Avaliação do suporte à simulação de redes OpenFlow no NS-3

Marcelo Conterato
PPGCC, PUCRS
Porto Alegre – Brasil
marcelo.conterato@acad.pucrs.br

Israel de Oliveira
PPGCC, PUCRS
Porto Alegre – Brasil
israel.oliveira@acad.pucrs.br

Tiago Ferreto
PPGCC, PUCRS
Porto Alegre – Brasil
tiago.ferreto@pucrs.br

César A. F. De Rose
PPGCC, PUCRS
Porto Alegre – Brasil
cesar.derose@pucrs.br

***Resumo*—A avaliação de ambientes simulados usando o protocolo *Openflow* está se tornando uma tendência, em virtude do aumento da demanda por soluções para redes definidas por software (SDN). O protocolo *OpenFlow* pode ser utilizado em diversos cenários como: segurança, roteamento, tolerância a falhas, desempenho, balanceamento de carga, entre outros. Por ser um protocolo experimental, existe uma demanda por ferramentas para avaliação do protocolo nestes cenários. Este artigo tem como objetivo discutir as vantagens e desvantagens do uso do simulador de redes NS-3 em conjunto com o protocolo *OpenFlow*. O artigo faz um levantamento das características das opções disponíveis em ambientes reais, emulados e simulados, levando em consideração o suporte parcial ao protocolo *OpenFlow* pelo NS-3.**

I. Introdução

As redes definidas por software (SDN – *Software Defined Networks*) [8] são caracterizadas pelo uso de um controlador para programar o funcionamento da rede de acordo com diferentes necessidades e propósitos, tendo por objetivo facilitar inovações estruturais na rede. As redes SDN fornecem abstrações basicamente em três áreas da rede: distribuição do fluxo de rede, encaminhamento e configuração. Além disso, as redes SDN facilitam a criação de políticas de encaminhamento de pacotes através do uso de fluxos, permitindo o uso de caminhos com características específicas, como por exemplo: maior largura de banda, menor latência ou menor número de saltos, contribuindo ainda, para uma redução no consumo de energia.

Para a implementação das redes SDN, dois requisitos devem ser plenamente atendidos: (i) deve existir uma arquitetura comum em todos os equipamentos de rede envolvidos na comunicação (e.g., *switches* e roteadores) ou qualquer outro dispositivo gerenciado por um controlador SDN e, (ii) deve existir um protocolo padronizado seguro para a comunicação entre o controlador SDN e os dispositivos de rede. Ambos os requisitos são satisfeitos pelo protocolo *OpenFlow* que tem como objetivos configurar os fluxos nos dispositivos de rede e gerenciar toda a infraestrutura definindo políticas de gestão do tráfego de rede.

Permitir que os usuários definam fluxos e determinem qual o caminho que esses fluxos devem tomar através da rede, sem interromper o tráfego normal, é um dos objetivos das pesquisas em Internet do Futuro. O termo Internet do Futuro é relativo a uma ampla iniciativa mundial para identificar os rumos tecnológicos que a rede deverá tomar nos próximos anos.

Atualmente, a demanda por ferramentas para avaliação de cenários de redes vem crescendo em virtude da necessidade de testar as soluções para redes antes mesmo da sua utilização no mundo real. Alguns fatores como: investimento, relação custo-benefício, complexidade de gerenciamento e tempo necessário para implementação são algumas das preocupações associadas aos ambientes de avaliação das redes atuais e das tecnologias de redes que estão surgindo. Por este motivo, o estudo dos ambientes possíveis para a execução de experimentos se torna importante no contexto atual das redes. A avaliação do funcionamento de cada ambiente, seja através de experimentos em um ambiente real, emulado ou simulado pode trazer para benefícios para os estudos de novas tecnologias SDN.

O NS (*Network Simulator*) é um simulador de redes bastante conhecido e popular na pesquisa de redes de computadores. O NS permite avaliar o funcionamento da rede em termos de tráfego e desempenho. A terceira versão do simulador (NS-3) possui suporte à simulação de redes *OpenFlow*.

Este artigo tem como objetivo avaliar as vantagens e desvantagens do uso do simulador de redes NS-3 em conjunto com o protocolo *OpenFlow*. O artigo apresenta também uma comparação das opções disponíveis para avaliação de redes *OpenFlow* em ambientes reais, emulados e simulados.

O artigo está organizado da seguinte forma: na Seção 2 é apresentado o funcionamento do protocolo *OpenFlow* e o levantamento das opções para avaliação do protocolo *OpenFlow* usando ambientes reais, emulados e simulados; a Seção 3 apresenta uma análise sobre a simulação do protocolo *OpenFlow* com o simulador NS-3; a Seção 4 apresenta uma comparação entre as estratégias de avaliação do protocolo *OpenFlow* usando ambientes reais, emulados ou simulados, e as conclusões do artigo são apresentadas na Seção 5.

II. Suporte para experimentação do protocolo OpenFlow

O protocolo *OpenFlow* [9] foi proposto pela Universidade de Stanford em 2008 e atualmente está na versão 1.3.1 [13]. A arquitetura de uma rede que utiliza o protocolo *OpenFlow* é composta por *switches* e controladores. O objetivo do controlador é, através de um canal seguro, modificar a tabela de fluxos de um *switch OpenFlow*. Quando o controlador recebe uma mensagem de um *switch*, ele examina o cabeçalho desta mensagem e verifica se um novo fluxo precisa ser criado ou qual ação precisa ser realizada. Se uma nova entrada precisa ser criada, o controlador envia uma mensagem ao *switch* para

instalar um novo fluxo. O controlador tem a capacidade de adicionar, atualizar e remover fluxos da tabela dos *switches OpenFlow*.

A Figura 1 apresenta uma arquitetura de rede usando o protocolo *OpenFlow*.

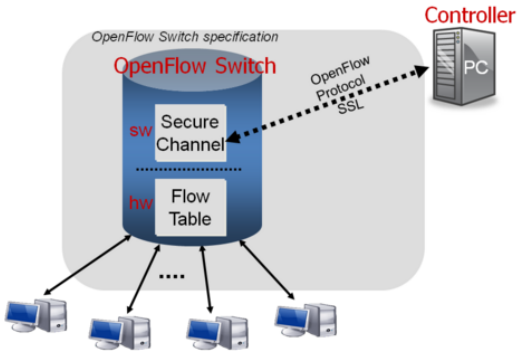

Figura 1. Arquitetura de rede usando o protocolo OpenFlow [9]

Em equipamentos de rede como roteadores e *switches* clássicos, o rápido encaminhamento de pacotes (plano de dados) e as decisões de encaminhamento (plano de controle) ocorrem em um mesmo dispositivo. Um *switch OpenFlow* tem a finalidade de separar estas duas funções. A função de encaminhamento dos dados permanece sendo função do *switch*, enquanto que, as decisões de encaminhamento de alto nível são funções agora de um controlador remoto, normalmente localizado em uma máquina servidora. O controlador e o *switch OpenFlow* realizam a comunicação através do protocolo *OpenFlow*, que define as mensagens, tais como: pacotes recebidos, encaminhamento dos pacotes de saída, modificação da tabela de encaminhamento, etc. A Figura 2 detalha o protocolo *OpenFlow*, apresentando a separação do plano de dados e do plano de controle.

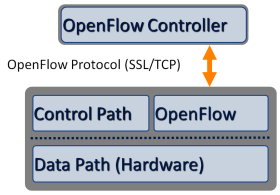

Figura 2. Separação dos planos de dados e controle no OpenFlow [10]

O *OpenFlow* permite o controle dos fluxos de dados, escolhendo o caminho que cada pacote segue e o processamento que irá receber. Desta forma, o *OpenFlow* permite experimentar novos protocolos de roteamento, mecanismos de segurança ou modelos de endereçamento na rede.

Atualmente, o protocolo *OpenFlow* pode ser avaliado usando um ambiente real, emulado ou simulado.

A. Ambientes Reais

Vários fabricantes já disponibilizam *switches* com suporte ao protocolo *OpenFlow* para criação de ambientes e realização de testes. Atualmente, também é possível alterar o *firmware* original de alguns *switches* por um *firmware open source* chamado Indigo [6]. O Indigo foi originalmente desenvolvido pela Universidade de Stanford e é uma implementação que suporta a especificação *OpenFlow* 1.0. A solução suporta até 48 portas 10-*gigabit ethernet*.

O projeto NetFPGA[1] (*Network Field Programmable Gate Array*) permite que pesquisadores, professores e estudantes realizem a prototipagem de dispositivos de rede (e.g. switches e roteadores) usando *hardware* programável. Em virtude de ter estas características, ele é amplamente utilizado nas pesquisas em Internet do Futuro com redes SDN. Por ser uma plataforma aberta, pesquisadores têm utilizado o NetFPGA para construir sistemas de rede avançadas para processamento de fluxos. O sistema consiste em uma placa programável que pode rotear pacotes de quatro sub-redes. Várias placas NetFPGAs podem ser instaladas em uma mesma máquina. Atualmente, existem duas plataformas: NetFPGA-1G (1G) que fornece 4 portas *gigabit ethernet* e a NetFPGA-10G (10G) que fornece 4 portas *10-gigabit ethernet*.

Podemos citar ainda, como solução para uso em ambientes reais, a utilização do *OpenWRT* [2]. Esta solução consiste em um sistema operacional *Linux* utilizado principalmente para dispositivos embarcados, onde é possível personalizar o *firmware* de alguns modelos específicos de dispositivos, habilitando o suporte a utilização do protocolo *OpenFlow*. Atualmente o OpenWRT encontra-se na versão 12.09 [14].

B. Ambientes Emulados

O MiniNet [2] é uma ferramenta para a simulação de redes definidas por software que permite a emulação de uma grande infraestrutura virtual de rede com a utilização de apenas uma máquina. O Mininet possibilita a criação de ambientes escaláveis usando redes virtuais utilizando primitivas de virtualização do sistema operacional. Com essas primitivas, o usuário pode rapidamente criar, interagir, personalizar e compartilhar um protótipo de rede definida por *software* (SDN) para simular uma topologia de rede que utiliza *switches OpenFlow*.

O MiniNet permite desenvolver topologias personalizadas utilizando *scripts* em *Python*, com a possibilidade de interação com a rede física existente utilizando o Mininet CLI (juntamente com a API disponível), sendo ainda, possível a sua implantação em *hardware* real. O MiniNet utiliza controladores reais, como NOX [3], POX[3] e o *FloodLigth*[4], para o gerenciamento das tabelas de fluxos.

C. Ambientes Simulados

Atualmente, a simulação tem um papel decisivo no projeto, análise e implementação de sistemas de comunicação, principalmente quando estes sistemas são caros e complexos. A simulação de um sistema real pode ser definida como o processo de avaliação numérica de um modelo de simulação, que deve representar o mais fielmente possível o sistema real a ser simulado. As informações resultantes deste processo são utilizadas para estimar variáveis de interesse deste sistema.

[1] http://www.netfpga.org/
[2] http://mininet.org/
[3] http://www.noxrepo.org/pox/about-pox/
[4] http://floodlight.openflowhub.org/

A utilização de ambientes de simulação vem aumentando de forma significativa uma vez que estes permitem o estudo e a avaliação de sistemas a custos reduzidos. Os simuladores de rede desempenham um papel importante na tarefa de desenvolver, analisar e aperfeiçoar protocolos de comunicação. Destacam-se três importantes vantagens do uso de simulação [4]:

1) Permitem testar o comportamento dos protocolos em diversas redes e ambientes, cuja preparação num laboratório ou em uma empresa poderia ser impraticável, isso por questão de custos, ou em tempo de instalação, ou mesmo do ponto de vista administrativo;
2) Facilitam a execução de testes em um ambiente controlado, onde é mais fácil fazer variar parâmetros relevantes, mantendo os restantes parâmetros constantes;
3) Facilitam a execução dos protocolos em múltiplos cenários de execução.

Segundo [5], alguns dos principais motivos pelos quais é recomendável que os ambientes de redes sejam previamente simulados antes da sua instalação são:

- A experimentação no mundo real pode causar alguma espécie de dano ao ambiente;
- A modelagem e análise requerem altos níveis de abstração e possíveis falhas podem surgir na especificação de detalhes;
- Alterações em redes existente podem requerer uma carga de trabalho expressivo;
- Falhas na configuração de determinadas tecnológicas podem gerar um custo elevado.

Existem diversos simuladores de rede disponíveis, porém poucos são os simuladores que oferecem suporte ao protocolo *OpenFlow*. Um destes simuladores é o *Network Simulator* 3 (NS-3). A Seção 3 apresenta em detalhes o suporte à simulação do protocolo *OpenFlow* no NS-3.

III. Simulação do protocolo OpenFlow no NS-3

O NS-3 é um simulador de rede de eventos discretos utilizado principalmente por pesquisadores, principalmente por possuir distribuição gratuita e código aberto [12]. Tal fato o torna adequado a situações em que é necessário desenvolver novas funcionalidades, como em teses e projetos de pesquisa aplicada. A execução de experimentos de simulação no NS-3 exige a implementação de códigos em C++ ou *Python*.

O NS-3 atualmente está na versão 3.18, tendo seu desenvolvimento sido iniciado em julho de 2008. Ao contrário do que sugere a versão do simulador, o NS-3 é um simulador totalmente novo, e não uma extensão da versão anterior do simulador. O NS-3 não suporta as APIs do NS-2, porém algumas funcionalidades do NS-2 foram portadas para o NS-3.

O simulador de redes NS-3 é considerado uma das ferramentas de simulação que apresenta melhor desempenho [16]. O NS-3 tem sua arquitetura baseada na técnica de rastreamento [1]. O rastreamento facilita a descoberta de eventos significativos na simulação e permite que o pesquisador obtenha métricas importantes de uma simulação que podem ser utilizadas para comparação entre diferentes cenários.

Atualmente, a implementação do protocolo *OpenFlow* existente no NS-3 disponibiliza dois controladores, que executam funções básicas, permitindo apenas que determinados fluxos de dados ou sejam descartados pelo *switch*, ou ainda, que seja realizado o processo de aprendizagem tradicional dos switches, onde é cada máquina detectada é mapeada para uma porta específica do switch. Os controladores implementados no NS-3 são:

- *DropController*: Quando um *switch* encaminha para o controlador um pacote que ele não sabe o que fazer, o controlador retorna ao *switch* a regra para descartar os pacotes;
- *LearningController*: Quando um *switch* encaminha para o controlador um pacote que ele não sabe o que fazer, o controlador verifica em uma tabela interna se existe alguma porta específica mapeada para o destinatário do pacote. Se existir, ele configura o switch para enviar o pacote para a porta específica, caso contrário, ele indica para o switch enviar o pacote para todas as suas portas (*flooding). Logo após, ele obtém a identificação da máquina e porta origem do pacote e envia uma regra ao switch mapeando o endereço da máquina para a porta específica.*

Por vários aspectos o NS-3 não simula de forma fidedigna uma rede *OpenFlow* real. O módulo *OpenFlow* apresenta as seguintes limitações:

- não tem suporte ao tráfego TCP entre o *switch* e o controlador;
- não tem suporte ao protocolo *Spanning Tree* e *Multi Protocol Label Switching* (MPLS);
- não possibilita a utilização de controladores externos;
- a implementação do *switch* não suporta remontagem do pacote IP.

O NS-3 possibilita a captura do tráfego gerado durante a simulação para fins de análise dos dados. O NS-3 utiliza o formato pcap para arquivos de captura, utilizado também pela ferramenta TCPDump [7]. A ferramenta Wireshark [15] também pode ser utilizada para visualização dos fluxos, exigindo previamente a configuração do plugin *OpenFlow Wireshark Dissector* [5]. Através da ferramenta NetAnim [11] é possível visualizar graficamente simulações armazenadas em arquivos de rastreamento especiais armazenados em formato XML.

IV. Comparação entre as soluções para avaliação do protocolo OpenFlow

Esta seção apresenta uma comparação das principais soluções para avaliação do protocolo *OpenFlow* usando ambientes reais, emulados e simulados. A Tabela I resume as principais funções dos ambientes vistos anteriormente.

Um dos fatores que mais chamam a atenção é a escalabilidade, onde o Mininet foi classificado como de escalabilidade

[5] http://goo.gl/IdLrk5

Tabela I. RECURSOS DISPONÍVEIS

	Switches OF / NetFPGA	Mininet	NS-3
Especificação OpenFlow	1.3.1	1.3	0.8.9
Tipo de Ambiente	Real	Emulação	Simulação
Controlador no mundo real	X	X	-
Escalabilidade	Alta	Média	Alta
Suporte GUI	Configuração Observação	Observação Config C++	Observação Config Phyton

média em virtude de necessitar executar um processo *shell* (por exemplo /bin/bash) para conseguir emular cada nodo, sendo ainda necessário, executar um processo para cada *vSwitch* no espaço do usuário para simular cada *switch OpenFlow*. Como resultado, o Mininet é menos escalável que NS-3, pois em virtude do nodos, dos *switches OpenFlow* e dos controladores serem todos simulados no NS-3, eles estão todos implementados como módulos C++ e conectados entre si como um único processo. Os ambientes reais também são classificados como de escalabilidade alta, pois sua única limitação é o custo de aquisição dos equipamentos.

O MiniNet até o momento ainda não fornece desempenho e qualidade fiéis a uma rede real, embora o código utilizado nele sirva para uma rede real baseada em placas NetFPGAs, ou *switches* comerciais. Isso se deve aos recursos que são tratados pelo *kernel* da máquina simuladora em tempo real, uma vez que a largura de banda total é limitada por restrições de CPU e memória da mesma.

Já o NS-3 apresenta uma limitação relevante que é a inexistência de uma comunicação TCP entre o *switch OpenFlow* e o controlador. Atualmente somente a comunicação UDP é suportada. Em cenários reais, a falta de controle do congestionamento que é realizado pelo protocolo TCP pode afetar o desempenho da rede ocasionando perda de pacotes, afetando assim a comunicação entre o controlador e o *switch*. Outra limitação existente no NS-3 é a manipulação da tabela de fluxos pelos usuários finais. Os recursos de adição e modificação de fluxos estão disponíveis no NS-3, porém sua utilização requer o conhecimento da implementação do protocolo *OpenFlow* através da manipulação da estrutura ofp_flow_mod onde estão localizados os parâmetros responsáveis pela tomada de decisão dos fluxos.

V. CONCLUSÃO

Neste artigo foi apresentado um levantamento das vantagens e desvantagens do uso do simulador de redes NS-3 para estudo do protocolo *OpenFlow*. Foi realizado um levantamento das opções disponíveis em ambientes reais, emulados e simulados. O uso de simuladores, como o NS-3, torna-se mais interessante por sua flexibilidade quanto a programação, permitindo ao usuário a criação de seu próprio controlador *OpenFlow* e de uma rotina de testes, agilizando a execução das simulações.

Devido a dificuldade de demonstrar experimentalmente seu comportamento em condições próximas das reais, essas novas tecnologias são de alto custo, tornando as soluções de emulação e simulação atrativas. Dessa forma, a simulação do protocolo *OpenFlow* no NS-3 ainda necessita evoluir em vários aspectos, onde destacamos a conexão com controladores *OpenFlow* externos, não permitindo assim, quantificar os efeitos de simular múltiplos *switches* conectados a um único controlador *OpenFlow*.

REFERÊNCIAS

[1] Gustavo Carneiro, Pedro Fortuna, and Manuel Ricardo. Flowmonitor: a network monitoring framework for the network simulator 3 (ns-3). In *Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools*, page 1. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009.

[2] Florian Fainelli. The openwrt embedded development framework. In *Proceedings of the Free and Open Source Software Developers European Meeting*, 2008.

[3] Natasha Gude, Teemu Koponen, Justin Pettit, Ben Pfaff, Martín Casado, Nick McKeown, and Scott Shenker. Nox: towards an operating system for networks. *ACM SIGCOMM Computer Communication Review*, 38(3):105–110, 2008.

[4] Conceição V. Nuno C. Rodrigues L. Guedes, S. Plataforma de desenvolvimento e simulação de protocolos. 2005.

[5] J. Hughes. Network simulation introduction. Website OpenXtra, 2009. http://www.openxtra.co.uk/articles/network-simulation.

[6] Indigo. Indigo project. Website, 2013. http://www.openflowhub.org/display/Indigo/Indigo+-+Open+Source+OpenFlow+Switches+-+First+Generation/.

[7] Van Jacobson, Craig Leres, and S McCanne. The tcpdump manual page. *Lawrence Berkeley Laboratory, Berkeley, CA*, 1989.

[8] Bob Lantz, Brandon Heller, and Nick McKeown. A network in a laptop: rapid prototyping for software-defined networks. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, page 19. ACM, 2010.

[9] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74, 2008.

[10] Nick McKeown and Guru Parulkar. State of openflow and sdn, 2010.

[11] Development NS-3-Team. Netanim - offline animator. Website, 2013. http://www.nsnam.org/wiki/index.php/NetAnim.

[12] Development NS-3-Team. Ns-3 tutorial. Website, 2013. http://http://www.nsnam.org.

[13] OpenFlow. Openflow switch specification. Website, 2011. http://www.openflow.org/documents/openflow-wp-latest.pdf.

[14] OpenWRT. Openwrt project. Website, 2013. https://openwrt.org/.

[15] Angela Orebaugh, Gilbert Ramirez, and Jay Beale. *Wireshark & Ethereal network protocol analyzer toolkit*. Syngress, 2006.

[16] Elias Weingartner, Hendrik Vom Lehn, and Klaus Wehrle. A performance comparison of recent network simulators. In *Communications, 2009. ICC'09. IEEE International Conference on*, pages 1–5. IEEE, 2009.