OFNMS: Gerenciamento de Redes com OpenFlow

Jones F. R. Nickel
Faculdade de Informática – PUCRS
jones.nickel@acad.pucrs.br

Cristina M. Nunes
Faculdade de Informática – PUCRS
cristina.nunes@pucrs.br

***Resumo* — Este trabalho apresenta uma ferramenta para o monitoramento de *switches* em redes OpenFlow, visando ajudar o administrador de redes nas tarefas do dia a dia. A ferramenta possibilita a coleta de dados estatísticos de interfaces e fluxos nos *switches* da rede, apresentando estes dados graficamente.**

I. INTRODUÇÃO

O protocolo OpenFlow [2][4] vem prover-nos uma forma centralizada de controle da lógica de encaminhamento de pacotes da rede. A partir deste, um controlador se conecta nos *switches* de forma centralizada e manipula suas tabelas de encaminhamento de mensagens [2][4]. Ao chegar um pacote a um determinado *switch*, este verifica se existe alguma regra para o pacote, se existir aplica-a e o encaminha, caso contrário envia o pacote para o controlador decidir.

Este trabalho visa explorar as novas tecnologias providas pelo OpenFlow. Com base nas informações apresentadas a seguir, este trabalho descreve a implementação de uma ferramenta de monitoramento para redes OpenFlow, chamada de OFNMS (*OpenFlow Network Monitor System*). Tal ferramenta é baseada no controlador NOX [5] [6], com o objetivo de persistir dados dos switches da rede e expor estes dados na forma de informações úteis para a tomada de decisão na gerência dos recursos da mesma. A ferramenta possui uma interface web e apresenta os dados por meio de gráficos.

Este documento está organizado da seguinte forma: na Seção II são apresentados conceitos do protocolo OpenFlow, e a seguir o controlador NOX que serviu de base para este trabalho. A Seção III apresenta as principais ferramentas e trabalhos de pesquisa em destaque, baseados no protocolo OpenFlow e no controlador NOX. A Seção IV apresenta detalhes do desenvolvimento da ferramenta descrita neste trabalho. A Seção V descreve os objetivos alcançados, mostrando brevemente a ferramenta. Por fim, na Seção VI são apresentadas as considerações finais, apresentando a conclusão sobre este trabalho e trabalhos futuros que poderão complementá-lo.

II. FUNDAMENTAÇÃO TEÓRICA

A seguir serão apresentados alguns conceitos fundamentais para que se possa haver um melhor entendimento sobre a arquitetura e o funcionamento de uma rede OpenFlow. Também será apresentado o NOX, que serviu de base para a implementação do OFNMS (*OpenFlow Network Monitor System*).

A. OpenFlow

OpenFlow [2][3][4] é um protocolo aberto que permite estabelecer conexões entre um "Controlador Remoto" e os *switches* e roteadores que fazem parte da topologia da rede. O OpenFlow permite que o controlador manipule remotamente as regras de encaminhamento de mensagens dos equipamentos, obtendo informações dos mesmos e possibilitando a alteração dessas regras.

A arquitetura OpenFlow possui os componentes apresentados na Figura 1 e descritos a seguir:

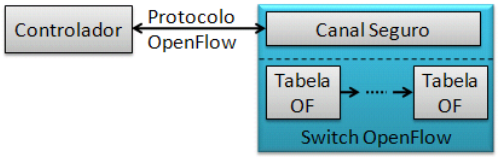

Figura 1 – Arquitetura OpenFlow.

1) Canal OpenFlow: canal seguro que conecta cada *switch* OpenFlow a um controlador. Através desse canal, o controlador configura e gerencia o *switch*, recebe eventos e envia pacotes através do mesmo.

2) Tabelas OpenFlow: formam um *pipeline* de processamento de fluxo. Cada tabela de fluxo contém múltiplas entradas de fluxo. O *pipeline* OpenFlow define como os pacotes recebidos pelo *switch* interagem com as tabelas de fluxo [3].

3) Protocolo OpenFlow: as mensagen suportadas pelo protocolo incluem: "controlador para *switch*", que tem início no controlador e são usadas para gerenciar ou inspecionar diretamente o estado do *switch*; "mensagens assíncronas", que têm início no *switch* e são usadas para notificar o controlador sobre os eventos da rede e mudanças no estado do *switch*; e "mensagens simétricas", que são iniciadas tanto pelo *switch* quanto pelo controlador e são enviadas sem solicitação.

B. NOX: Um Sistema Operacional de Rede

O NOX é um controlador OpenFlow de código aberto [5][6]. Foi concebido para fornecer uma plataforma simples de desenvolvimento de aplicativos que controlem ou monitorem redes OpenFlow. O NOX é o que pode ser considerado como um "Sistema Operacional de Rede" [6]. Ele fornece uma API de alto nível que possibilita conectar-se aos *switches* OpenFlow, podendo alterar suas tabelas de fluxo, buscar estatísticas dos *switches* e consultar o estado da rede.

O NOX mantém uma visão da rede [6], ou seja, uma base de dados com a observação do estado da rede (*Network View*). As aplicações usam essa visão para tomar decisões de administração. Para o NOX controlar o tráfego da rede, ele deve manipular os *switches* que compõem a topologia.

Quando um pacote recebido pelo *switch* encontra uma entrada de fluxo em suas tabelas, o *switch* aplica as ações correspondentes a esse pacote. Se o pacote não encontrar uma entrada de fluxo, ele é encaminhado para o

controlador, no caso o NOX. A aplicação de controle que roda sobre a API do NOX decide a regra de controle para o pacote e adiciona essa regra na tabela de fluxo do *switch*. Após essa ação todos os pacotes com mesmo cabeçalho utilizarão esta mesma regra [5][6].

III. TRABALHOS RELACIONADOS

A seguir estão apresentadas as principais ferramentas e trabalhos de pesquisa em destaque hoje baseados no protocolo OpenFlow e no controlador NOX.

A. *OMNI (Openflow MaNagement Infrastructure)*

O OMNI é uma ferramenta de código aberto que provê uma interface remota de administração para redes OpenFlow, possibilitando gerenciar facilmente essas redes [1]. A ferramenta é baseada em uma arquitetura orientada a serviços, permitindo o monitoramento e a configuração dinâmica dos encaminhamentos da rede. O OMNI dispõe de uma interface web que permite a configuração das rotas na rede. A arquitetura da ferramenta utiliza como base o controlador NOX modificado para seu propósito, onde foram desenvolvidos componentes que dispõem informações necessárias para a sua interface web.

As principais funcionalidades que a ferramenta dispõe incluem: visualizar as estatísticas da rede OpenFlow, adicionar novos fluxo, remover fluxos, migrar fluxos de um caminho para outro, monitorar os fluxos, tomar ações em um cenário com perda de pacotes e administrar uma rede OpenFlow.

B. *RouteFlow*

O RouteFlow é uma proposta de plataforma de roteamento remoto e centralizado [3]. Esta tecnologia visa o desacoplamento do plano de encaminhamento e o plano de controle, flexibilizando as redes IP quanto à facilidade de adição, remoção e especialização de protocolos e algoritmos. O RouteFlow utiliza máquinas virtuais (MVs) para representar a rede física através de uma topologia lógica de roteamento. Dessa forma, cada MV roda uma engine de roteamento padrão, onde suas interfaces representam as portas dos roteadores. Assim, a lógica de encaminhamento é executada na topologia virtual pelas engines de roteamento, separando o plano de controle do plano de dados. Após as decisões serem tomadas no plano virtual, estas são enviadas para o controlador que as instala nos respectivos elementos físicos da rede [9].

C. *QuagFlow*

O QuagFlow [10] é uma proposta de união transparente, não modificada, da suíte de roteamento de código aberto Quagga [10] com a rede OpenFlow. O QuagFlow explora a viabilidade de mover completamente a pilha de protocolos legados para controladores centralizados utilizando o protocolo OpenFlow [10]. Considera um passo intermediário rumo às redes programáveis para garantir a interoperabilidade com as redes legadas, possibilitando um benefício imediato em uma parceria entre o controle remoto e Quagga.

O QuagFlow replica a topologia física configurando as MVs em uma topologia virtual rodando o plano de controle do Quagga. A arquitetura do QuagFlow é composta por um controlador QuagFlow (QF-C), desenvolvido como um componente do NOX, e uma série de serviços QuagFlow (QF-S) rodando transparente nas MVs [10], onde estas hospedam o Quagga.

Durante a pesquisa na literatura, foram encontradas ferramentas que possibilitam a administração da rede OpenFlow, como por exemplo o OMNI. Entretanto, não foi identificada, até o momento, nenhuma que possibilite ao administrador de redes fazer uma análise do comportamento e saturação dos recursos da rede OpenFlow.

IV. OFNMS: OPENFLOW NETWORK MONITOR SYSTEM

A ferramenta desenvolvida neste trabalho tem como objetivo coletar periodicamente e de forma automática, os dados estatísticos dos *switches* em uma rede OpenFlow, persistindo esses dados em uma base de dados e possibilitando a consulta e geração de gráficos a partir desta. Dessa forma, será possível que o administrador da rede realize uma análise do comportamento dos recursos da mesma, como por exemplo, reconhecer o período em que os mesmos ficam sobrecarregados, ou estimar o crescimento futuro no uso da rede OpenFlow.

A. *Arquitetura*

A ferramenta desenvolvida foi dividida em três componentes principais, como ilustrado na Figura 2, sendo descritos a seguir.

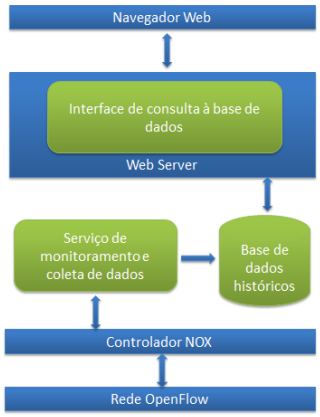

Figura 2 – Arquitetura da Ferramenta OPNMS.

1) *Base de dados históricos*

Responsável por armazenar os dados coletados pelo serviço de monitoramento e dispor dessas informações para que a interface web possa consultá-los. Para armazenar os dados coletados, considerando a necessidade dos dados serem facilmente recuperados para a geração de gráficos, utilizou-se a ferramenta chamada RRDTool [7] [8]. A ferramenta apresentada neste trabalho se difere de ferramentas tradicionais de visualização de tráfego, baseadas em RRDTool, principalmente na forma de captura das informações, as quais são capturados pelo controlador OpenFlow.

2) *Serviço de monitoramento e coleta de dados*

Responsável por coletar os dados estatísticos dos *switches* e persistir em banco de dados. O serviço de

monitoramento é iniciado juntamente com o NOX e é executado como um componente do mesmo.

Para a coleta dos dados, o serviço de monitoramento utiliza o componente *switchstats* do NOX, responsável por coletar os dados estatísticos dos *switches*.

A descoberta dos *switches* que compõem a topologia é feita de forma automática, o componente *switchstats* fornece essa informação através da propriedade "*dp_stats*", que contém uma coleção de estatísticas e informações gerais de cada *switch* da rede. Esta propriedade é consultada para obter a lista de *switches* e assim poder criar as bases de dados para armazenar suas informações, através do método "*create*()" da ferramenta RRDTool. Em seguida, os dados estatísticos das interfaces e dos fluxos nos *switches* são adquiridos, através das propriedades "dp_port_stats", que contém uma coleção de estatísticas e informações das interfaces de cada *switch* da rede, e "dp_flow_stats", que contém uma coleção de estatísticas e informações dos fluxos de cada *switch* da rede. Esses dados são armazenados na base de dados através da chamada do método "*update*()", da ferramenta RRDTool, o qual ocorre em um intervalo de tempo pré-definido.

Os dados retornados dos *switches* estão disponíveis em informações sobre suas interfaces, também chamadas de portas, e informações sobre os fluxos configurados em cada *switch*.

As informações utilizadas para monitorar as interfaces foram:

- ***tx_bandwidth***: contém a largura de banda de transmissão utilizada;
- ***rx_bandwidth***: contém largura de banda de recebimento utilizada;
- ***tx_dropped_rate***: taxa de pacotes transmitidos que foram eliminados;
- ***rx_dropped_rate***: taxa de pacotes recebidos que foram eliminados;
- ***tx_errors***: quantidade de pacotes com erro transmitidos;
- ***rx_errors*** : quantidade de pacotes com erro recebidos.

Para monitorar os fluxos foram utilizadas:

- ***packet_count***: quantidade de pacotes que combinaram com o fluxo e aplicaram sua regra;
- ***byte_count***: quantidade de bytes que foram transmitidos por um fluxo.

3) Interface de consulta à base de dados

Interface utilizada pelo administrador da rede para consultar os dados históricos. Foi desenvolvido sobre plataforma web, podendo ser hospedado no próprio servidor onde roda o serviço de monitoramento e coleta de dados ou em um servidor separado.

A execução da interface web inicia quando seu endereço é requisitado ao serviço HTTP a partir de um navegador web, como apresentado na Figura 3. O serviço HTTP carrega o script correspondente, passando os parâmetros necessários. Esse, por sua vez, consulta a base de dados RRDTool fazendo a chamada ao método "*graph*()". O RRDTool gera o gráfico correspondente aos parâmetros que foram passados na chamada do método, retornando-o ao script que monta a página HTML a ser enviada ao navegador web pelo serviço HTTP.

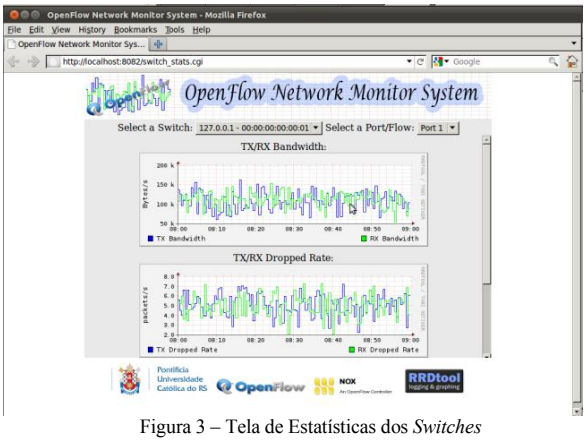

Figura 3 – Tela de Estatísticas dos *Switches*

V. OBJETIVOS ALCANÇADOS

O cenário utilizado para executar o OPNMS, apresentado na Figura 4, foi composto por dois *switches* (s1 e s2), cada um com duas interfaces (eth0 e eth1), sendo que os dois *switches* estão interligados por uma das suas interfaces (s1-eth1 conectado a s2-eth1). Conectados a estes *switches* tem-se dois *hosts* (h3 e h4). O *host* h3 conectado ao *switch* s1 (h3-eth0 conectado a s1-eth0) e o *host* h4 conectado ao *switch* s2 (h4-eth0 conectado a s2-eth0). Por fim, tem-se um controlador c0 conectado aos *switches* através de um canal seguro.

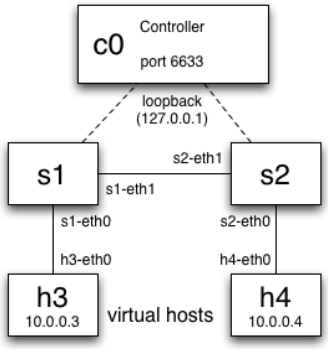

Figura 4 – Topologia utilizada para executar o OFNMS (imagem extraída de [4]).

Iniciando a ferramenta, a página inicial é carregada e disponibiliza um botão chamado "*Switches Stats*", concedendo acesso à página de estatísticas dos *switches*, apresentada na Figura 3. Nesta tela, os *switches* são mostrados em uma lista suspensa (*ComboBox*) em "*Select a Switch*". Ao selecionar um *switch*, outra lista suspensa é preenchida com as interfaces e fluxos do mesmo. Selecionando-se uma interface na lista suspensa "*Select a Port/Flow*", são apresentados os gráficos com informações sobre "*TX/RX BandWidth*", "*TX/RX Dropped Rate*" e "*TX/RX Errors*".

O gráfico da Figura 5 mostra a taxa de Largura de Banda, sendo que durante uma hora de tráfego foram transmitidos entre 50 e 150 KiloBytes por segundo. A linha azul indica a taxa de dados transferidos e a linha verde indica a taxa de dados recebidos pela interface.

O gráfico da Figura 6 mostra a taxa de pacotes descartados, sendo que durante uma hora de tráfego foram descartados entre 2 e 7 pacotes por segundo, pacotes estes que foram enviados corrompidos propositalmente para simular uma situação de descartes de pacotes. A linha azul indica a taxa de pacotes transferidos e a linha verde indica a taxa de pacotes recebidos pela interface.

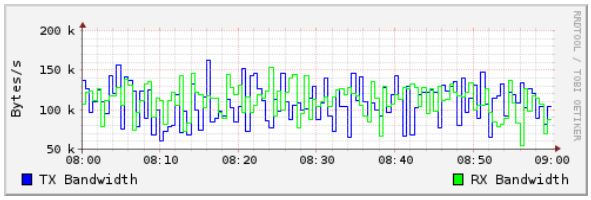

Figura 5 – TX/RX Port1 Bandwidth.

O gráfico da Figura 7 mostra a taxa de pacotes com erro, sendo que durante uma hora de tráfego foram contabilizados entre 2 e 7 pacotes por segundo. A linha azul indica a taxa de pacotes transferidos e a linha verde indica a taxa de pacotes recebidos pela interface.

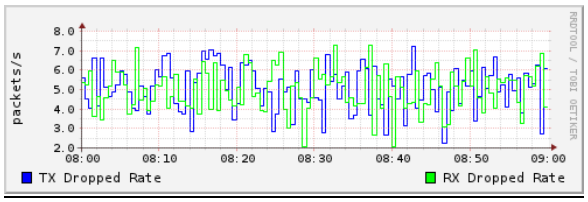

Figura 6 - TX/RX Port1 Dropped Rate Packets.

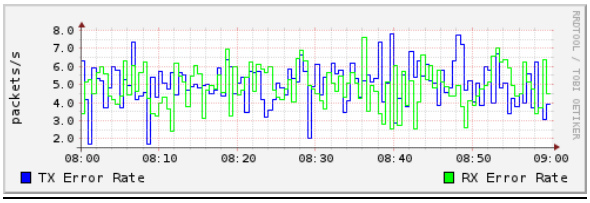

Figura 7 - TX/RX Port1 Error Rate Packets.

Selecionando-se um fluxo na lista suspensa "*Select a Port/Flow*", os gráficos com informações sobre "*Bandwidth*" e "*Packets Rate*" são carregados, apresentados na Figura 8 e Figura 9, respectivamente.

O gráfico da Figura 8 mostra a taxa da Largura de Banda utilizada pelos dados que foram encaminhados pelo fluxo, sendo que durante uma hora de tráfego foram contabilizados entre 60 e 150 KiloBytes por segundo.

O gráfico da Figura 9 mostra a taxa de pacotes encaminhados pelo fluxo, sendo que durante uma hora de tráfego foram contabilizados entre 800 a 1800 pacotes por segundo.

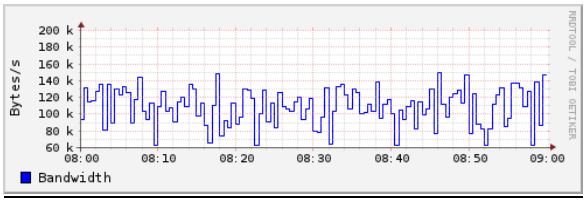

Figura 8 – Flow 1 Badnwidth.

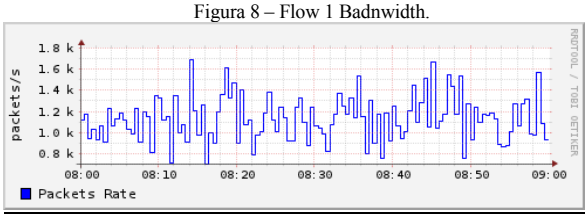

Figura 9 – Flow 1 Packets Rate.

VI. CONSIDERAÇÕES FINAIS

Este documento apresentou o desenvolvimento de uma ferramenta de monitoramento de *switches* OpenFlow. Tal ferramenta permite coletar dados estatísticos das interfaces e fluxos nos *switches* da rede, persiste esses dados em base de dados, possibilitando a sua visualização graficamente.

Como trabalhos futuros, pode-se desenvolver o monitoramento dos *hosts* e topologia da rede OpenFlow, além de incluir o período de coleta dedados.

REFERÊNCIAS

[1] GTA/UFRJ. "OMNI Openflow Management Infrastructure", Disponível em: <http://www.gta.ufrj.br/omni/>. Acesso em: 25 mai. 2012.

[2] McKeown, Nick. Anderson, Tom. Balakrishnan, Hari. Parulkar, Guru. Peterson, Larry. Rexford, Jennifer. Shenker, Scott. Turner, Jonathan "OpenFlow: Enabling Innovation in Campus Networks", ACM SIGCOMM'10, New York, NY, USA, Outubro 2010.

[3] OpenFlow Consortium. "OpenFlow Switch Specification Version 1.1", Disponívelem http://www.openflow.org/documents/openflow-specv1.1.0.pdf. Ultimo Acesso em: Set 2011.

[4] OpenFlow Website. Disponível em <http://www.openflow.org/ >. Acesso em: Set 2011.

[5] NOX Website. Disponível em: <http://noxrepo.org>. Acesso em: 05 set 2011.

[6] Gude, Natasha. Koponen, Teemu. Pettit, Justin. Pfaff, Ben. Casado, Martín. McKeown, Nick. Shenker.Scott "NOX: Towards an Operating System for Networks", ACM SIGCOMM'08, New York, NY, USA, Julho 2008.

[7] RRDTool Website. Disponível em: <http://oss.oetiker.ch/rrdtool/index.en.html>. Acesso em: 05 jun 2012.

[8] RRDTool Documentation. Disponível em <http://www.luteus.biz/Download/LoriotPro_Doc/RRD%20documentation/Introduction_RRD_EN.htm>. Acesso em 25 mai 2012.

[9] Nascimento, Marcelo R. Rothenberg, Chistian E., Denicol, Rodrigo R., Salvador, Marcos R., Magalhães, Maurício F.. "RouteFlow: Roteamento Commodity Sobre Redes Programáveis" XXIX Simpósio Brasileiro de Redes de Computadores. SBRC 2011, Campo Grande, MS, Brasil, Maio 2011.

[10] Nascimento, Marcelo R.; Rothenberg, Christian E.; Salvador, Marcos R.; Magalhães, Maurício F. "QuagFlow: Partnering Quagga with OpenFlow", ACM SIGCOMM'10, New Delhi, India, Agosto 2010.