Uma Filosofia para Testes de Protocolos em Internet do Futuro usando *Openflow*

Eliel Marlon de Lima Pinto
CEAVI – UDESC
{marlon@inf.ufsc.br}

Tatianne Dias Moreira
PPGCC – UFSC
{tatianne.diasmoreira@gmail.com}

***Resumo* – Esse trabalho procura abordar uma nova filosofia para testes de protocolos, baseada numa padronização aberta, sem colocar em risco propriedade industrial de equipamento proprietários de redes e estender o volume de possibilidade de redes públicas e privadas para experimentações de protocolos.**

I. Introdução

Se considerarmos a heterogeneidade da rede mundial de computadores, seus roteadores, *switchs,* protocolos, equipamentos esses, com suas regras que se estendem desde a camada física até a camada de aplicação, pode-se tentar entender um pouco da necessidade de existir uma estrutura que ofereça flexibilidade, acessibilidade, segurança, liberdade e que esteja disponível a qualquer usuário para testes de redes e seus protocolos. Tudo isso, sem colocar em risco, informações sigilosas de fornecedores de produtos como roteadores, protocolos, *switchs.* Tentando atender a carências tão sensíveis, que o *OpenFlow* vem sendo estudado, desenvolvido e a cada dia aprimorado para que seja uma alternativa de teste sem comprometer sigilos industriais dos variados fabricantes de equipamentos de redes. Este artigo procura mostrar de forma mais simples a idéia do *OpenFlow*. Onde esse pode ser uma considerada opção à comunidade científica executar protocolos experimentais nas variações de redes utilizadas diariamente.

II. Iniciativas Relacionadas ao *OpenFlow*

Visando permitir experimentos de novos protocolos nas mais variadas arquiteturas de redes de computadores, iniciativas têm sido de grande relevância para um avanço significativo para a virtualização de redes, propondo a filosofia de redes virtuais programáveis. Nesse tipo de rede a experimentação de protocolos ocorre em larga escala. Dessa forma, tais redes devem dispor de roteadores e comutadores, distribuídos geograficamente, e esses poderão ser programados para executar protocolos experimentais. Iniciativas têm sido propostas visando a suporte necessário à filosofia *OpenFlow* como é o caso do *GENI* , o *NOX* e o *OPENVISION*.

A. GENI

Esse projeto é uma também uma iniciativa que propõe a virtualização de redes. Pois, cada pesquisador interessado em executar seus experimentos é atribuído uma parte dos recursos da rede, que envolve enlaces de rede, roteadores, comutadores e terminais clientes. Assim, o pesquisador tem a liberdade de configurar esses recursos como desejar sem interferir no curso de trabalho daqueles equipamentos [4].

B. NOX

Essa proposta visa facilitar o gerenciamento de redes de grande escala. A idéia básica é oferecer uma interface de alto nível para as aplicações utilizarem recursos de *hardware* e também controlar a interação entre essas aplicações [2]. A interface de alto nível oferecida tornou os softwares mais fáceis para desenvolvimento e de executarem em diferentes plataformas de *hardware*. Isso ocorre, pois os programadores não precisam mais se preocupar com interações de baixo nível da aplicação com o *hardware*, e escrevem seus *softwares* utilizando primitivas genéricas, que funcionam em diversas arquiteturas. Em redes de computadores o gerenciamento realizado por configurações de baixo nível independe do sistema operacional, mas é necessário um conhecimento prévio dessa por parte do pesquisador. Como o controle de acesso aos usuários de uma rede baseada numa ACL (*Access Control List* - Lista de Controle de Acesso) necessita do conhecimento do endereço IP do usuário, que é um parâmetro de baixo nível dependente da rede, não é diferente a necessidade do pesquisador ter conhecimento preliminar dos recursos que esse terá disponível. Assim, é imprescindível a necessidade de um sistema operacional de redes que forneça interfaces para controlar e analisar uma rede, semelhantes às interfaces de leitura e escrita em diversos recursos oferecidos por um sistema operacional de comum [2].

Contudo, o sistema operacional de redes deverá fornecer uma interface genérica de programação que permite o desenvolvimento de aplicações de gerenciamento da rede. Esse sistema centraliza o gerenciamento da rede, necessitando haver uma grande preocupação com sua escalabilidade. O sistemas operacional *NOX* é um exemplo de sistema operacional de rede que procura atender os requisitos já citados. Esse sistema foi desenvolvido para ser executado nos controladores de redes *OpenFlow*. Apesar do objetivo principal do *OpenFlow* ser o experimento de novas propostas, o *NOX* pode ser utilizado também no gerenciamento de redes de produção. A Figura 2 mostra os principais componentes de uma rede baseada no *NOX*.

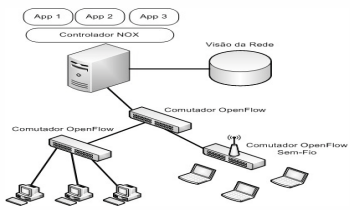

Figura 02 - Componentes de uma rede baseada no NOX.

Essas redes possuem um ou mais servidores executando o *NOX*. Cada um realiza o papel de

controlador *OpenFlow.* Nos controladores se encontram aplicações executando a partir do *NOX.* Todos os controladores compartilham uma única Visão da Rede, que é mantida na base de dados de um dos servidores.

A Visão da Rede é montada com base em informações da rede coletadas pelo *NOX,* essa é usada na tomada de decisões pelas aplicações de gerenciamento. As informações coletadas e armazenadas variam desde a topologia dos comutadores, passando pela localização dos elementos da rede como usuários, clientes até os serviços oferecidos. Nesse caso, as aplicações poderão manipular o tráfego da rede a partir de configuração remota dos comutadores *OpenFlow.* Esse controle é realizado no nível de fluxo, ou seja, sempre quando um determinado controle é realizado no primeiro pacote de um fluxo todos os outros pacotes desse fluxo sofrerão a mesma ação. O funcionamento de uma rede baseada no NOX ocorre quando: ao receber um pacote, o comutador verifica o cabeçalho do pacote para ver se ele está definido em sua Tabela de Fluxos. Se estiver definido, o comutador executa a ação especificada na Tabela de Fluxos e já atualiza os contadores referentes à estatística do fluxo.

C. *FlowVisor*

É um mecanismo muito útil que possibilita a virtualização de redes *OpenFlow* [3]. A virtualização de redes é um assunto bastante pesquisado atualmente [7].

Ela é importante para permitir que numa mesma infra-estrutura física possam coexistir redes com diferentes mecanismos de encaminhamento, endereçamentos, entre outros. Assim, os recursos da infra-estrutura necessitam ser divididos entre as diferentes partes da rede. Um exemplo de aplicação de virtualização das redes *OpenFlow* é permitir, por exemplo, que diferentes tipos de experimentos sejam conduzidos ao mesmo tempo na mesma rede de computadores de uma universidade. Os recursos da infraestrutura divididos pelo *FlowVisor* são [3]:

1) Banda Passante: Essa não passa de um enlace que deve ser dividido entre as partes da rede.

2) Topologia: Nesse caso, cada parte da rede deverá possuir sua própria visão dos nós da rede e a conectividade entre eles.

3) Tráfego: O tráfego de uma parte da rede não deve interferir na outra. Esse tráfego pode ser definido de várias formas como, por exemplo, todos os pacotes oriundos de um conjunto de endereços, todo o tráfego de um grupo de usuários ou algo mais específico como todo o tráfego HTTP.

4) CPU dos elementos da rede: A CPU dos comutadores e roteadores da rede deverão ser divididas entre as fatias. Isso é importante, por exemplo, em casos onde os pacotes passam pelo chamado *slow-path*, como no caso de pacotes IP com opções, no qual a CPU do comutador necessita ser utilizada.

5) Tabela de Encaminhamento: Considerando à limitação da capacidade dos comutadores para armazenarem regras de encaminhamento, há necessidade de controlar esses recursos entre as partes da rede.

Em uma rede com o *FlowVisor* cada parte da rede em teste possui um controlador *OpenFlow* com o *NOX*. O *FlowVisor* é um controlador especial que atua como um *proxy* entre os controladores *NOX* e os comutadores da rede. Todas as mensagens *OpenFlow*, tanto dos controladores para os comutadores, como no sentido oposto são interceptadas pelo *FlowVisor,* esse que decidirá o que fazer com as mensagens baseadas nas políticas de cada parte da rede testada. Na rede virtualizada pelo *FlowVisor* não é necessária a modificação dos controladores, nem dos comutadores da rede. A interceptação de mensagem é realizada de forma transparente a esses elementos.

Um exemplo de operação do *FlowVisor* está mostrado na figura 3, nela a rede é dividida entre três controladores. Dois deles são para os experimentos dos pesquisadores Maria e José e o outro controla o tráfego de produção. José realiza um experimento usando seu controlador que, por exemplo, atua como um balanceador de carga de tráfego HTTP. Nesse experimento o controlador dissemina tráfego para todos os servidores de um conjunto especificado previamente. As políticas da parte da rede dada para o José são feitas de tal forma que o seu controlador só enxerga os fluxos provenientes de um determinando IP de origem.

Devido à transparência do *FlowVisor*, o controlador do José acredita que pode controlar os fluxos para todo tráfego HTTP vindo de qualquer IP de origem. Assim, a exemplo da figura 3, quando o controlador do José envia uma mensagem para adicionar uma entrada na Tabela de Fluxos de um comutador, o *FlowVisor* intercepta essa mensagem (número 1). Após a interceptação, o *FlowVisor* consulta as políticas da fatia do José (número 2), e reescreve essa entrada para incluir apenas o tráfego oriundo do IP de origem permitido para José. Após isso envia uma mensagem com essa entrada para o comutador (número 3). Da mesma forma, as mensagens enviadas dos comutadores para o controlador do José são interceptadas pelo *FlowVisor* (número 4) [3].

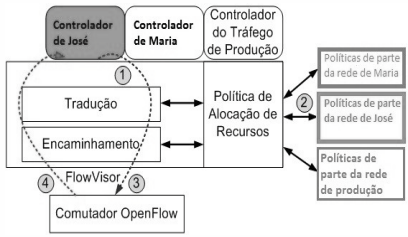

Figura 03: Esquema do *FlowVisor*

III. Proposição

O *OpenFlow* é baseado em um *switch Ethernet*, com um fluxo interno de mesa, e uma interface padronizada para adicionar e remover entradas de fluxo. Isso dará liberdade para fornecedores de rede para adicionar *OpenFlow* aos seus produtos, alternar para implantação de backbones de campus universitário e armários de cablagem. Se Acredita que o *OpenFlow* é um compromisso pragmático que por um lado permite que

pesquisadores executem experiências em switches heterogêneos de forma uniforme, a taxa de linha e com alta densidade de portas, enquanto por outro lado, os vendedores não precisam expor ao mercado o funcionamento interno de seus aparelhos. A maioria dos comutadores Ethernet e roteadores modernos possuem tabelas de fluxos que são utilizadas para implementar diferentes funcionalidades, como *QoS*(*Quality of Service* - Qualidade de Serviço), *Firewalls* e coleta de estatísticas. Assim, para programar o seu mecanismo, a proposta do *OpenFlow* tem como ideia básica utilizar funções de manipulação dessas tabelas de fluxos, que são oferecidas nos comutadores [1]. Como essas funções podem diferir de acordo com o fabricante, o *OpenFlow* identifica algumas funções comuns à maioria dos comutadores e roteadores objetivando tornar-se algo comum por um maior número de equipamentos.

A partir dessas funções, o *OpenFlow* fornece um protocolo aberto para programar as tabelas de fluxos desses equipamentos. O plano de dados de um Comutador *OpenFlow* consiste em uma Tabela de Fluxos, com uma ação correspondente a cada um desses fluxos. Portanto, isso permite que os fluxos sejam tratados de maneira diferente. Assim, pode ser definido um fluxo de tráfego experimental e outro de tráfego de produção, possibilitando isolamento entre eles.

Esse tráfego de produção é tratado da mesma forma que seria na ausência do *OpenFlow*. Com isso, ele pode ser visto com uma generalização do conceito de VLANs, que também permitem o isolamento de tráfego mas de forma menos flexível [1]. Como o *OpenFlow* passa ao pesquisador a liberdade na definição de um fluxo, esse pode configurar as tabelas de fluxo especificando a rota que seu tráfego irá percorrer e como os pacotes dos seus fluxos serão tratados, permitindo a experimentação de novas propostas. Um Comutador *OpenFlow* é composto por, no mínimo, três partes: a Tabela de Fluxos, que possui uma ação para cada fluxo definido, como explicitado anteriormente; o Canal Seguro que conecta o Comutador ao Controlador da rede. Esse Controlador é utilizado para a configuração da Tabela de Fluxos e por último o Protocolo *OpenFlow*, que possibilita a comunicação do Comutador com o Controlador. Com esse protocolo, os pesquisadores podem configurar a Tabela de Fluxos sem a necessidade de programar o comutador [1]. Existem dois tipos de comutadores *OpenFlow*, comutador dedicado, que não suporta encaminhamento comum de Nível 2 e Nível 3; e um comutador ou roteador comercial com *OpenFlow* habilitado que, além de suportar as funcionalidades do *OpenFlow*, realiza as funções comuns de um comutador ou roteador.

A. Comutador OpenFlow dedicado

Esse tipo de comutador, exemplificado na figura 4, é um elemento que apenas encaminha o tráfego entre as portas do comutador de acordo com a Tabela de Fluxos configurada remotamente. O fluxo pode ser definido de diferentes formas. Por exemplo, um fluxo pode ser definido como todos os pacotes provenientes de certa porta TCP ou os pacotes com um determinado endereço MAC de destino. Além disso, alguns comutadores OpenFlow podem tratar pacotes que não utilizam o IPv4, verificando campos arbitrários de seu cabeçalho. Isso mostra a flexibilidade do OpenFlow para suportar diferentes tipos de experimentos.

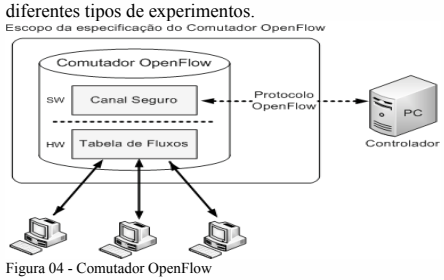

Figura 04 - Comutador OpenFlow

Para cada entrada da Tabela de Fluxos é definida uma ação para ser tomada com os pacotes oriundos de um determinado fluxo. As três ações básicas que todos Comutadores OpenFlow devem suportar, são as seguintes:

1) Encaminhar os pacotes do fluxo para uma (ou várias) porta(s) específica(s): Isso permite que os pacotes sejam roteados pela rede.

2) Encapsular os pacotes e encaminhá-los para o Controlador utilizando o Canal Seguro de comunicação: Tal ação pode ser executada no momento do envio do primeiro pacote de um fluxo, que ainda não tenha sido definido nas Tabelas de Fluxo dos comutadores, com o objetivo do Controlador configurar os comutadores com esse novo fluxo. Além disso, a ação pode ser realizada em um experimento no qual é necessário processamento adicional nos pacotes de um fluxo.

3) Descartar o pacote: Essa ação pode ser utilizada em aplicações de segurança para impedir, por exemplo, ataques de negação de serviço.

Uma entrada na Tabela de Fluxos possui três campos. O primeiro identifica o cabeçalho, que define o fluxo. Por exemplo, no cabeçalho da primeira geração de comutadores OpenFlow (comutadores "tipo 0"), um fluxo pode ser definido por 10 parâmetros. Esses parâmetros podem ser o MAC e IP de origem ou destino, as portas TCP usadas, entre outros. Em outras gerações de comutadores OpenFlow esses parâmetros podem ser definidos arbitrariamente, permitindo o experimento de protocolos que não utilizam o IP. O segundo campo identifica a ação a ser tomada e o terceiro armazena as estatísticas do fluxo. Essas estatísticas podem ser o número de pacotes ou bytes, referentes ao fluxo que passaram pelo comutador e o intervalo de tempo, desde a última vez que um pacote do fluxo foi identificado pelo comutador. Esse último é importante, por exemplo, para remoção de um fluxo da tabela caso esteja inativo.

B. Comutador com OpenFlow habilitado:

Este comutador consiste nos equipamentos que já realizam encaminhamento normal de Nível 2 e Nível 3, mas adicionaram o *OpenFlow* com uma funcionalidade. Dessa forma, o tráfego de produção pode ser encaminhado

utilizando o encaminhamento normal e permanece isolado do tráfego experimental, que utiliza o mecanismo do *OpenFlow*. Para isso, esses comutadores podem adicionar outra ação além das três ações básicas citadas anteriormente:

1) Encaminhar os pacotes do fluxo pelo pipeline normal do comutador: Nessa ação, um fluxo identificado como não sendo referente ao *OpenFlow* será processado utilizando o encaminhamento normal. Como alternativa ao uso da ação anterior, um comutador pode isolar o tráfego de produção do tráfego *OpenFlow* através do uso de VLANs. Alguns comutadores podem suportar tanto essa alternativa como a anterior.

IV. Ambiente de Utilização

Como visto anteriormente, o OpenFlow permite o experimento de novas proposta na área de Redes de Computadores utilizando uma infraestrtura de rede já existente nas universidades. Dentre os possíveis experimentos permitidos podem ser citados os seguintes [1]:

A. Novos Protocolos de Roteamento: Os algoritmos de um protocolo de roteamento podem ser implementados, para executarem no Controlador de uma rede *OpenFlow*. Assim, quando um pacote do experimento chegar a um comutador ele é encaminhado para o Controlador, que é responsável por escolher a melhor rota para o pacote seguir na rede a partir dos mecanismos do protocolo de roteamento proposto. Após isso, o Controlador adiciona entradas na Tabela de Fluxos de cada comutador pertencente a rota escolhida. Os próximos pacotes desse fluxo que forem encaminhados na rede não necessitarão serem enviados para o Controlador.

B. Mobilidade: Uma rede *OpenFlow* pode possuir pontos de acesso sem-fio permitindo que, clientes móveis utilizem sua infra-estrutura, para se conectarem a Servidores ou à Internet. Assim, mecanismos de *handoff* podem ser executados no Controlador realizando alteração dinâmica das tabelas de fluxo dos comutadores de acordo com a movimentação do cliente, permitindo a redefinição da rota utilizada.

C. Redes Não-IP: Como visto anteriormente, um comutador *OpenFlow* pode ser desenvolvido para analisar campos arbitrários de um pacote, permitindo flexibilidade na definição do fluxo. Assim, podem ser testadas, por exemplo, novas formas de endereçamento e roteamento. Nos comutadores do "tipo 0", os pacotes Não-IP podem ser definidos pelo endereço MAC de origem, ou destino ou a partir de um novo Tipo de Ethernet ou IP.

D. Redes com Processamento de Pacotes: Existem propostas de protocolos que realizam processamento de cada pacote de um fluxo como, por exemplo, os protocolos de Redes Ativas [5]. Esse tipo de proposta pode ser implementado no *OpenFlow* forçando que, cada pacote que um comutador receba e seja enviado para o Controlador para ser processado. Outra alternativa é enviar esses pacotes para processamento por um comutador programável, como é o caso de um roteador programável baseado em Net-FPGA [6].

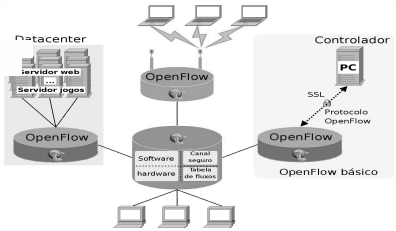

Figura 5: Exemplo de Rede com *OpenFlow*

V. Conclusão

Neste trabalho, se procurou abordar uma nova metodologia para experimentação de novas tecnologias de redes. A filosofia abordada no decorrer do texto procurou clarear e desmistificar a filosofia de trabalho que o *OpenFlow* e seus componentes propõem para a Internet do Futuro. Mas é necessário prudência e muitos estudos ainda, como forma de amadurecimento da tecnologia. Por outro lado, o fato de já existir uma filosofia aberta que permita tanto aos pesquisadores, quanto aos fornecedores de produtos estudos e teste de maneira segura e eficaz, nos deixa esperançoso para o aprimoramento de uma filosofia tanto no âmbito do *OpenFlow* Dedicado, quanto no *OpenFlow* Habilitado. Não deixando de considerar variedade de experimentos que o *OpenFlow* permite, essa ferramenta possibilitará um maior avanço na área de Redes de Computadores. Assim, muitas novas propostas que são validadas apenas com simulação, poderão passar a contar com experimentos em cenários reais. No Brasil, diversos projetos já utilizam o *OpenFlow* como é o caso do Horizon [8], que é um projeto sobre Internet do Futuro envolvendo universidades e empresas brasileiras e francesas.

REFERÊNCIAS

[1]. N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. "OpenFlow: enabling innovation in campus networks", ACM SIGCOMM Computer Communication Review, 38(2):69–74, 2008.

[2]. N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker, "NOX: Towards an operating system for networks", ACM SIGCOMM Computer Communication Review, 2008.

[3]. R. Sherwood, G. Gibb, K. Yap, G. Appenzeller, N. McKeown, and G. Parulkar. "Flowvisor: A network virtualization layer", Relatório Técnico, 2009. GENI: Global Environment for Network Innovations, http://www.geni.net/ - Accessado em Novembro/2010.

[4]. D. L. Tennenhouse, J. M. Smith, W. D. Sincoskie, D. J.Wetherall, and G. J. Minden. "A survery of active network research", IEEE Communications, 1997.

[5]. J. Lockwood, N. McKeown, G. Watson, G. Gibb, P. Hartke, J. Naous, R. Raghuraman, and J. Luo. "NetFPGA–An Open Platform for Gigabit-rate Network Switching and Routing", IEEE International Conference on Microelectronic Systems Education, 2007.

[6].Fernandes, N. C., Moreira, M. D. D., Moraes, I. M., Ferraz, L. H. G., Couto, R. S., Carvalho, H. E. T., Campista, M. E. M., Costa, L. H. M. K., and Duarte, O. C. M. B. - "Virtual Networks: Isolation, Performance, and Trends", Annals of Telecommunications, 2010.

[7]. Horizon Project: A New Horizon to The Internet, http://www.gta.ufrj.br/horizon/ - Accessado em Novembro/2010.

[8]. Macapuna, Carlos A. B. OpenFlow e NOX: Propostas para Experimentação de Novas Tecnologias de rede www.macapuna.com.br/index/phocadownload/.../main-of**nox**.pdf. Acessado em 10 de agosto de 2011.