

Simulando Cenários para Redes Definidas por Software

Gabriel Marchesan¹, Roseclea Duarte Medina¹

Universidade Federal de Santa Maria (UFSM)

Avenida Roraima, 1000 – 97.105-900 – Santa Maria – RS – Brasil

¹Grupo de Redes de Computadores e Computação Aplicada (GRECA)

{gmarchesan, rose}@inf.ufsm.br

Abstract. Nowadays the processes which involve the implementation and deployment of the computer networks are limited by the choices of standardized equipments from renowned manufacturers who in the most times use proprietary softwares. To overcome these problems, the Software-Defined Network (SDN) architecture was created, which makes it possible to the networks to be customized according to the capability and requirement of the environment. In this paper, it will be shown, in a general way, the main concepts of the Software Defined Network by simulating two use case scenarios with the SDN MiniNet tool.

Resumo. Atualmente, os processos que envolvem a implementação e implantação das redes de computadores são limitados às escolhas de equipamentos padronizados de fabricantes renomados, que na maioria das vezes fazem uso de softwares proprietários. Para superar estes problemas, surgiu a arquitetura Software-Defined Network (SDN), fazendo com que as redes sejam customizadas de acordo com a capacidade e necessidade do ambiente. Neste trabalho, será discutido, de uma maneira geral, os principais conceitos sobre Redes Definidas por Software, simulando dois cenários de uso com a ferramenta SDN MiniNet.

1. Introdução

Para mitigar alguns dos problemas que existem nas redes atuais, como por exemplo, limitação dos equipamentos padronizados que fazem uso de *softwares* proprietários, grande demanda das tabelas de roteamento, complexidade de diversos protocolos, grande número de aplicações executadas pelos usuários gerando gargalos na rede, surge um novo conceito de arquitetura de rede, chamado de *Software-Defined Network* (SDN) ou Redes Definidas por Software.

As SDNs constituem um novo paradigma para o desenvolvimento de pesquisas em redes de computadores. Este paradigma vem adquirindo espaço e grande atenção de parte da academia e das grandes indústrias da área de redes de computadores [GUEDES et al., 2012].

As SDNs vão permitir que os programadores inventem novos métodos de implantação de redes adequadas às diferentes necessidades dos usuários para otimizar o desempenho de suas aplicações e conseqüentemente suas tarefas. Esse novo paradigma sugere o desacoplamento entre o plano de dados (*hardware* especializado) e

plano de controle (executado em um ou mais servidores, os quais são responsáveis pela programação das ações realizadas pelo *hardware*) [GUEDES et al., 2012]. Através desta separação, e utilizando-se o protocolo *OpenFlow*, será permitido que os administradores possam definir fluxos de dados e determinar quais caminhos os mesmos devem percorrer, por exemplo, quando um *switch* ou roteador recebe um pacote, ao invés de tomar a decisão sozinho, o pacote será enviado ao controlador, que utilizará critérios que foram estabelecidos para tomar a decisão, criando dessa maneira as regras de encaminhamento do fluxo de dados [COSTA, 2013].

Como as redes de computadores se tornaram uma infraestrutura crítica na nossa sociedade globalizada, fazer experimentos em redes reais torna-se impraticável, pois além do alto valor agregado dos equipamentos, os riscos de ocorrer possíveis problemas seriam grandes e causariam transtornos no dia a dia nos diversos serviços que dependem do uso da rede. Dessa maneira, com o plano de controle da rede à disposição dos estudantes e pesquisadores da área, os mesmos podem aplicar suas novas ideias, em simuladores de redes, sem violar as regras existentes de roteamento e sem causar problemas nas redes em funcionamento [GUEDES et al., 2012].

Considerando o que foi exposto acima, o objetivo deste trabalho é explorar as funcionalidades e os recursos oferecidas pelo simulador SDN MiniNet, dessa forma possibilitando que professores e alunos possam utilizar o mesmo para realizar o estudo e a prática das SDNs através de testes e simulações sem interferir na rede em produção. Ademais, também tem como objetivo através do uso das simulações, deixar mais claro o entendimento teórico e o comportamento prático dos dispositivos simulados, dessa forma facilitando a interpretação e a abstração de informações aprendidas em ambientes acadêmicos, conseqüentemente agregando novas competências e assim contribuindo para um melhor ensino-aprendizagem da área de redes de computadores.

A próxima seção apresenta uma abordagem geral sobre SDN, também citando o controlador utilizado no desenvolvimento deste trabalho. A seção 3, descreve o simulador utilizado para a realização dos cenários, citando outros dois simuladores que também implementam o protocolo *OpenFlow*, o qual permite a implementação das SDNs. A seção 4, descreve as principais características do simulador MiniNet. A seção 5, descreve os cenários que foram simulados no trabalho. Por fim, a seção 6 descreve as considerações finais e algumas sugestões para trabalhos futuros.

2. Redes Definidas por Software

Apesar da grande expansão da Internet, em termos de quantidade de dados trafegados, de usuários conectados, de penetração e uma vasta gama de aplicações, não observou-se significativamente uma evolução em sua arquitetura nos últimos anos. Algumas modificações que já foram realizadas na arquitetura da Internet, não estão sendo suficientes para atender as demandas de novas aplicações que vem sendo inseridas todos dias na Rede.

Ao longo dos anos, a Internet tornou-se comercial, e os equipamentos de rede tornaram-se “caixas pretas”, ou seja, implementações integradas verticalmente baseadas em *software* fechado sobre *hardware* proprietário [COSTA, 2013]. Além disso, as redes vem se tornando parte da infraestrutura crítica de diversos ambientes, pois sua utilização é essencial em trabalhos empresariais, comerciais, domésticos, acadêmicos, entre outros.

Com todos esses problemas surgindo, muitos pesquisadores afirmam que a arquitetura de redes de computadores em geral e a rede mundial (a Internet) atingiram um nível de

amadurecimento que as tornaram pouco flexíveis [GUEDES et al., 2012]. O resultado desse modelo é o já reconhecido engessamento da Internet [CHOWDHURY; BOUTABA, 2009]. Dentre estas novas propostas, a Universidade de Stanford e a Universidade da Califórnia em Berkeley, propuseram esse novo paradigma de rede, denominado SDN, como um novo conceito aberto a toda a comunidade de pesquisadores de tecnologias e soluções de redes [MCKEOWN et al., 2008].

Esse novo paradigma, apresentado na figura 1, sugere a separação entre o plano de dados (também chamado de plano de encaminhamento) responsável pelo encaminhamento dos pacotes com base em regras, e o plano de controle, responsável pelo controle da rede em geral, permitindo ao controlador o gerenciamento das entradas da tabela de encaminhamento e das regras associadas ao tráfego desejado, dessa maneira proporcionando uma melhor visão global sobre toda a rede e ficando a cargo dele toda a inteligência da rede.

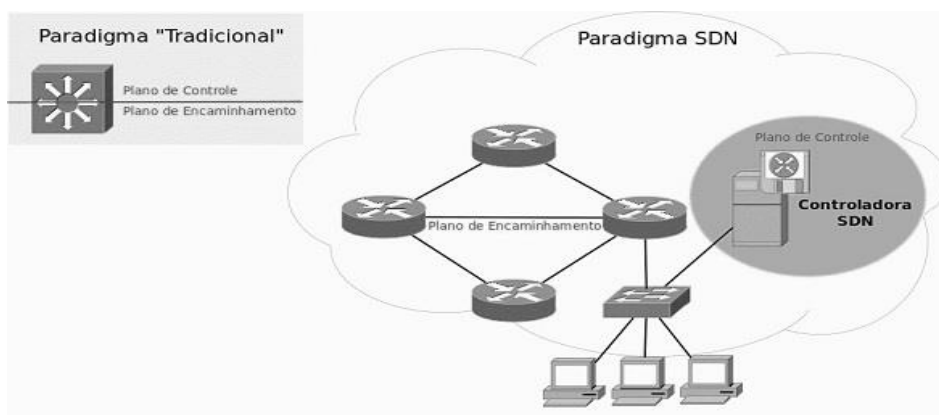


Figura 1 – Paradigma SDN (BRITO, 2013)

Para estabelecer a comunicação entre o plano de controle e o plano de encaminhamento foi preciso criar e padronizar uma API, sendo então estabelecido o protocolo *OpenFlow*, um protocolo aberto que possibilita o desenvolvimento de mecanismos programáveis baseado em tabelas de fluxos em diferentes *switches* e roteadores [MCKEOWN et al.,2008], ele estabelece uma comunicação segura entre os *switches* e o controlador, o qual utiliza esse canal para monitorar e estabelecer fluxos conforme a inteligência estabelecida pelo *software* [MOZZAQUATRO et al., 2013].

Para a realização deste trabalho foi utilizado o controlador POX, o mesmo possui uma boa curva de aprendizado, e além disso, foi especialmente projetado para o desenvolvimento de *softwares*, componentes, contribuindo muito em pesquisas e no ensino do protocolo *OpenFlow*, portanto sendo escolhido para utilização do presente trabalho [HERVÁS, 2014]. Ademais, o POX disponibiliza uma interface mais simples e uma pilha SDN mais organizada, facilitando nas pesquisas e nos estudos desta área [COSTA, 2013].

É importante ressaltar que SDN é uma abstração de mais alto nível do que o protocolo *OpenFlow*, pois esse é uma tecnologia recente e que pode ser usada para viabilizar a implementação das SDNs.

3. Simuladores SDN

Como já citado anteriormente neste trabalho, as redes de computadores se tornaram uma infraestrutura crítica na sociedade globalizada. Nesta situação, dificilmente pode-se fazer experimentos em redes reais, pois além dos possíveis riscos que podem ser gerados, é impraticável a construção de laboratórios e experimentos reais devido ao alto valor dos equipamentos.

Com isso pode-se optar em fazer os testes e experimentos através de simulações em simuladores de redes, assim, obtém-se uma modelagem de um ambiente mais próximo do mundo real, dessa forma, possibilitando a avaliação de mais cenários, a um custo e tempo razoavelmente pequeno se comparado com testes em ambientes físicos, além disso, os simuladores são ferramentas essenciais para o estudo e evolução da computação, especialmente na área de redes de computadores [DOURADO; FILHO; MARQUES, 2011].

Em relação ao mundo acadêmico, por meio dos simuladores, professores podem complementar o ensino de diversos temas e conteúdos mais técnicos na área de redes, ajudando aos alunos na elucidação de conceitos abstratos e difíceis de se entender, assim, agregando no aprendizado do aluno e instigando os mesmos a participarem mais das aulas, dessa forma, melhorando como um todo o processo de ensino-aprendizagem.

Neste trabalho foi utilizado o simulador MiniNet para a realização das simulações, o qual permite uma rápida prototipação de uma grande infraestrutura virtual de rede, executando em um *kernel* real, com a utilização de apenas um computador [MININET, 2015]. Além disso, o MiniNet cria uma rede virtual *OpenFlow* com um controlador, *switches*, *hosts* e *links*, também permite desenvolver topologias personalizadas utilizando *scripts* em Python [CONTERATO et al., 2013].

É importante ressaltar que além do MiniNet, existem outros simuladores SDN que implementam o protocolo *OpenFlow*, como por exemplo, o NS-3 (*Network Simulator 3*) e o EstiNet. O NS-3 disponibiliza dois controladores, que executam funções básicas, permitindo que determinados fluxos de dados ou sejam descartados pelo *switch*, ou realizem o processo de aprendizagem tradicional dos *switches*, onde cada máquina é detectada e mapeada para uma porta específica do *switch* [CONTERATO et al., 2013].

Ademais, o NS-3 também possui algumas limitações didáticas e técnicas, como por exemplo, uma curva de aprendizado difícil, um módulo *OpenFlow* que não simula de forma tão fidedigna uma rede *OpenFlow* real, não possuindo suporte ao tráfego TCP entre o *switch* e o controlador e ao *Spanning Tree Protocol* (STP), entre outras limitações. Já o EstiNet é um simulador proprietário, podendo ser utilizado para a realização de simulações com vários controladores, também possibilita o modo de simulação quanto o de emulação. O EstiNet apresenta como algumas desvantagens, o fato de ainda não existir muito material disponível na Internet para estudo e pesquisa, ainda ser pouco conhecido e utilizado para simulações em redes em geral, em especial para SDNs, possuir uma curva de aprendizado de nível médio para difícil, e principalmente por ser um *software* proprietário, o que inviabiliza um trabalho acadêmico.

4. Cenários

Para a realização dos cenários de simulação, foi utilizado alguns dos módulos/componentes fornecidos juntamente quando se instala o controlador POX, sendo recomendado instalar o mesmo na própria máquina virtual VM MiniNet. É importante ressaltar que em algumas versões mais recentes da VM MiniNet o controlador POX já encontra-se instalado, já nas mais antigas, é preciso instalar o

mesmo. Para este trabalho, foi realizada as simulações do cenário *Switch* e de um cenário que possui *loops* em sua topologia de rede, sendo eliminado os *loops* através do uso do STP (*Spanning Tree Protocol*).

4.1 Switch

Neste cenário foi simulado o comportamento do dispositivo *switch* em SDN. O *switch* ou “comutador”, é um dispositivo que opera na camada 2 (enlace de dados) do modelo de referência OSI. Ao contrário dos *Hubs*, os *switches* registram na tabela de repasse os endereços MAC dos dispositivos que estão ligados a cada porta do equipamento. Então o *switch* encaminha os pacotes de acordo com o endereço MAC de destino, sendo considerado um dispositivo “inteligente” que tem a função de aprender com a rede e depois apenas encaminhar o pacote para à máquina de destino específico.

Para a implementação deste cenário foi utilizado o módulo “*forwarding.l2_learning*”. Para a realização deste cenário, foi utilizada uma topologia de rede que consiste de três *hosts* ligados a um *switch OpenFlow*, e este a um controlador *OpenFlow*. Para criar essa topologia, em um terminal SSH, basta apenas executar o comando:

- “*sudo mn --topo single,3 --mac --switch ovsk --controller remote*”

Após isso, para abrir três janelas separadas, uma para cada host de modo que facilite a visualização e o gerenciamento dos *hosts*, basta executar o comando “*xterm h1 h2 h3*”.

Após isso, em outro terminal SSH, para poder ser inicializado o componente “*forwarding.l2_learning*”, é preciso apenas acessar o diretório do “*pox*” e executar o seguinte comando:

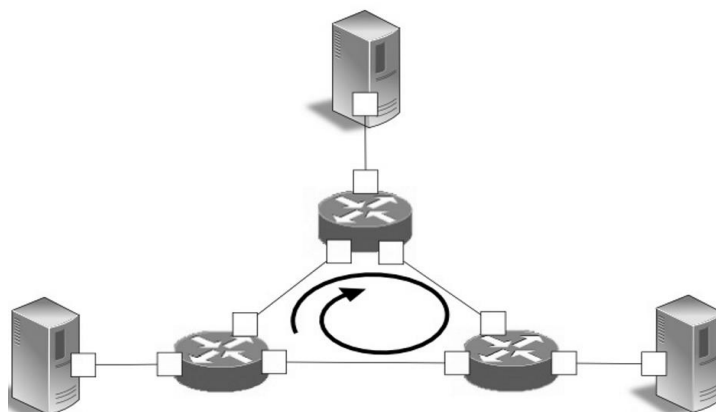
- “*./pox.py log.level --DEBUG forwarding.l2_learning*”

Prosseguindo o cenário, é executado um “*ping*” do h2 para o h3 (neste trabalho utilizado com o endereço IP 10.0.0.3), após pingar um pacote do h2 para h3, através de alguma ferramenta de captura de pacotes (por exemplo, “*tcpdump*” ou “*wireshark*” que já vem instalado nas versões mais recentes da VM MiniNet) é possível observar que não há fluxos de pacotes trafegando no h1. Portanto, como na outra máquina (h1) da rede não está trafegando pacotes, e sim somente no destino específico (h3), assim, pode-se observar que o componente está se comportando como um *switch*.

Através dessa simulação com a utilização das ferramentas de capturas de pacotes, fica muito mais fácil de entender o funcionamento do dispositivo *switch*, pois é possível visualizar de forma prática como é o comportamento do mesmo nas redes de computadores, elucidando os conceitos teóricos, como por exemplo, o uso do protocolo ARP para realizar a descoberta dos endereços MAC dos *hosts* da rede.

4.2 STP (*Spanning Tree Protocol*)

Neste cenário, será simulado uma topologia de rede que contém *loops*, isto é, uma topologia que forma um anel/ciclo fechado, dessa forma a rede não consegue achar as portas para realizar o encaminhamento dos pacotes. Para solucionar esse problema, é utilizado o *Spanning Tree Protocol* (STP). Para realizar esse cenário, a figura 2 mostra a topologia utilizada. Essa topologia é constituída de três *switches* e três *hosts*, sendo que cada *host* está ligado a um *switch*, formando um ciclo fechado, consequentemente gerando *loop* na rede.



**Figura 2 – Topologia ocorrendo loop na rede
(Autoria própria)**

Neste trabalho não será explicado como foi implementado o *script*, utilizando-se a linguagem Python, da topologia que foi exposta anteriormente. Após fazer o *script*, é preciso nomear ele (nesse exemplo, nomeou-se como “topo_loop.py”), salvar o mesmo no diretório “/mininet/custom” e então para executar o *script* e gerar a topologia, basta executar:

- `sudo mn --custom topo_loop.py --topo mytopo`

Para fazer um teste, é executado o comando “*pingall*” para ver se os *hosts* irão comunicar-se, é possível então observar que os *hosts* não comunicam-se, evidenciando assim que a topologia realmente apresenta *loop* na rede. Para solucionar o problema de *loop* presente na rede, em outro terminal SSH, entre no diretório do “*pox*” e simplesmente execute o comando a seguir.

- `./pox.py openflow.spanning_tree --no-flood --hold-down openflow.discovery forwarding.l2_pairs`

Agora será preciso mais uma vez executar o *script* e gerar novamente a topologia citada anteriormente, após isso, pode-se executar novamente o comando “*pingall*” e é possível observar que agora os *hosts* conseguem se comunicar. Portanto, pode-se constatar que realmente funciona o STP, eliminando o *loop* da rede.

Neste cenário foi utilizado uma topologia mais simples, mas é importante ressaltar que com uma topologia mais complexa, como por exemplo com 8 *switches* e 16 *hosts* foi realizado a simulação e o STP também funcionou perfeitamente. Percebe-se que a grande vantagem da utilização da SDN para esse caso, é que o controlador POX vai ser o elemento responsável da rede, então é preciso somente implementar e executar a aplicação do STP no controlador, que dessa forma o STP já será aplicado em todos os equipamentos da rede, assim não sendo preciso por exemplo, fazer a configuração manualmente do STP em cada equipamento da rede, facilitando o trabalho do administrador de redes.

6. Considerações Finais

O paradigma SDN é um assunto bastante recente e começou a ser explorado a pouco tempo. Mesmo assim, levantou grande interesse na área devido às suas capacidades de aumentar significativamente a flexibilidade da estrutura da rede, a fim de fornecer serviços com mais facilidade, ajustando-se às características da aplicação. Como prova

do interesse de especialistas nas SDNs, está o fato de já existirem no mercado os primeiros dispositivos comerciais baseados no protocolo *OpenFlow*.

Através deste trabalho buscou-se apresentar inicialmente um breve contexto das redes de computadores, ficando evidenciado que a atual arquitetura das Redes é pouco flexível, pois mesmo que tenha sofrido algumas modificações nos últimos anos, estas não foram suficientes para atender as demandas de novas aplicações que cada vez mais vem sendo inseridas no contexto tecnológico.

Ainda, o presente trabalho buscou apresentar os principais conceitos sobre Redes Definidas por Software, como por exemplo, os vários aspectos que envolvem o protocolo *OpenFlow*, o controlador da rede, sendo utilizado o POX, o qual foi especialmente desenvolvido para o ensino e pesquisa das SDNs. Por fim, através do simulador SDN MiniNet juntamente com a utilização dos recursos oferecidos pelo controlador POX, foi realizada a implementação e a simulação dos cenários *Switch*, e do cenário que possui *loop* em sua topologia, sendo eliminado o mesmo através do uso do protocolo STP.

Em suma, apesar do paradigma ser recente, considerando o presente sucesso das SDNs, acredita-se que esta nova tecnologia realmente irá trazer muitos avanços para o desenvolvimento de novos serviços, aplicações e soluções para a área de redes de computadores como um todo.

Por fim, como sugestão de trabalhos futuros, pode-se citar o desenvolvimento e a implementação de mais cenários, a integração com Ambientes Virtuais de Aprendizagem, a realização de testes em equipamentos físicos em uma rede real sem afetar o tráfego em produção, a utilização e o estudo dessa nova tecnologia em cursos de redes de computadores, expandindo o paradigma SDN, dessa maneira agregando novos conhecimentos, contribuindo tanto na formação técnica quanto na científica de seus entusiastas.

7. Referências

- Brito, S. H. B. (2013) “Blog LabCisco: paradigma sdn de redes programáveis”. Disponível em <<http://labcisco.blogspot.com.br/2013/07/paradigma-sdn-de-redes-programaveis.html>>. Último acesso em: 24 jun. 2015.
- Chowdhury, N.; Boutaba, R. (2009) “Network virtualization: state of the art and research challenges”. *IEEE Communications Magazine*, v.47, n.7, p.20–26.
- Conterato, M.; Oliveira, I.; Ferreto, T.; Rose, César A. F. (2013) “Avaliação do suporte à simulação de redes OpenFlow no NS-3”, p.3–6.
- Costa, L. R. (2013) “OpenFlow e o Paradigma de Redes Definidas por Software”. Monografia de Graduação – Instituto de Ciências Exatas – Departamento de Ciência da Computação - Universidade de Brasília.
- Dourado, G. G. M.; Filho, G. P. R.; Marques, M. (2011) “A Importância da Simulação e o uso do Network Simulator 3 na Pesquisa Científica”.
- Guedes, D., Vieira, L.F.M., Vieira, M.M., Rodrigues, H. e Nunes, R.V. (2012) “Redes Definidas por Software: uma abordagem sistêmica para o desenvolvimento das pesquisas em Redes de Computadores”. XXX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC, p.160–210.

- Hervás, O. R. (2014) “Software Defined Networking”. Dissertação de Mestrado - Universitat Politècnica de Catalunya.
- Mckeown, N.; Anderson, T.; Balakrishnan, H.; Parulkar, G.; Peterson, L.; Rexford, J.; Shenker, S.; Turner, J. (2008) “OpenFlow : enabling innovation in campus networks”, p.1–6.
- Mininet (2015) “Mininet: an instant virtual network on your laptop (or other pc)”. Disponível em < <http://mininet.org/> >. Último acesso em 24/06/2015.
- Mozzaquatro, B. A.; Machado, C. C.; Lucca, M. R. B. (2013) COFFEE: controlador openflow de fluxo para escoamento eficiente de streaming de vídeo. XII Simpósio de Informática - SIRC 2013, p.20–25.