UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

TRABALHO DE CONCLUSÃO DE CURSO

PROXY ROUTEFLOW BASEADO EM JAVA

FABIANO SILVA MATHILDE

ORIENTADOR: PROF. DR. CESAR AUGUSTO CAVALHEIRO MARCONDES
COORIENTADOR: DR. CHRISTIAN ESTEVE ROTHENBERG

São Carlos – SP

Dezembro/2012

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

TRABALHO DE CONCLUSÃO DE CURSO

PROXY ROUTEFLOW BASEADO EM JAVA

FABIANO SILVA MATHILDE

Dissertação apresentada ao Departamento de Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Bacharel em Engenharia de Computação, área de concentração: Sistemas Distribuídos e Redes

Orientador: Prof. Dr. Cesar Augusto Cavalheiro Marcondes

São Carlos – SP

Dezembro/2012

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

TRABALHO DE CONCLUSÃO DE CURSO

PROXY ROUTEFLOW BASEADO EM JAVA

FABIANO SILVA MATHILDE

Dissertação apresentada ao Departamento de Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Bacharel em Engenharia de Computação, área de concentração: Sistemas Distribuídos e Redes
Orientador: Prof. Dr. Cesar Augusto Cavalheiro Marcondes

Aprovado em 09 de Dezembro de 2012.

Membros da banca:

Prof. Dr. Cesar Augusto Cavalheiro Marcondes
(Orientador– DC-UFSCar)

São Carlos – SP

Dezembro/2012

RESUMO

Nos últimos anos o protocolo *OpenFlow* vem aumentando a visibilidade das tecnologias de redes definidas por software, fazendo com que um número cada vez maior de pesquisadores e desenvolvedores o adotem como principal ferramenta para simulações ou aplicações em ambientes reais. O projeto comunitário *RouteFlow* liderado pelo *CPqD* propõe uma plataforma de roteamento IP definido por software (do termo em inglês, *software-defined networking*) baseado no protocolo *OpenFlow*, que permite uma separação efetiva do plano de controle do plano de encaminhamento dos equipamentos de rede. O fato do sistema ter sido criado com o código totalmente aberto fez com que o número de usuários aumenta-se consideravelmente incentivando a equipe de desenvolvedores a atualizá-la constantemente para agregar cada vez mais ferramentas e tecnologias. O *RouteFlow* faz a manipulação do protocolo *OpenFlow* utilizando os softwares de controle mais famosos da literatura, o *NOX*, criado totalmente em *C++* e o *POX*, criado totalmente em *Python*. Para aumentar a capacidade do *RouteFlow* o trabalho em questão descreve a adição de suporte à um novo software de controle, o *Floodlight*, sendo criado totalmente em *Java*. Sendo assim o *RouteFlow* ganhará suporte a mais uma tecnologia mantendo-se sempre na vanguarda das tecnologias de redes definidas por software.

Palavras-chave: Redes Definidas por Software

ABSTRACT

Ainda a ser feito.

Keywords: Software Defined Network

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

ISP – *Internet Service Provider*

IP – *Internet Protocol*

UDP – *User Datagram Protocol*

TCP – *Transmission Control Protocol*

OF – *OpenFlow*

ARP – *Address Resolution Protocol*

MPLS – *Multiprotocol Label Switching*

SUMÁRIO

Capítulo 1

INTRODUÇÃO

1.1 Objetivo do Trabalho

O trabalho tem como objetivo principal agregar um nova funcionalidade ao *RouteFlow*. A funcionalidade será o suporte nativo ao controlador *Floodlight*. Os controladores são usados pelo *RouteFlow* como uma interface de comunicação entre os softwares de roteamento e os switches *Openflow*. Cada controlador possui certas características juntamente com recursos exclusivos e com isso esperá-se que o *RouteFlow* agregue as melhores ferramentas disponíveis no controlador *Floodlight*.

1.2 Contribuições

Como principal contribuição do trabalho podemos citar a integração que haverá entre o *RouteFlow* e a comunidade de usuários do controlador *Floodlight*. A comunidade poderá realizar simulações ou até experimentos em ambientes reais contribuindo ainda mais com o avanço do *RouteFlow*. Outra contribuição importante que pode ser citada é a absorção pelo *RouteFlow* das melhores ferramentas providas pelo *Floodlight*, tornando-o cada vez mais completo.

Capítulo 2

ARQUITETURA BÁSICA DO ROUTEFLOW

2.1 Descrição do Protocolo Openflow

O *OpenFlow* foi proposto pela Universidade de Stanford para atender à demanda de validação de novas propostas de arquiteturas e protocolos de rede (incluindo as abordagens *clean slate*) sobre equipamentos comerciais. O *OpenFlow* define um protocolo-padrão para determinar as ações de encaminhamento de pacotes em dispositivos de rede, como, por exemplo, comutadores, roteadores e pontos de acesso sem fio. As regras e ações instaladas no hardware de rede são responsabilidade de um elemento externo, denominado controlador, que pode ser implementado em um servidor comum, conforme Figura 2.1.

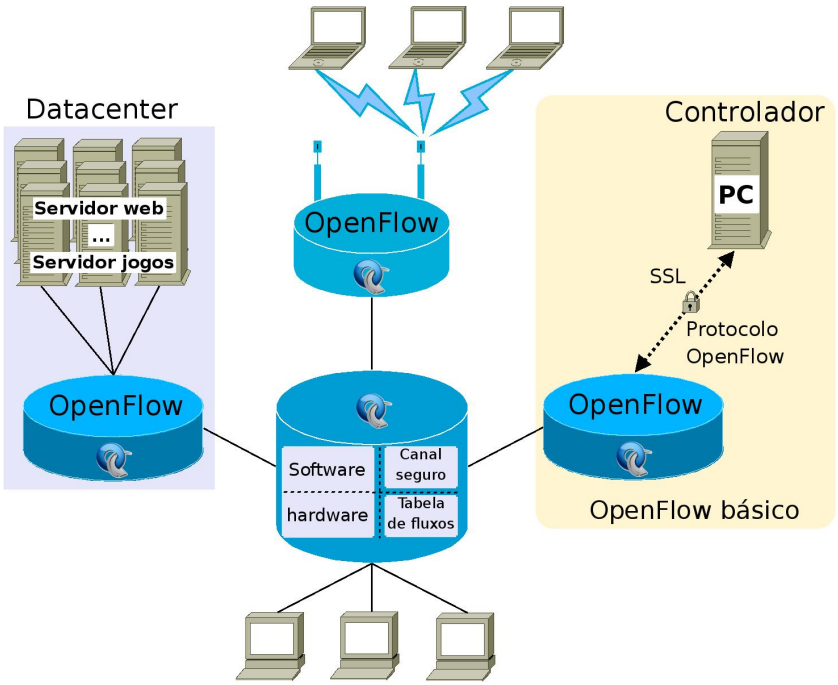

Figura 2.1: Rede com OpenFlow Habilitado

A principal abstração utilizada na especificação *OpenFlow* é o conceito de fluxo. Um fluxo é constituído pela combinação de campos do cabeçalho do pacote a ser processado pelo dispositivo, conforme Figura 2.2. As tuplas podem ser formadas por campos das camadas de enlace, de rede ou de transporte, segundo o modelo *TCP/IP*. Deve-se enfatizar que a abstração da tabela de fluxos ainda está sujeita a refinamentos, com o objetivo de oferecer uma melhor exposição dos recursos do hardware e, nesse caso, permitir a concatenação de várias tabelas já disponíveis, como, por exemplo, tabelas *IP/Ethernet/MPLS*. Nesse sentido, a contribuição mais importante do paradigma do *OpenFlow* é a generalização do plano de dados – qualquer modelo de encaminhamento de dados baseado na tomada de decisão fundamentada em algum valor, ou combinação de valores, dos campos de cabeçalho dos pacotes pode ser suportado.

inport	Ethernet			VLAN		IP				TCP/UDP	
	src	dst	type	id	pri	src	dst	proto	ToS	src	dst

Figura 2.2: Cabeçalho OpenFlow para a especificação dos fluxos

De forma pragmática, a especificação *OpenFlow* (OPENFLOW, 2010) procura reutilizar as funcionalidades do hardware existente (por exemplo, Access Control List – ACL em switches e roteadores para implementar serviços como NAT, firewall e VLANs) através da definição de um conjunto simples de regras e das ações associadas: encaminhar, descartar, enviar para o controlador, reescrever campos do cabeçalho, etc.

2.2 Introdução ao RouteFlow

O RouteFlow é uma proposta de oferta de serviços de roteamento IP remoto de forma centralizada, e que visa um desacoplamento efetivo entre o plano de encaminhamento e o plano de controle (ROUTEFLOW, 2011). O objetivo é tornar as redes IP mais flexíveis pela facilidade de adição, remoção e especialização de protocolos e algoritmos. O RouteFlow armazena a lógica de controle dos switches OpenFlow na infraestrutura de rede, através de uma rede virtual composta por máquinas virtuais (MV), cada uma executando um código (engine) de roteamento de domínio público (open source). Essas MVs podem ser interconectadas de maneira a formar uma topologia lógica, espelhando a topologia de uma rede física correspondente ou uma topologia virtual simplificada. O ambiente virtual é armazenado em um servidor externo, ou um conjunto deles, que se comunicam com os equipamentos do plano de dados através de um controlador OpenFlow, que transporta para o plano de encaminhamento as decisões tomadas pelos protocolos de roteamento no plano de controle (OSPF, BGP, RIP). A Figura 2.3 ilustra uma subrede com switches programáveis, em que a lógica de roteamento é implementada no servidor

RouteFlow. O resultado consiste numa solução flexível de alto desempenho e comercialmente competitiva, a partir da combinação de recursos disponíveis, como, por exemplo:

a) switches programáveis de baixo custo e software embarcado reduzido (OpenFlow);

b) pilhas de protocolos de roteamento open source (QUAGGA, 2009; XORP, 2011); e

c) servidor de prateleira de alto poder de processamento e, também, de baixo custo.

Cabe ressaltar que, apesar de o controle estar fisicamente centralizado, ele continua distribuído logicamente. Dessa forma, não é necessária qualquer alteração dos protocolos de roteamento existentes. Além disso, a solução pode tornar-se mais escalável no futuro, com o uso de vários servidores de alto desempenho.

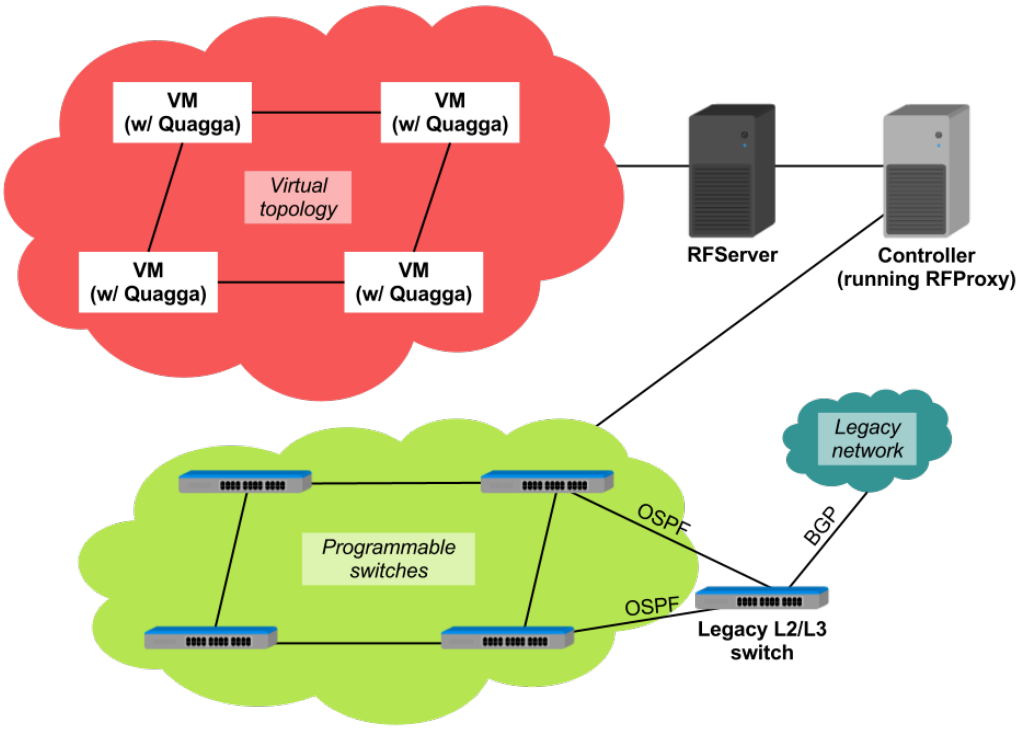

Figura 2.3: Visão geral do RouteFlow

2.3 Descrições dos Principais Componentes do RouteFlow

O RouteFlow é dividido basicamente em três aplicações básicas: RFClient, RFServer e RFProxy. Na Figura 2.4 temos uma visão geral das aplicações:

- RFClient executa como um programa executável em um máquina virtual, detectando mudanças na tabela ARP do Linux e na tabela de roteamento. As informações de roteamento são enviadas para o RFServer quando são atualizadas.

- RFServer é uma aplicação independente que gerencia as máquinas virtuais que estão executando o RFClient. O RFServer mantem o mapeamento entre as instâncias das máquinas virtuais executando o RFClient e as interfaces correspondentes aos switches e suas respectivas portas. É conectado ao RFProxy para instruí-lo como configurar os fluxos e também como configurar o Open vSwitch que mantém a conectividade de todo o ambiente composto pelas máquinas virtuais.

- RFProxy é uma aplicação responsável pelas interações entre os switches OpenFlow (identificados pelos seus datapaths) via o protocolo OpenFlow. Ele aguarda instruções do RFServer e o notifica à respeito de eventos na rede. Atualmente é executado como um módulo vinculado aos controladores OpenFlow. O RouteFlow tem suporte aos controladores NOX e POX, sendo que a proposta do trabalho é adicionar suporte ao Floodlight.

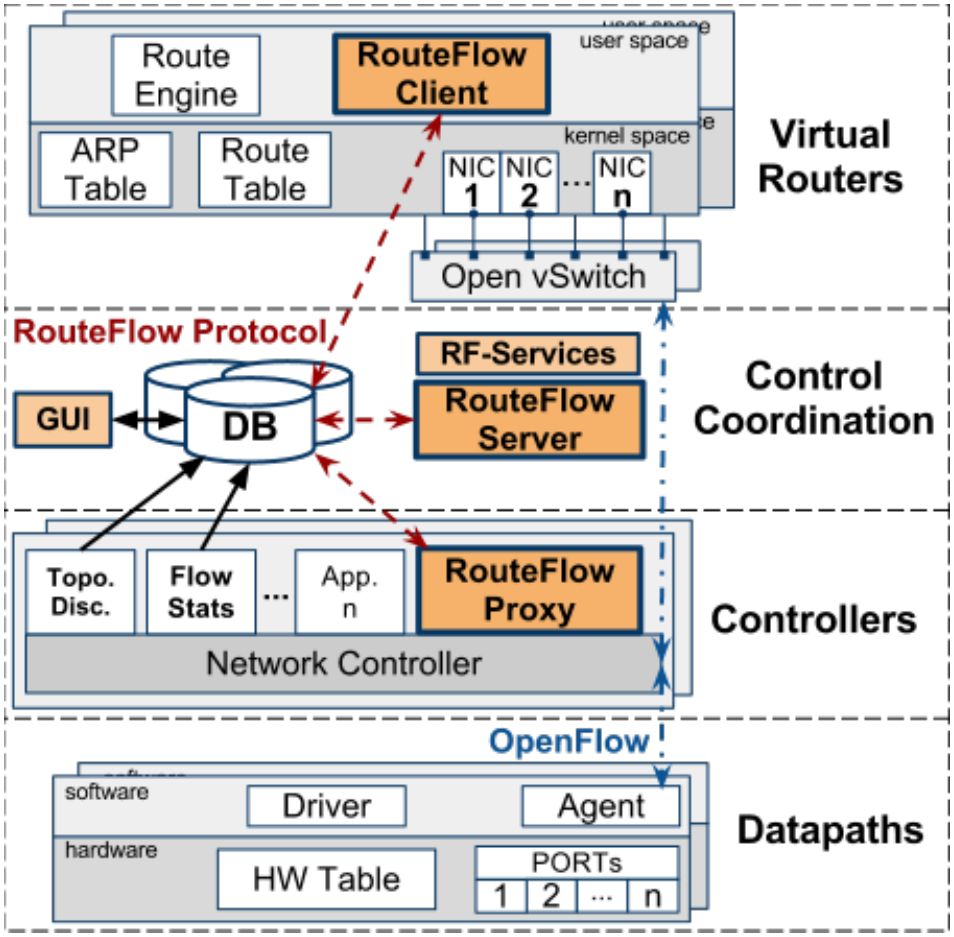

Figura 2.4: Componentes principais do RouteFlow

2.4 Protocolo RouteFlow

É o protocolo desenvolvido e usado para a comunicação entre os componentes do RouteFlow. Nele, estão definidas as mensagens e os comandos básicos para conexão e configuração das máquinas virtuais e, também, gerenciamento das entradas de roteamento em hardware. Entre os campos da mensagem-padrão estão: identificação do controlador, identificação da máquina virtual, tipo da mensagem, comprimento e dados. O proxy RouteFlow recebe os comandos do servidor RouteFlow através deste protocolo e de acordo com o tipo de comando executa as principais ações, muitas delas exigindo a comunicação com os switches físicos. A comunicação com os switches físicos é feita via protocolo OpenFlow, fazendo o proxy RouteFlow agir como uma espécie de ”tradutor”entre os dois protocolos.

Capítulo 3

PROXY ROUTEFLOW EM JAVA

3.1 Descrição da Estrutura do Proxy RouteFlow em Java

Todos os componentes do proxy RouteFlow em Java foram agrupados em classes. O agrupamento em classes facilita a organização geral do código bem como a sua futura manutenção. Abaixo temos os componentes do proxy e sua descrição:

- *MongoIPCMessageService:* responsável pela comunicação entre o servidor RouteFlow e o proxy RouteFlow;
- *RFProtocolFactory:* responsável pela criação das mensagens do protocolo RouteFlow.

3.2 Descrição das Mensagens Traduzidas pelo Proxy RouteFlow

Capítulo 4

RESULTADOS

Capítulo 5

CONCLUSÃO

5.1 Trabalhos Futuros

REFERÊNCIAS