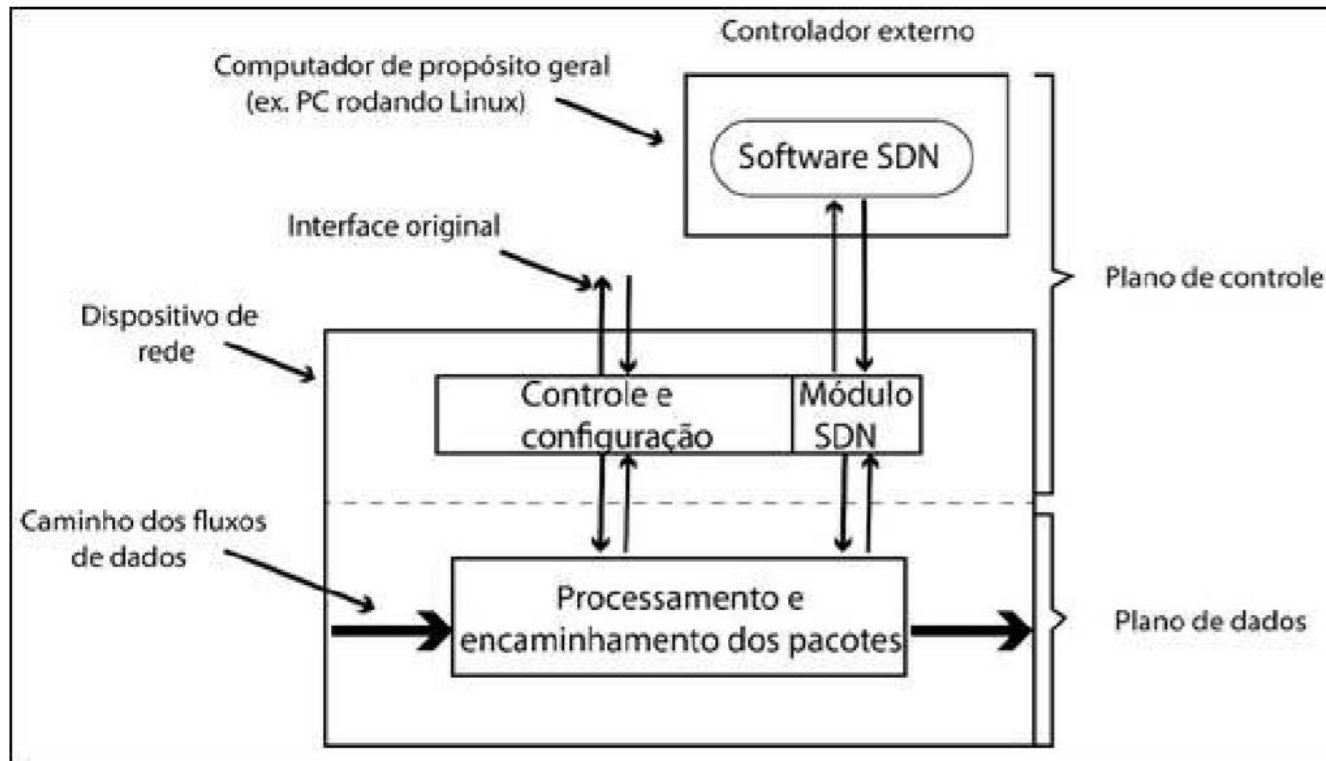
- Introdução
- Elementos de uma rede tradicional
- O padrão OpenFlow
- **O princípio de Redes Definidas por Software**
- Cenários de aplicação
- O controlador SDN
- Exemplos de controladores existentes

Uma nova forma de olhar para o problema:

- Dotar novamente o núcleo da rede de recursos de programação

Origens:

- Sane (2006), Ethane (2007): controle centralizado de regras de firewall distribuídas em uma rede corporativa
- OpenFlow (2008): extensão do princípio dos trabalhos anteriores como uma forma de se programar os elementos de comutação da rede

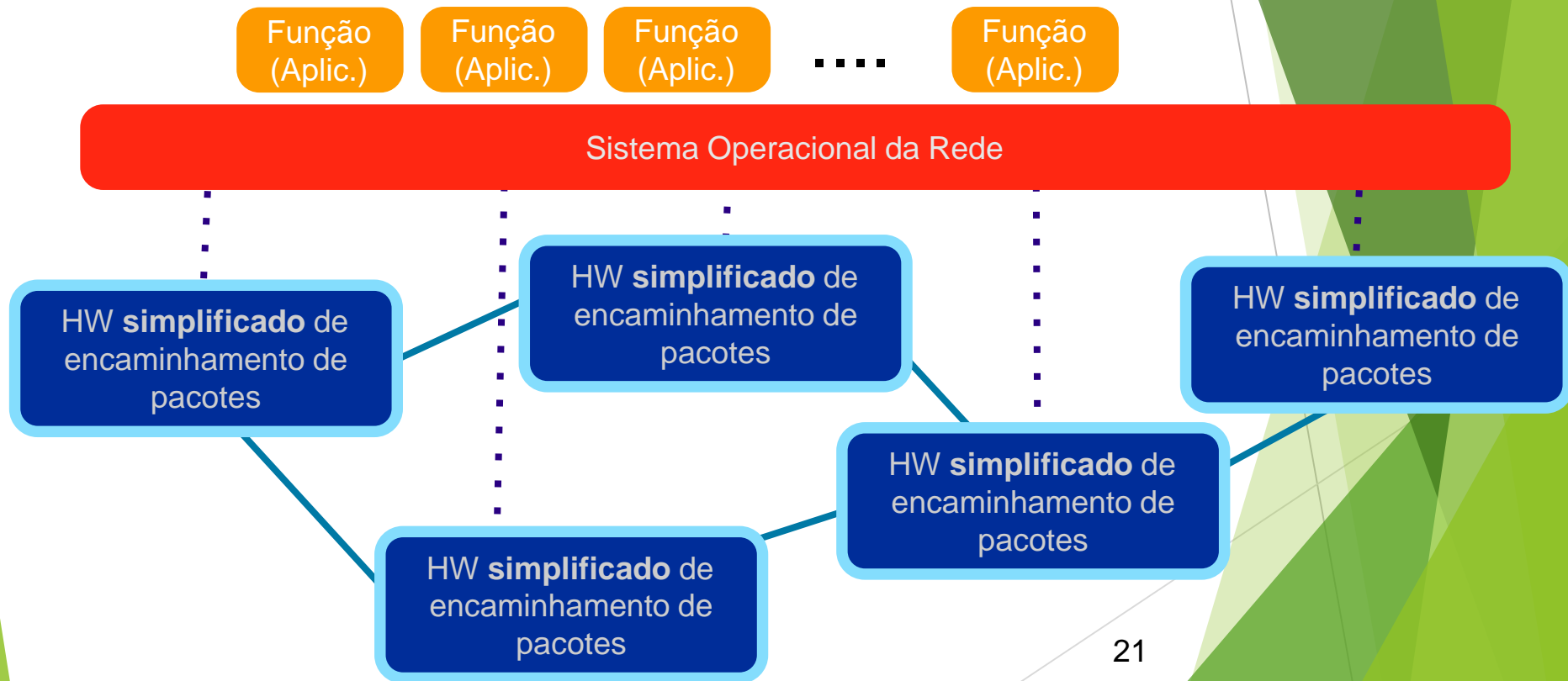


A proposta de Redes Definidas por Software

- ↯ Retirar do comutador (switch/roteador) a lógica do plano de controle
- ↯ Transferir essa funcionalidade para um controlador da rede
- ↯ Controlador pode executar programas que comandam funções
 - O controlador atua como um Sistema Operacional da rede
 - Os programas são as aplicações que executam na rede
- ↯ O controlador programa as tabelas de encaminhamento
- ↯ Comutadores consultam o controlador para casos indefinidos

Essa “lobotomia” da inteligência do equipamento da rede para controladores logicamente centralizados possibilita a definição do comportamento da rede em software não apenas pelos fabricantes do equipamento, mas também por fornecedores ou pelos próprios usuários, como, por exemplo, operadores de rede.

Rothenberg



Função
(Aplic.)

Função
(Aplic.)

Função
(Aplic.)

...

Função
(Aplic.)

Sistema Operacional da Rede (Controlador SDN)

Protocolo OpenFlow

OpenFlow Switch

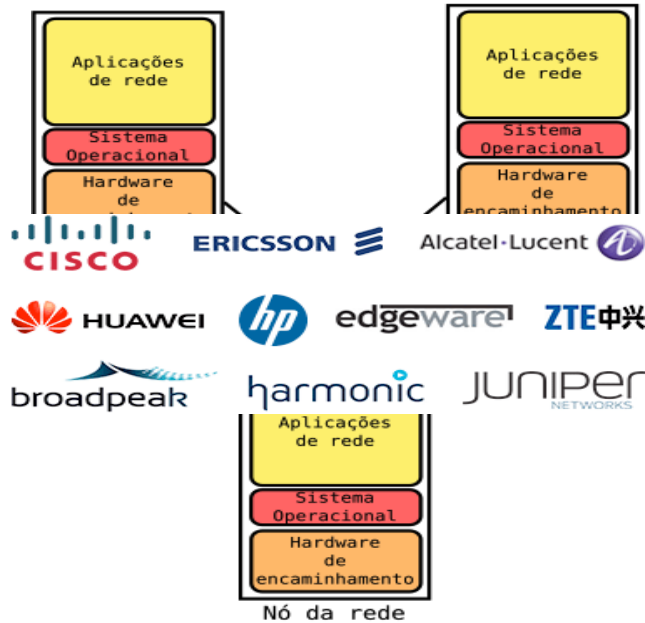
OpenFlow Switch

OpenFlow Switch

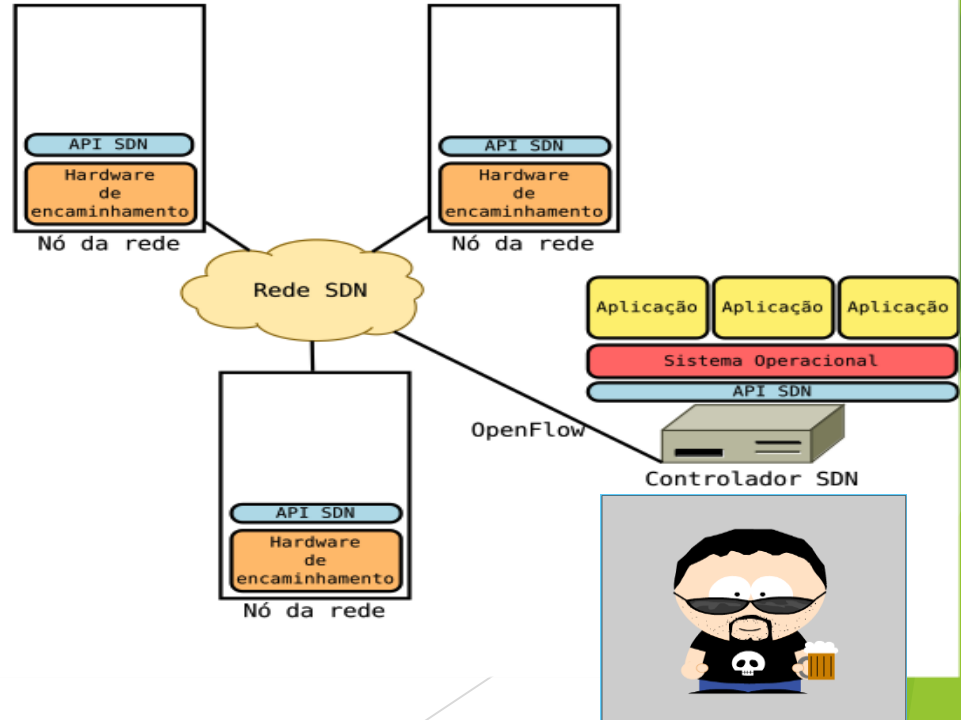
OpenFlow Switch

OpenFlow Switch

Rede tradicional



Rede definida por software



- Introdução
- Elementos de uma rede tradicional
- **O padrão OpenFlow**
- O princípio de Redes Definidas por Software
- Cenários de aplicação
- O controlador SDN
- Exemplos de controladores existentes

- ▶ O padrão OpenFlow descreve um protocolo aberto que permite que aplicações em software possam coordenar as tabelas de fluxo de diferentes dispositivos .
- ▶ O protocolo OpenFlow é uma proposta para padronizar a comunicação entre um elemento externo baseado em software, denominado controlador, e os dispositivos de redes (switch OpenFlow) na arquitetura SDN

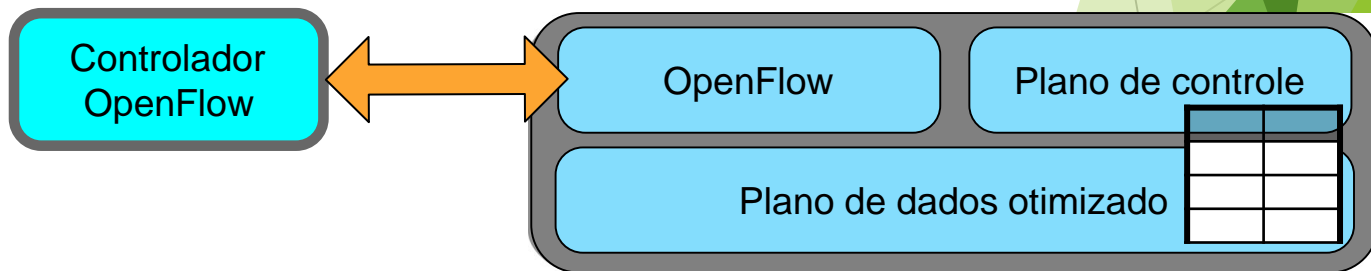
A ideia básica é simples: exploramos o fato de que a maioria dos switches e roteadores ethernet modernos contém tabelas de fluxo que funcionam a uma taxa de linha para implementação de Firewalls, protocolo NAT, QoS (Quality of Service), além de coletar estatísticas. Enquanto cada fluxograma de cada fornecedor é diferente, identificamos um conjunto comum de funções que são utilizadas em muitos switches e roteadores. OpenFlow explora este conjunto comum de funções.

Organização de um comutador (switch) OpenFlow

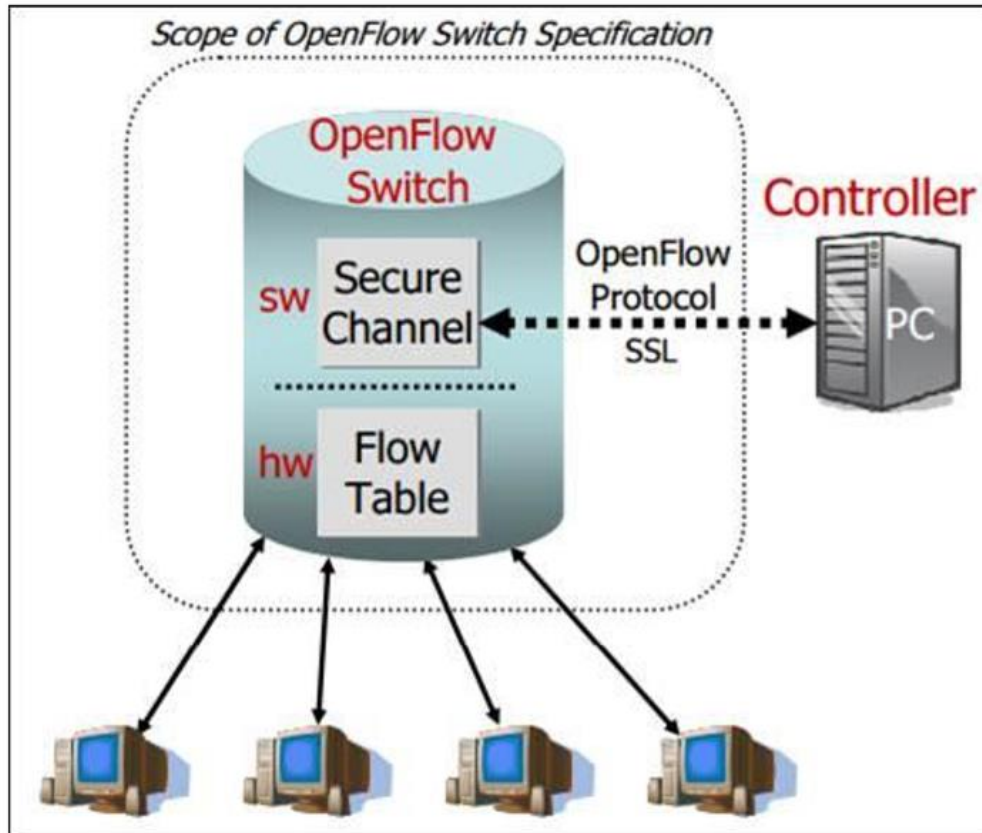
Plano de dados: o caminho de cada pacote

- pacote chega
- tabela de encaminhamento é consultada – e atualizada
- tabela determina como o pacote deve ser tratado

Plano de controle: só dá acesso à tabela de encaminhamento



Arquitetura OpenFlow



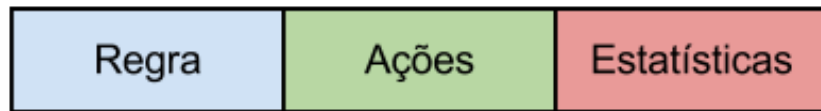
OpenFlow: entradas da tabela de encaminhamento (ou fluxo)

Entradas na tabela de encaminhamento possuem **três** partes:

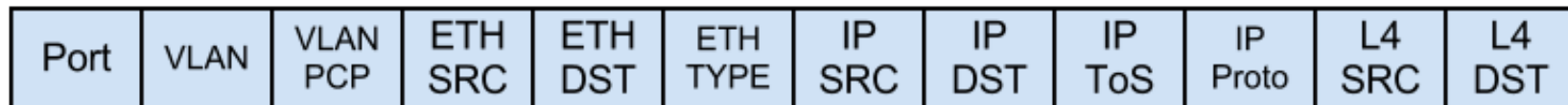
1. **Regras:** padrões para identificar fluxos
2. **Ações:** o que fazer para cada pacote do fluxo
3. **Estatísticas:** contadores de bytes e pacotes do fluxo

Caso não haja uma regra definida para um tipo de pacote (um fluxo), envia o pacote para o controlador

Campos de Cabeçalho	Contadores	Ações
----------------------------	-------------------	--------------

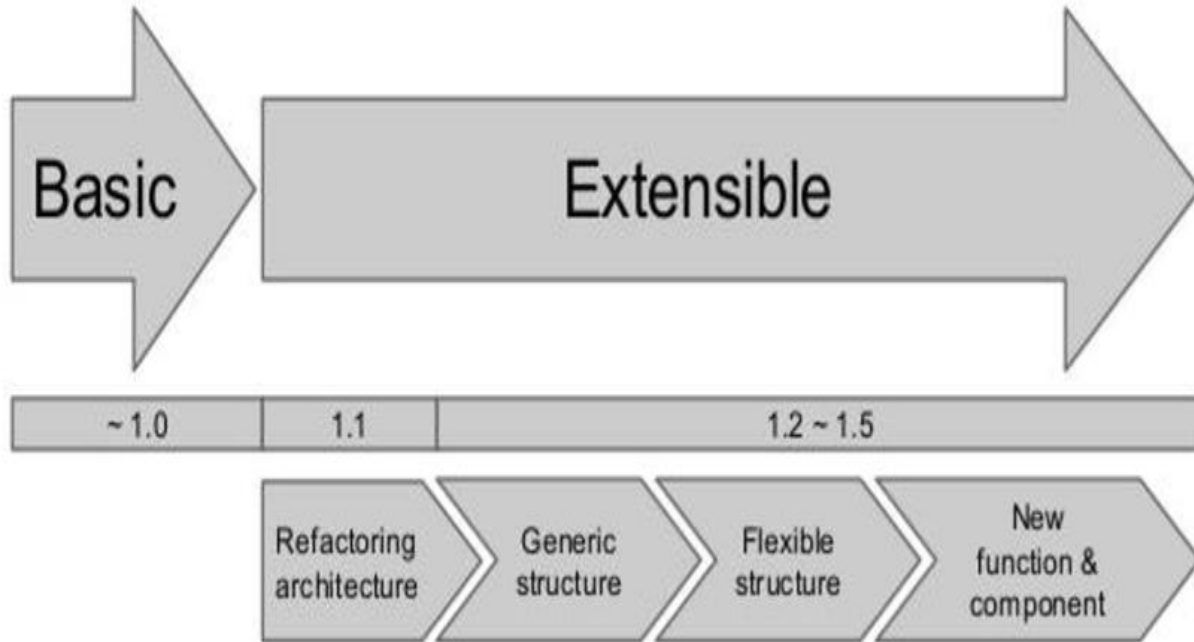


- Encaminhar pacote
- Enviar pacote para controlador
- Alterar campos
- ...



(OpenFlow 1.1)

PORT	VLAN	MAC Src	MAC Dst	Eth Type	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Ação
*	*	*	00:1f...	*	*	*	*	*	*	Porta 6
p3	vl1	00:20...	00:1f...	0800	1.2.3.4	5.6.7.8	4	1765	80	Porta 6
*	*	*	*	*	*	*	*	*	22	Drop
*	*	*	*	*	*	5.6.7.8	*	*	*	Porta 6
*	vl1	00:1f...	*	*	*	*	*	*		Portas 6,7,8



OpenFlow: versões

Existem diversas versões do padrão OpenFlow em uso/desenvolvimento:

- A versão mais estável ainda é a 1.0
- Muitos controladores implementam 1.1+
- A partir da versão 1.2 do OpenFlow é possível definir padrões arbitrários
- A versão 1.2 é em geral evitada
- IPv6 e tunelamento são suportados a partir da versão 1.3
- Alguns switches afirmam implementar a versão 1.4
- A versão 1.5 é a mais recente



Transmode PON TM-Series



Pica8 Open Switches



CISCO Nexus 3000/9000



Brocade ADX/MLX series



HP 2920



Intune Optical Packet Switch



Infinera MultiGb OTN



HP 3500



Juniper EX Series



ARISTA
Introducing the
Arista Spline™ Network



Extreme Networks BlackDiamond



Huawei Agile Switches

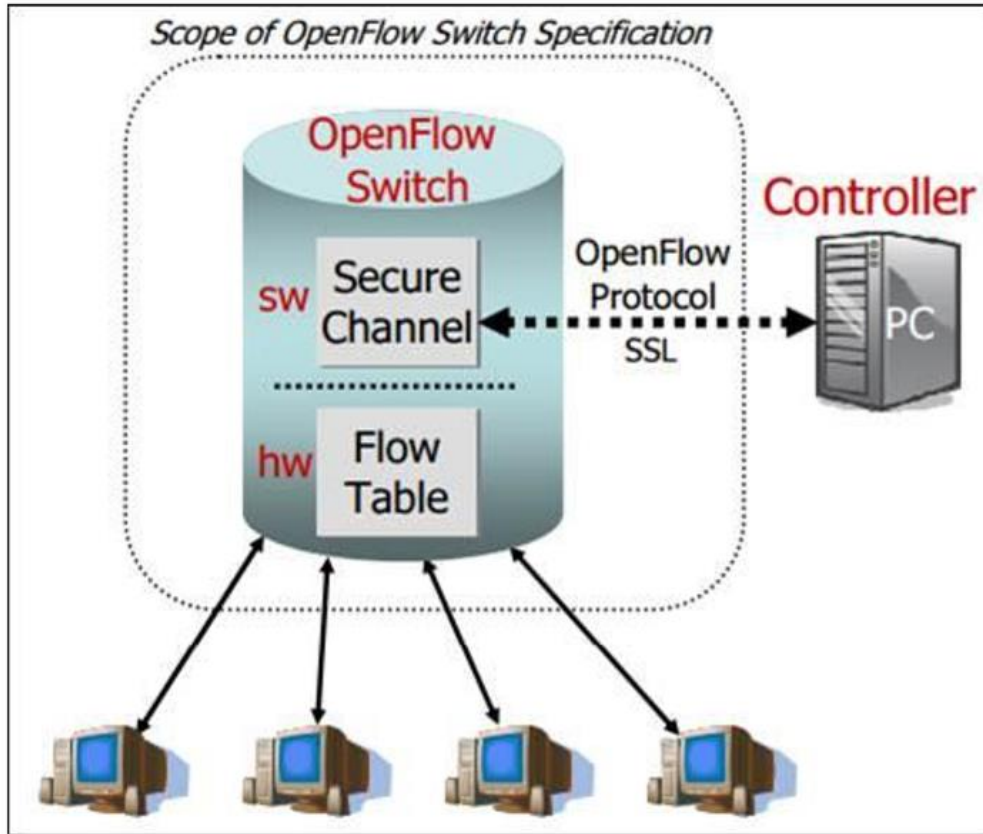


NEC ProgrammableFlow®



HP 5200





- Introdução
- Elementos de uma rede tradicional
- O padrão OpenFlow
- O princípio de Redes Definidas por Software
- **Cenários de aplicação**
- O controlador SDN
- Exemplos de controladores existentes

SDN é legal, mas... o que eu faço com isso?

- Controle de acesso, segurança
- Gerência de redes
- Redes domiciliares
- Gerência de energia
- Comutador virtual distribuído
- Roteador expansível (*scale-out*)
- Datacenters multi-usuários
- Serviços customizados
- Outras aplicações aplicadas a POP's

Controle de acesso

- Contexto que originou o OpenFlow (Ethane)
- Implementação de políticas fica mais simples com a visão global
 - Visão de alto nível permite desassociar pessoas de identificadores de hardware (p.ex., MAC address)
 - Políticas podem se adaptar a cenários complexos
- Substituição de *middle-boxes*

- A visão global criada pelo controlador oferece uma forma simples de agregar informações sobre todos os eventos relevantes para a administração da rede
- A interface de programação dos *switches* oferece uma forma simples para atuar sobre o sistema
 - A escolha do nome “NIB” não parece aleatória...
 - Há uma relação forte com a área de gerência de redes (MIB)

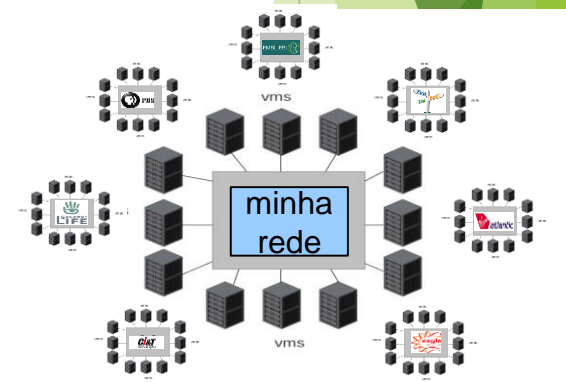
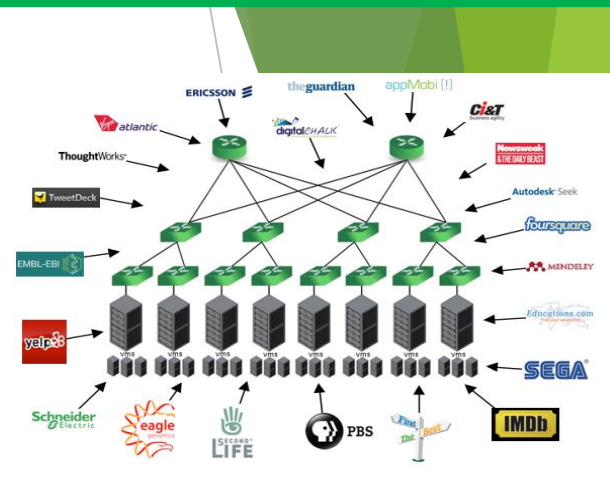
Redes Domiciliares

- Um problema para o usuário doméstico: o que está acontecendo na minha rede?
- Um controlador SDN pode ver cada fluxo que surge na rede
- A visão de todos os fluxos cria uma oportunidade para entender os padrões de acesso dos moradores e detectar desvios
- Padrões de tráfego doméstico podem ser reveladores para detecção de malware
- O controlador da rede pode exportar informações para um provedor de serviços de gerência e configuração (proj. BISmark, Gatech)

- Técnicas de posicionamento e distribuição de carga se beneficiariam de um maior conhecimento do que passa pela rede
 - Desvio parcial do fluxo de uma máquina
 - Evitar sobrecarga
 - Desvio total do fluxo de uma máquina ou switch
 - Permite o desligamento (manutenção/economia)
- Migração de VMs pode ser usada para distribuir carga
 - SDN pode simplificar o processo de migração

Comutador (*switch*) virtual distribuído

- Em datacenters, o ambiente virtualizado inclui o uso de *switches* virtuais para interligar as diversas VMs
- VMs de usuários diferentes (tenants) compartilham máquinas e a rede
- Com SDN, a rede pode ser programada para exportar a visão de que VMs de um mesmo usuário operem em uma rede isolada
- A visão global da rede podem ser usada para criar a abstração de um *switch* único interligando as VMs de um usuário



Aplicações: Roteador expansível (*scale-out*)

- Roteadores de borda de alta capacidade têm custo bastante elevado
- Um conjunto de *switches* controlados por uma SDN podem exportar a visão de um roteador único
- Um conjunto de rotas podem ser transformado em regras de encaminhamento entre switches que sejam agrupados para substituir o roteador

Desafios – problemas interessantes

Existem ainda diversos problemas de pesquisa que estão sendo abordados no contexto de Redes Definidas por Software:

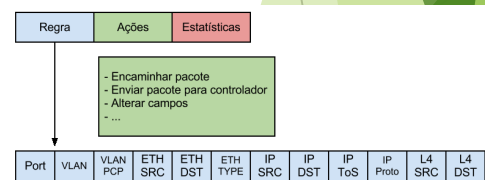
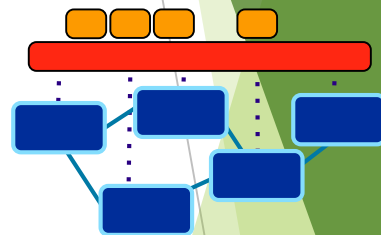
- Visão da rede/abstrações de programação
- Aplicação no contexto de virtualização de serviços e redes
- API para programação dos *switches* – OpenFlow não é obrigatório!
- Especificação de aplicações
- Depuração – como ter certeza que tudo está funcionando?
- Sistema distribuído para desempenho e tolerância a falhas

- Introdução
- Elementos de uma rede tradicional
- O padrão OpenFlow
- O princípio de Redes Definidas por Software
- Cenários de aplicação
- **O controlador SDN**
- Exemplos de controladores existentes

Formas de organização do controlador e da rede

Baseado nos conceitos gerais de OpenFlow e SDN, há diferentes formas de se organizar uma rede:

- Como o controlador é implementado?
- Qual a unidade de identificação do tráfego?
- Quando o controlador toma decisões?



O controlador SDN

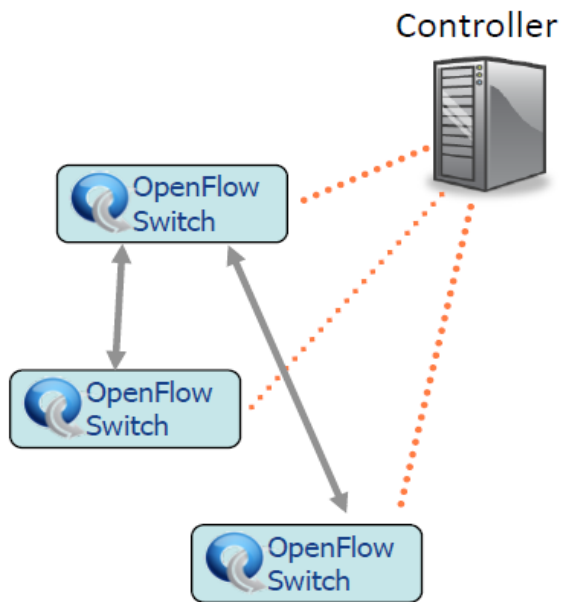
- É responsável por tomar decisões e adicionar ou remover as entradas nas tabelas de fluxo
- Torna-se o ponto estratégico de controle
- Centraliza **logicamente** a inteligência do sistema
- Cria uma visão única e global dos elementos da rede
- Gerencia a rede diretamente

O controlador SDN

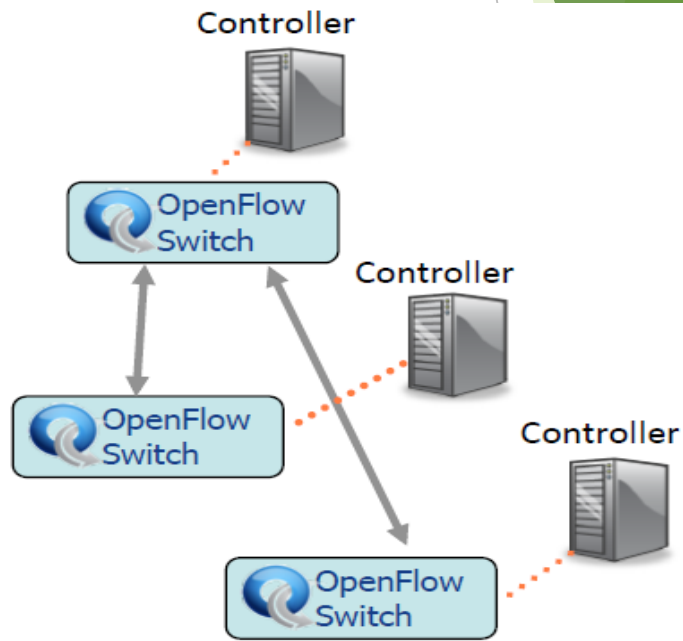
- Abstrai a complexidade da infraestrutura física
- Oferece uma interface de programação para o desenvolvedor
- As regras de processamento de pacote realizam diversas tarefas, como:
 - Roteamento e monitoramento do tráfego
 - Controle de acesso, engenharia de tráfego
- Existem diferentes formas de se organizar esse sistema, como veremos em breve.

Controle Centralizado x Distribuído

Controle centralizado



Controle distribuído



Qual a unidade de identificação do tráfego?

Baseado em Fluxos:

- Todo fluxo é individualmente identificado pelo controlador
- Padrões na tabela são sempre definidos com casamento (“*match*”)
- Tabela de fluxos contém uma entrada (*flow entry*) por fluxo
- Bom para controle de granularidade fina (ex: rede de um campus)

Por Agregação:

- Uma entrada na tabela engloba um grupo de fluxos
- Padrões na tabela definidos usando curingas
- Tabela de Fluxos contém uma entrada por categoria de fluxos
- Bom para grandes número de fluxos (ex: *backbone*)

Quando o controlador toma decisões?

Como mencionado anteriormente, é o controlador que determina o que fazer com cada fluxo/grupo – ele cria as regras da tabela

- ♦ Mas quando essas regras são efetivamente criadas?
 - Quando um fluxo é identificado (controle reativo)
 - Previamente, quando o controlador identifica uma necessidade futura (controle proativo)
 - Em momentos diferentes, dependendo do caso (híbrido)

Reativo vs. Pró-ativo

Reativo

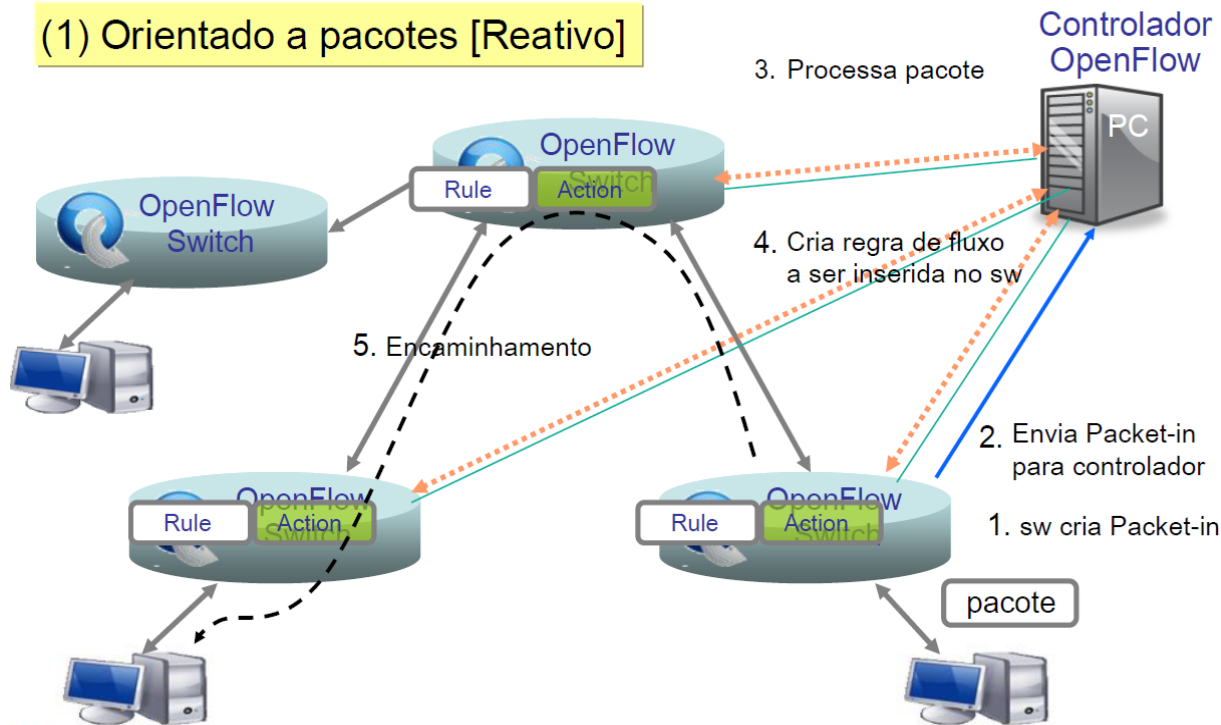
- Primeiro pacote do fluxo aciona o controlador para inserir entradas de fluxo
- Uso eficiente da tabela de fluxo
- Se há perda de conexão com o controlador , funcionalidade fica limitada

Pró-ativo

- Controlador pré-configura tabela de fluxo no switch
- Tempo de configuração novos fluxos é zero adição
- Perda de controle de conexão não interrompe o tráfego

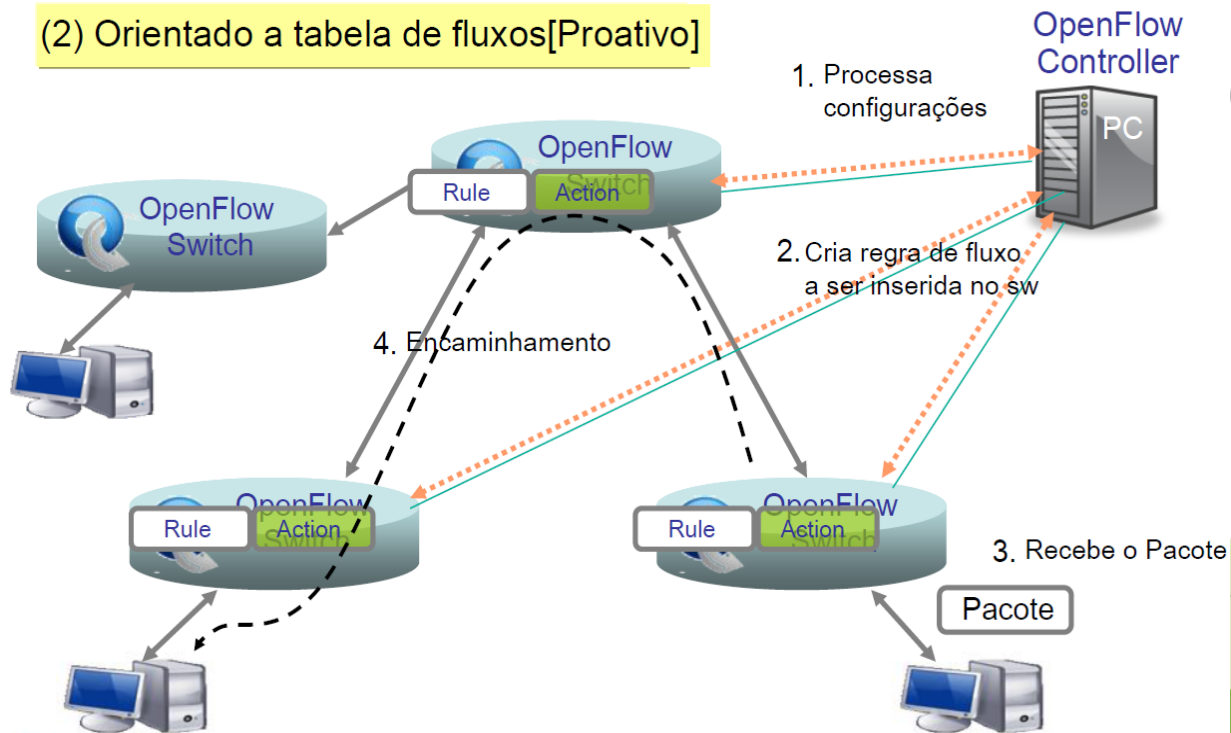
Reativo x Proativo x Híbrido

(1) Orientado a pacotes [Reativo]



Reativo x Proativo x Híbrido

(2) Orientado a tabela de fluxos[Proativo]



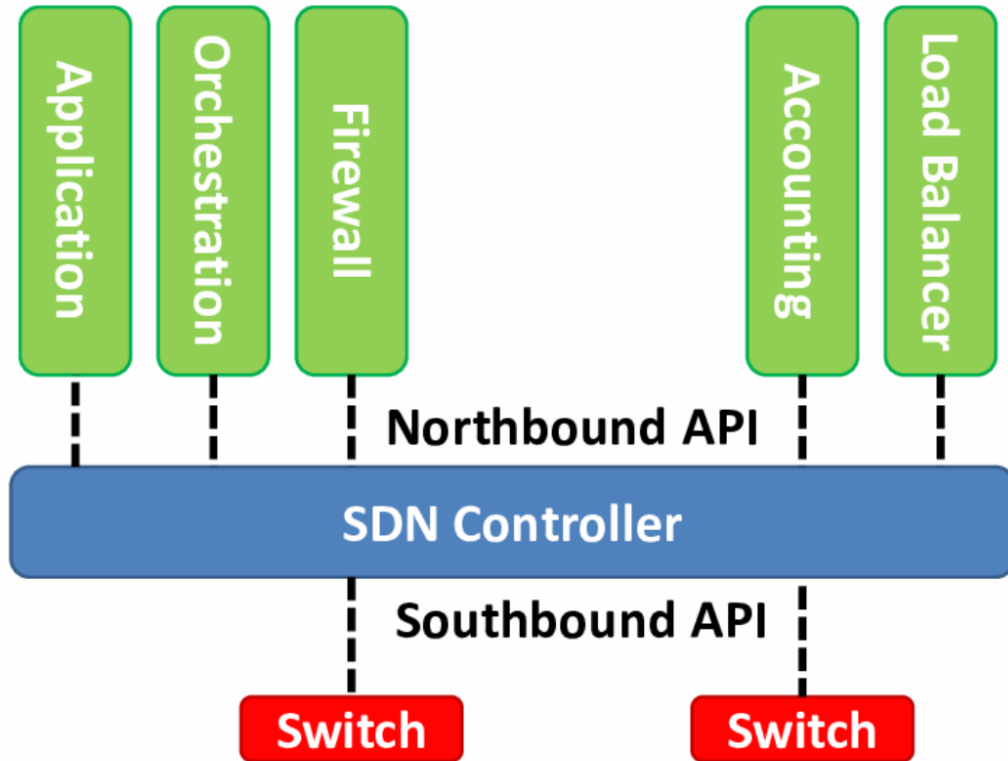
Híbrido

- Combinação dos dois métodos
- Sempre que possível, utiliza o modo proativo, evitando latências maiores na construção de uma entrada na tabela
- Para casos onde a flexibilidade é essencial, como uma decisão dependente do estado da rede, utiliza o modo reativo

Controladores SDN: características a considerar

- Linguagem de programação
- Nível de Abstração
- Curva de aprendizado
- Tamanho da base de usuários
- Suporte (comunidade ou empresa)
- Foco:
 - *Southbound* API ou *Northbound* API?
 - Integração com ambientes de nuvem?
 - Educação, pesquisa ou produção?

Northbound API X Southbound API



Principais controladores SDN

- NOX/POX
- Ryu
- Floodlight
- OpenDaylight
- Beacon
- Onix
- ONOS

Outros controladores:

<https://www.sdxcentral.com/resources/sdn/sdn-controllers/open-source-sdn-controllers/>

- Introdução
- Elementos de uma rede tradicional
- O padrão OpenFlow
- O princípio de Redes Definidas por Software
- Cenários de aplicação
- O controlador SDN
- **Exemplos de controladores existentes**

- Primeiro controlador SDN
- Desenvolvido pela Nicira em paralelo ao padrão OpenFlow (alta sinergia entre conceitos)
- Distribuído como *GPL* em 2008 (muito usado)
 - Diversos controladores mais abstratos construídos sobre ele
- Mantido pela comunidade de pesquisa
- *API* deixa a desejar
- Difícil Instalação e configuração

- Linux
- C++ e Python (“cola” para módulos C++)
- *Cooperative multithreading*
- Sistema de componentes
- Programação orientada a eventos
- Algumas aplicações distribuídas com o Software

POX

- Implementação Python, código aberto, baseada no NOX
- Suporta apenas a versão 1.0 do protocolo OpenFlow
- Amplamente utilizado
- Possui suporte oficial
- Menor tempo de aprendizagem quando comparado com outros controladores
- Baixo desempenho

- Controlador Python de código aberto
- Suporta as versões de OpenFlow 1.0, 1.2, 1.3 e 1.4
- Possui integração com o ambiente de virtualização OpenStack;
- Tempo de aprendizagem moderado
- Baixo desempenho

FloodLight

- Controlador Java/Python de código aberto
- Suporta apenas a versão 1.0 do protocolo Open Flow
- Suporte da Big Switch Networks
- Documentação de boa qualidade
- Alto desempenho
- Permite interação com o controlador a partir de uma API *REST*
- Integração com *OpenStack*
- Maior tempo de aprendizagem

- Controlador Java de código aberto
- Plataforma de programação de redes baseada em diferentes padrões industriais
- Desempenho em níveis adequados para produção
- Integração com o sistema *OpenStack*
- Maior complexidade e tempo de aprendizado
- *Northbound* e *Southbound* API claramente definidas

- Projeto conjunto NEC, Google, Yahoo!
- Código fechado, apenas descrito em um artigo
- Define uma estrutura de dados que representa a visão global da rede através de um grafo (NIB, *Network Information Base*)
- Controle da rede é feito alterando-se a *NIB*
 - Alterações se transformam em comandos para os elementos de comutação
- Ao particionar e replicar a NIB entre máquinas, atinge alta escalabilidade e confiabilidade
- Consistência entre máquinas é garantida pela implementação

- Controlador de código aberto em java
- Distribui o estado da rede em múltiplos controladores distribuídos
- Baseado nos conceitos descritos no artigo do ONIX
- Foco em atender a grandes operadoras e provedores de serviços, voltado para redes WAN
 - Abstrações bem definidas de *NorthBound* e *Southbound*
 - Software com ênfase em modularidade
 - Procura atender a requisitos de escalabilidade, desempenho e alta disponibilidade

Controladores			
NOX	1º Controlador construído; Muito Popular Difícil instalação e configuração API deixa a desejar	Mantido pela comunidade de pesquisa Stanford	Python e C++
POX	Simples de implementar, flexível Derivado do NOX	Ótimo para pesquisa, desempenho deficiente Prototipação rápida	Python
FloodLight	Implementado para redes corporativas Permite integrar com redes não OpenFlow	Deixa a desejar nos critérios de escalabilidade e alto desempenho	Java
Beacon	Multithread. Controlador pode ser modificado dinamicamente	Datacenter com 100 vSwitch + 20 switches	Java
Ryu	Curva de aprendizado moderada, suporte a várias versões do OF, integração com o OpenStack	Documentação extensa, vários cases desenvolvidos, desempenho a desejar	Python
OpenDayLight	Northbound e Southbound bem definidas Controlador de nível 'empresarial'	Curva de aprendizagem elevada Documentação escassa Rumo tomado pelo projeto não agradou a comunidade	Java
ONIX	Estado da rede em múltiplos controladores distribuídos; <i>Network Information Base (NIB)</i>	Projeto em conjunto com NEC, Google, Yahoo! Código fechado Foco: Escalabilidade e confiabilidade	C++, Python e Java
ONOS	Controlador recente, projetado desde o início para atender aos critérios de escalabilidade, alta disponibilidade e desempenho	Projeto em parceria	Java

Ethanol: Redes Sem Fio + SDN

- ▶ Controle de transmissões ARP
- ▶ QoS para tráfego Ethernet e sem fio
- ▶ Controle de processo de associação de estações sem fio
- ▶ Identificação de interfaces sem fio não ativas

"Ethanol: Software Defined Networking for 802.11 Wireless Networks." *IFIP/IEEE International Symposium on Integrated Network Management (IM)*. 2015.

Redes Definidas por Software, segundo seus criadores

“SDN will allow us to innovate on our own” (Nick McKeown, Stanford)

“A major change in paradigm” (Scott Shenker, Berkeley)

“The beginning of a software era in networking” (Scott Shenker, Berkeley)

“A common vision of making networking better through software”
(Martin Casado, VMware)

- Agradecimento: Alguns Slides foram tirados da palestra sobre SDN orquestrada pelo **Marcos Augusto Menezes Vieira**, da Universidade Federal de Minas gerais.