

UMA INTERFACE DE GERENCIAMENTO DE REDES DEFINIDAS POR SOFTWARE

Fagner Jefferson de Araújo Silva; Whasley Sousa Cardoso; Marcelo Portela Sousa.

Instituto Federal de Educação, Ciência e Tecnologia da Paraíba IFPB – Campina Grande.

1 - INTRODUÇÃO

O conceito de Redes Definidas por Software (SDN) é um assunto muito estudado e com diversos tipos de abordagens. É uma tecnologia de grande interesse acadêmico e comercial, pois ainda emergente. Por ser uma tecnologia ainda muito recente, na maioria das vezes exige do pesquisador ou desenvolvedor um esforço relativamente grande para montar um ambiente que propicie um bom desempenho no aprendizado da tecnologia, desenvolvimento dos recursos da rede, na configuração de regras de comunicação e análise de resultados.

Este artigo tem como objetivo apresentar uma ferramenta de código aberto capaz de ajudar no estudo e gerenciamento de redes definidas por software. O foco principal é abstrair o controlador da rede com uma aplicação que oferece uma interface de linha de comando. A comunicação com o controlador é realizada através de requisições web usando arquitetura Rest que é comum na maioria dos controladores SDN.

Os controladores SDN apesar de terem o mesmo propósito possuem muitas diferenças em termos de configuração, linguagem de desenvolvimento e usabilidade. A proposta para resolver essas dificuldades é uma interface genérica, que permita o gerenciamento de controladores compatíveis com Openflow, permitindo ao desenvolvedor ou pesquisador de redes definidas por software ter uma maior flexibilidade no desenvolvimento de novos comandos para a ferramenta abstraindo configurações mais específicas de ambiente ou controladores.

2 - METODOLOGIA

Para o início do desenvolvimento foi montado um ambiente da maneira mais simples, que permitisse simular virtualmente uma topologia de rede gerenciada por um controlador de redes definidas por software.

Uma máquina virtual foi utilizada neste experimento configurada com 1GB de memória, 8GB de armazenamento e um processador Pentium dual-Core de 2GHz. O sistema operacional instalado foi o Ubuntu Server 16.04

Para criar uma rede virtualizada foi utilizado o Mininet (Mininet, 2017). Mininet é um simulador de redes que possibilita a criação gerenciamento de topologias no contexto de SDN. O Floodlight (Floodlight, 2017) foi utilizado como o controlador SDN.

A linguagem de programação escolhida para o desenvolvimento foi Python 2.7 (Python, 2010), de fácil aprendizagem e presente na maioria dos cursos introdutórios de programação, além disso, possui um bom desempenho sendo usada para diversas finalidades. Foi necessário instalar também algumas bibliotecas do Python para desenvolver algumas funcionalidades.

- **Click:** segundo (Click, 2017) o Click é um pacote Python para o desenvolvimento de Interface em linha de comando usando Python;
- **SQLite3:** usado para fazer a persistência dos dados da configuração iniciais da aplicação;
- **Requests:** biblioteca usada para fazer requisições web com Python.

Feito o desenvolvimento das primeiras funcionalidades, a ferramenta está disponível no repositório do github para controle de versão com o nome de **sdnCommand** no link <https://github.com/fagnerfjas/SdnCommands>.

Para conferir a usabilidade do **sdnCommand** os serviços do controlador Floodlight foram iniciados executando o comando “**sudo java -jar floodlight/target/floodlight.jar**”. Em outro terminal, foi gerada uma topologia de teste no mininet, o comando utilizado foi: “**sudo mn -controller=remote, port=6653 -topo=tree,2**” no servidor Ubuntu. A Figura 1 representa a topologia criada com o Mininet.

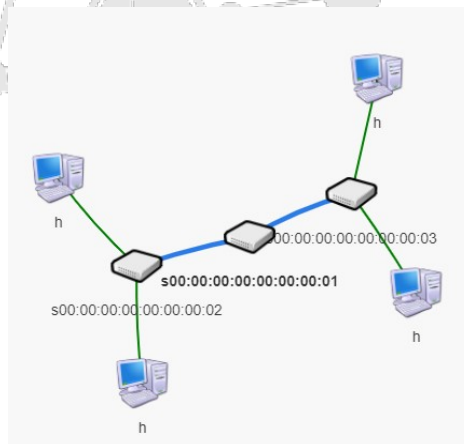


Figura 1 – Topologia gerada no Mininet e exibida na interface web do Floodlight.

3 - RESULTADO

O cenário está pronto e a interface de gerenciamento de redes definidas por software pode ser usada para o gerenciamento da rede. O primeiro passo ao usar o **sdnCommand** é verificar as opções disponíveis digitando a comando padrão “sdn” ou o comando padrão mais algum subcomando seguido da opção “--help”, por exemplo “sdn acl –help”.

```
fjas@SDNServer: ~
(venv) fjas@SDNServer:~$ sdn
Usage: sdn [OPTIONS] COMMAND [ARGS]...

Aplicação para gerenciamento de controladoras SDN, com interface em linha
de comando

V 0.1 - 2017
Autor: Fagner Jefferson
Organização: IFPB

Options:
  --clear  Usado juntamente com um subcomando
  --help  Show this message and exit.

Commands:
  acl      CONTROLE DE ACL (Access Control List)...
  conf     CONFIGURAÇÃO INICIAL! Grava em um banco de...
  device   Lista de dispositivos Mostra o json com a...
  file
  flow
  show_conf Exibe as configurações atuais.
  switches LISTA DE SWITCHS Mostra o json com a lista...
(venv) fjas@SDNServer:~$
```

Figura 2 – Documentação de ajuda do sdnCommand.

É preciso fazer uma configuração inicial, antes de executar comandos para gerenciar o controlador , usando os comandos “sdn show_conf” e “sdn conf”, respectivamente é possível ver a configuração atual e inserir uma nova configuração cujos parâmetros são IP do servidor onde foi instalado o controlador, a porta usada para as requisições Rest e um nome de identificação como ilustrado na Figura 3.

```
fjas@SDNServer: ~
(venv) fjas@SDNServer:~$ sdn show_conf
http://192.168.15.4:8080 -- name: floodlight
(venv) fjas@SDNServer:~$ sdn conf 10.0.0.105 8080 Controlador_FloodLight
New Conf: 10.0.0.105:8080 Controlador_FloodLight
(venv) fjas@SDNServer:~$ sdn show_conf
http://10.0.0.105:8080 -- name: Controlador_FloodLight
(venv) fjas@SDNServer:~$
```

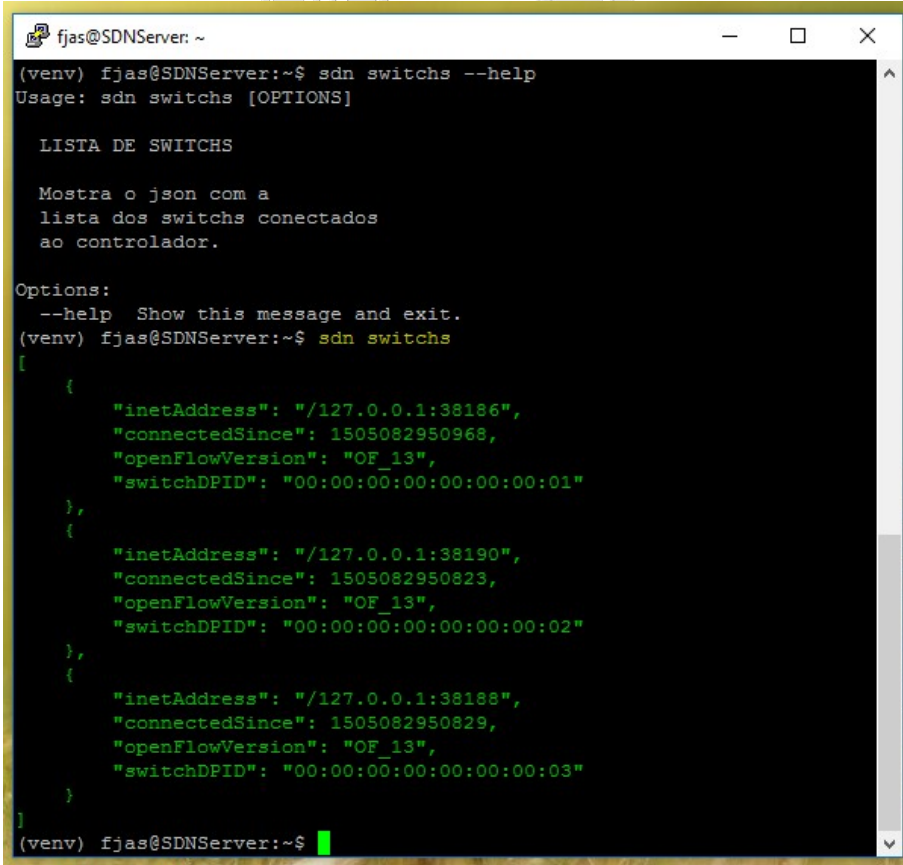
Figura 3 – Comandos de configuração.

Na versão inicial estão disponíveis os seguintes comandos:

- **sdn acl**, possui vários parâmetros para criar listas de controles de acesso. Este é um recurso disponível na Restful API do Floodlight que foi adaptado para linha de comando;
- **sdn device**, exibe as informações dos dispositivos conectados na rede;

- **sdn switches**, exibe uma lista com informações dos switches da rede;
- **sdn flow**, dependendo do parâmetro usado pode exibir a lista de fluxos instalados no controlador, ou inserir um novo fluxo escrito no formato json e em um arquivo txt;
- **sdn file**, abre um editor para edição de um arquivo com informações do fluxo.

Com essas configurações é possível fazer uso do **sdnCommand** pelo prompt de comandos da máquina onde está instalado. A Figura 4 mostra um exemplo da listagem dos switches na rede.



```
fjas@SDNServer: ~  
(venv) fjas@SDNServer:~$ sdn switches --help  
Usage: sdn switches [OPTIONS]  
  
LISTA DE SWITCHS  
  
Mostra o json com a  
lista dos switches conectados  
ao controlador.  
  
Options:  
--help Show this message and exit.  
(venv) fjas@SDNServer:~$ sdn switches  
[  
  {  
    "inetAddress": "/127.0.0.1:38186",  
    "connectedSince": 1505082950968,  
    "openFlowVersion": "OF_13",  
    "switchDPID": "00:00:00:00:00:00:01"  
  },  
  {  
    "inetAddress": "/127.0.0.1:38190",  
    "connectedSince": 1505082950823,  
    "openFlowVersion": "OF_13",  
    "switchDPID": "00:00:00:00:00:00:02"  
  },  
  {  
    "inetAddress": "/127.0.0.1:38188",  
    "connectedSince": 1505082950829,  
    "openFlowVersion": "OF_13",  
    "switchDPID": "00:00:00:00:00:00:03"  
  }  
]  
(venv) fjas@SDNServer:~$
```

Figura 4 – Lista de switches na topologia.

Uma das principais funcionalidades da ferramenta de comandos SDN é a administração de fluxos programados no controlador. Através da programação desses fluxos o controlador da rede pode orientar o comportamento dos equipamentos de comutação da rede.

Atualmente através da ferramenta de comandos SDN a forma de inserir o fluxo é com arquivos texto. As informações deverão ser escritas no arquivo seguindo o formato JSON para os fluxos Openflow como mostra a Figura 5 e em seguida a Figura 6 mostra o comando de inserção de fluxo.



JOIN

ENCONTRO INTERNACIONAL DE
JOVENS INVESTIGADORES
EDIÇÃO BRASIL

```
fjas@SDNServer: ~  
(venv) fjas@SDNServer:~$ vim sdnApp/flows/flow1.json  
1  
2 "switch": "00:00:00:00:00:00:02",  
3 "name": "flow-mod-1",  
4 "cookie": "0",  
5 "priority": "32768",  
6 "in_port": "1",  
7 "active": "true",  
8 "actions": "output=2"  
9
```

Figura 5 – Arquivo com informações de um fluxo no formato JSON.

```
fjas@SDNServer: ~  
(venv) fjas@SDNServer:~$ sdn flow -f sdnApp/flows/flow1.json  
{ "status" : "Entry pushed" }  
(venv) fjas@SDNServer:~$
```

Figura 6 – Comando para inserir fluxo no controlador.

O parâmetro “-f” indica a entrada de um arquivo de fluxo para ser adicionado. O parâmetro “-l” serve para listar os fluxos.

4 - CONCLUSÃO

A usabilidade da linha de comando pode não ser tão agradável para iniciantes em estudos relacionados à computação. Porém é uma forma muito intuitiva e prática de se trabalhar, no formato que foi implementada torna-se mais fácil o seu desenvolvimento. A ferramenta de comandos SDN está no início do seu desenvolvimento e já deu uma boa noção das poderosas possibilidades de implementação. A partir dos resultados obtidos pode-se destacar um ganho na flexibilidade de gerenciamento de fluxos em uma abordagem SDN. A ferramenta apresentada está disponibilizada de maneira gratuita para que outros pesquisadores possam fazer uso e contribuir para a melhoria de suas funcionalidades.

Nas as próximas etapas de desenvolvimentos será melhorar a visualização das informações, ao imprimir na tela as informações no formato de tabelas ao invés do formato JSON, também a adaptação de novos controladores como OpenDayLight e o Ryu Controller.

REFERÊNCIAS BIBLIOGRÁFICAS:

FloodLight, Get Started – Getting Started, Disponível em: <<https://floodlight.atlassian.net/wiki/spaces/floodlightcontroller/pages/1343542/Getting+Started>>. Último acesso em 11 de Setembro de 2017.

Mininet – An Instant Virtual Network on Your Laptop, Disponível em: <<http://mininet.org/>> Último acesso em 11 de Setembro de 2017.

Click – An Command Line Library For Python, Disponível em: <<http://click.pocoo.org/5/>> Último acesso em 11 de Setembro de 2017.

Luiz E. Borges Python 2010. **Python para Desenvolvedores ed 2**. Rio de Janeiro, Edição do Autor, 2010 ISBN 978-85-909451-1-6

Costa, L. R. **OpenFlow e o Paradigma de Redes Definidas Por Software**. (Monografia apresentada como requisito parcial para conclusão do Curso de Computação — Licenciatura) – Universidade de Brasília – UNB. Brasília 2013.

