

# CMFLOW: Uma Extensão do Protocolo *OpenFlow* para Criação de Circuitos em Redes Multicamadas

João J. Salvatti, Fernando N. N. Farias, José M. Dias, Hugo S. Toda, Eduardo C. Cerqueira, Antônio J. G. Abelém

Grupo de Pesquisa em Redes de Computadores e Comunicação Multimídia (GERCOM)  
Universidade Federal do Pará (UFPA) – Belém – PA – Brasil

{salvatti, fernnf, jdias, toda, cerqueira, abelem}@ufpa.br

*Abstract.* This paper presents an *Openflow* datapath implementation that enables its use on multilayer circuit networks. It also describes two operation models, one that runs directly on the switch and another that works as a virtual datapath..

*Resumo.* Este artigo apresenta uma implementação do datapath *OpenFlow* que permite sua utilização em redes de circuitos multicamadas. São descritos os dois modelos de funcionamento, um que é executado diretamente no hardware do switch e outro que funciona como um datapath virtual.

## 1. Introdução

O *OpenFlow* [McKeown 2008] é uma proposta que permite a programabilidade, via um controlador, de *switches* e roteadores. É capaz de manipular fluxos de pacotes baseados em informações de seus cabeçalhos que vai da camada 2 até a 4 da arquitetura TCP/IP. No entanto, a utilização de redes de transporte pelo protocolo *OpenFlow* ainda é um desafio a ser explorado. Neste contexto, extensões do *OpenFlow* com suporte a manipulação de redes de circuitos ópticos são requisitos para a Internet do Futuro.

Portanto, este artigo apresenta uma proposta preliminar de extensão para o *OpenFlow* que permita a manipulação e gerenciamento de circuitos em redes multicamadas, tais como: circuitos em camada 2, baseados em VLAN (*Virtual Local Area Network*) ID, camadas 1, via *time-slot* em SONET/SDH (*Synchronous Optical Networking/Synchronous Digital Hierarchy*) e camada 0, utilizando valor de comprimentos de ondas em redes WDM (*Wavelength Division Multiplexing*)

Além desta seção introdutória, este trabalho está dividido da seguinte maneira: Seção 2, apresentando a proposta, como modelo proposto e modo de funcionamento; e a Seção 3 com as considerações finais e trabalhos futuros.

## 2. Extensão do Protocolo *OpenFlow* para Circuitos

Esta seção apresenta as extensões *OpenFlow* para redes de circuito multicamadas focando no protocolo e nos modos de funcionamento.

## 2.1 Modelo Proposto

O modelo atual do protocolo *OpenFlow*, distribuídos na versões 1.0 e 1.1, possui suporte para redes de pacotes e com isso toda a sua operação está relacionada com o cabeçalho ilustrado na Figura 1a.

In Port	VLAN ID	Ethernet			IP			TCP	
		SA	DA	Type	SA	DA	Proto	Src	Dst

(a)

desc_no	in_port	out_port	eth_type	vlan_id	vlan_prio

(b)

**Figura 1. (a) cabeçalho *OpenFlow* original, (b) extensão para suporte a circuitos.**

É através dos campos apresentados na Figura 1a, que as ações para os fluxos de pacotes são criadas no *datapath* atual do Protocolo *OpenFlow*. Isso o impossibilita de trabalhar com circuitos em redes multicamadas, pois como observado na Figura 1a, a estrutura não contempla as informações necessárias para manipulação dos mesmos nas principais tecnologias existentes, tais como: em redes *ethernet*, a manipulação de circuitos do tipo VLAN; SONET/SDH, a sincronização para o uso de *time slots*; e em WDM, a alocação de comprimento de onda.

Para aplicar o modelo proposto, o *datapath* foi estendido para habilitar o suporte a circuitos. A Figura 1b, exemplifica a extensão para circuitos baseados em *ethernet* e as informações utilizadas. A partir disso, foi possível fazer com que esta implementação pudesse adicionar, remover e gerenciar circuitos independentemente da tecnologia de transporte na rede.

O modelo proposto possibilita uma visão homogênea para o controlador, abstraindo de suas aplicações questões específicas de cada tecnologia de circuito supracitada. Isso é alcançado através de uma definição de alto nível dos elementos básicos que compõem um circuito, como por exemplo, largura de banda, Qualidade de Serviço (QoS), atraso ou variação do atraso. Além disso, estes dados são refletidos em configurações específicas para cada tipo de tecnologia de circuito.

## 2.2 Modo de Funcionamento

Foram desenvolvidos dois modelos de funcionamento para o novo *datapath* *OpenFlow*: *datapath* interagindo diretamente com o *hardware*; e *datapath* virtual.

### 2.2.1. *Datapath* *OpenFlow* Interagindo Diretamente com *Hardware*

Este modelo de *datapath* funciona da seguinte maneira: detecta o tipo de porta que o equipamento possui e, baseado nisso, carrega todas as informações necessárias para sua operação dentro de estruturas de dados adequadas para cada tipo de tecnologia. Por exemplo, em um *switch ethernet*, o módulo carrega informações como: os modos de *links* suportados; velocidade da porta; MTU ou tipo de negociação da interface. Já para um *switch* óptico, são carregadas informações como: quantidade de portas ou quantidade de comprimentos de ondas.

No entanto, o acesso a essas informações podem variar dependendo do suporte do equipamento as interfaces, de modo que algumas informações podem ser inseridas de forma estática, por exemplo, via um arquivo de configuração.

Quando um circuito é gerado ao longo de um caminho com diversas tecnologias subjacentes, o novo *datapath* recebe a mensagem *OpenFlow* e converte em ações diretas ao sistema operacional do equipamento. Para um *switch ethernet*, o *datapath* trabalhará com VLANs, já em um *switch* óptico, irá operar com comprimentos de ondas, logo é necessário que todos os *datapaths* ao longo do caminho saibam como converter informações de um *tipo*, por exemplo VLAN, para outro *tipo*, por exemplo, um comprimento de onda. Na extensão proposta, o *datapath* faz isso através de arquivos de configuração e funções de conversão.

Quando um circuito está estabelecido, o *datapath* mantém informações do mesmo, porém dependendo da tecnologia do *switch*, algumas informações podem mudar. Se for feita uma consulta sobre os circuitos estabelecidos em um *switch ethernet* a saída do mesmo mostrará informações como, porta de entrada, porta de saída, ID de VLAN, prioridade, tempo que o circuito está estabelecido e tempo para que ele expire, ou seja, o tempo que falta para que o circuito seja destruído pelo *datapath*.

Para o *datapath* da proposta foram adicionadas novas estruturas de dados e mensagens ao protocolo *OpenFlow* original, como a descrição do tipo de tecnologia que o protocolo irá suportar. As operações são baseadas no documento do projeto *OpenFlow* para suporte a circuitos [Das, 2009]. As portas *ethernets* não foram modificadas, porém, adicionou-se uma estrutura para descrever circuitos utilizando esta tecnologia (Figura 1b). As mensagens que foram adicionadas são para criação, remoção, e monitoramento de circuitos.

### 2.2.2. Datapath OpenFlow Virtualizado

Atualmente o uso do *OpenFlow* está em fase de padronização, podendo ser utilizado apenas em máquinas GNU/Linux funcionando como roteador ou *switch*, e alguns *switches* comerciais disponíveis no mercado, que possuem suporte *OpenFlow*. Deste modo, muitas vezes, fica inviável sua utilização em equipamentos de circuito sem suporte *OpenFlow*, devido o custo de mudança das tecnologias já utilizadas.

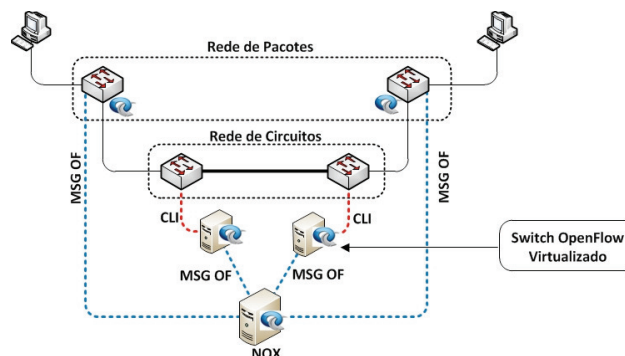


Figura 2. Funcionamento do *datapath* virtual.

Para solucionar este problema foi desenvolvido um *datapath OpenFlow* virtual que age como um *proxy* entre o equipamento que não suporta *OpenFlow* e o controlador *OpenFlow*. Para cada *switch* sem suporte *Openflow* de uma rede, deve-se ter um *datapath* virtual. Este executa no sistema operacional GNU/Linux, em uma máquina real ou virtual. A Figura 2 exibe o funcionamento do *datapath* virtual.

Ao ser inicializado, o *datapath* verifica um arquivo de configuração que o informa sobre o endereço IP, o tipo do *switch* (*ethernet*, SONET/SDH ou WDM), e o

método de acesso ao mesmo (CLI, Webservice ou SNMP). Utilizando um desses métodos e com as devidas credenciais de acesso, o *datapath* se inicializa fazendo consultas no *switch* e pegando todas as informações necessárias para seu correto funcionamento. Com estas informações, o *datapath* carrega um módulo para o *kernel* Linux que cria portas virtuais com os dados das portas do *switch* sem *OpenFlow*.

Posteriormente, o *datapath* procede com sua inicialização, registrando para si as portas virtuais do sistema Linux e não as do *switch*. Quando o controlador da rede requer uma nova configuração, como por exemplo, a criação de uma nova VLAN, o controlador manda a mensagem para o *datapath* virtual, este traduz a mensagem em comandos compreensíveis pelo *switch* usando um dos métodos de acesso, por exemplo, CLI. Se a rede possuir *switches* com suporte a *OpenFlow*, estes não necessitam de *datapaths* virtuais e se registram diretamente no controlador da rede.

### 3. Conclusões e Trabalhos Futuros

O *OpenFlow* permite que sejam testados protocolos experimentais em redes de produção. A atual implementação do *OpenFlow* só tem suporte para redes de circuitos. Além disso, não existe a compatibilidade entre equipamento com e sem *OpenFlow*.

O modelo proposto apresenta soluções para estes dois problemas. Para isso, foram implementados dois tipos de *datapaths*, um com suporte a circuitos nas principais tecnologias existentes, como *ethernet*, SONET/SDH e WDM. Permitindo que o controlador possa criar circuitos de uma maneira uniforme e sem se preocupar com as tecnologias subjacentes. O outro *datapath* funciona de modo virtual que recebe as mensagens *OpenFlow* e as traduz em comandos para o equipamento que não possui suporte ao protocolo. Isso possibilita a integração dos equipamentos sem a necessidade de troca do mesmo, com isso reduzindo custos e aumentando a flexibilidade.

Como trabalho futuro pretende-se continuar o desenvolvimento da solução, corrigindo problemas e refinando a implementação. Também pretende-se receber colaborações de outras instituições, uma vez que o código fonte estará disponível para *download*.

### Agradecimentos

Este trabalho foi desenvolvido com o apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e da Fundação de Amparo à Pesquisa do Estado do Pará (Fapespa).

### Referências

- Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. 2008. *OpenFlow*: enabling innovation in campus networks. SIGCOMM Comput. Commun. Rev. 38, 2 (March 2008), 69-74.
- Das, Saurav. 2009. Extensions to *OpenFlow* Protocol in support Circuit Switching. *OpenFlow*. Relatório Técnico. Disponível em: <http://www.OpenFlow.org/wp/wp-content/uploads/2009/09/OpenFlow-circuitspec-v02.pdf>