
OrchFlow

Uma Ferramenta para Orquestração de
Múltiplos Controladores OpenFlow

OrchFlow
Manual de instalação e uso

UFSCAR - Universidade Federal de São Carlos
SOROCABA - 2016



**LERIS - Laboratory of Studies in
Networks, Innovation and Software**
UFSCAR - Sorocaba
<http://www.sorocaba.ufscar.br/ufscar>

Título:

OrchFlow - Uma Ferramenta para Orquestração de Múltiplos Controladores OpenFlow

Evento:

Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)

Fórum:

Salão de Ferramentas do SBRC - 2016

Grupo de trabalho:

Universidade Federal de São Carlos (UFSCAR) - Sorocaba

Participante(s):

Marcelo Frate
Marcelo K. M. Marczuk

Orientador(s):

Fábio L. Verdi

Cópias: 1

Numero de páginas: 21

Data de Publicação:

5 de março de 2016

Resumo:

O principal objetivo das redes definidas por software é a centralização da lógica de controle, porém é possível dividir esta lógica entre dois ou mais controladores com o intuito de garantir a escalabilidade. O protocolo OpenFlow define a comunicação entre switches e controladores, entretanto não prevê a comunicação entre controladores, necessária para qualquer tipo de distribuição no plano de controle. Faz-se necessário portanto o desenvolvimento de soluções independentes do protocolo, capazes de distribuir essa lógica dentro de um mesmo domínio administrativo. Neste cenário, o OrchFlow surge como uma ferramenta capaz de orquestrar uma rede definida por software, com dois ou mais controladores OpenFlow, permitindo a gerência e o monitoramento da topologia em tempo real.

O conteúdo deste manual está disponível gratuitamente, mas a publicação (com referência) somente com a devida autorização dos autores.

Sumário

Prefácio	ix
1 Introdução	1
1.1 Arquitetura do OrchFlow	1
2 Funcionamento do OrchFlow	3
2.1 Detalhes de implementação	3
3 Requisitos de Software	5
3.1 Funcionalidades	5
3.2 Requisitos	5
4 Requisitos de Software	7
4.1 Funcionalidades	7
4.2 Requisitos	7
4.3 Inicialização do sistema de testes	8
5 Desenvolvimento do Módulo	13
5.1 Banco de Dados - Neo4j	13
5.2 Módulo ARPreply	14
5.3 Módulo Reactive	14
6 Interface	15
7 Conclusão	17
Bibliografia	19
A Appendix A name	21

Todo list

Prefácio

Universidade Federal de São Carlos (UFSCar) Sorocaba, 5 de março de 2016

Marcelo Frate
<frate@ifsp.edu.br>

Marcelo K. M. Marczuk
<marcelo.marczuk@dcomp.sor.ufscar.br>

Fábio L. Verdi
<verdi@ufscar.br>

Capítulo 1

Introdução

O OrchFlow, surge como uma ferramenta que tem como objetivo, funcionar como um *Middleware*, um software orquestrador para as SDN, com dois ou mais controladores baseadas no protocolo OpenFlow [5], proporcionando ao administrador da rede uma visão global, capaz de aprovisionar serviços, com monitoramento da topologia em tempo real. OrchFlow é capaz de receber solicitações de serviços através de uma interface Norte, (*Northbound*), processá-las e então mapeá-las através de uma interface Sul (*Southbound*), de forma a prover o serviço solicitado em uma rede controlada por múltiplos controladores OpenFlow. Assim, quando um determinado serviço é solicitado, o OrchFlow define como deverá ser o tratamento por parte dos diversos controladores na rede até que sejam aplicadas todas as regras OpenFlow aos elementos de rede, permitindo que os recursos estejam disponíveis.

1.1 Arquitetura do OrchFlow

Como se pode ver na Figura 1.1, o OrchFlow atuará como um agente integrador entre as diversas aplicações disponíveis na rede e os diferentes controladores OpenFlow, sob um mesmo controle administrativo, definido aqui como Domínio Administrativo.

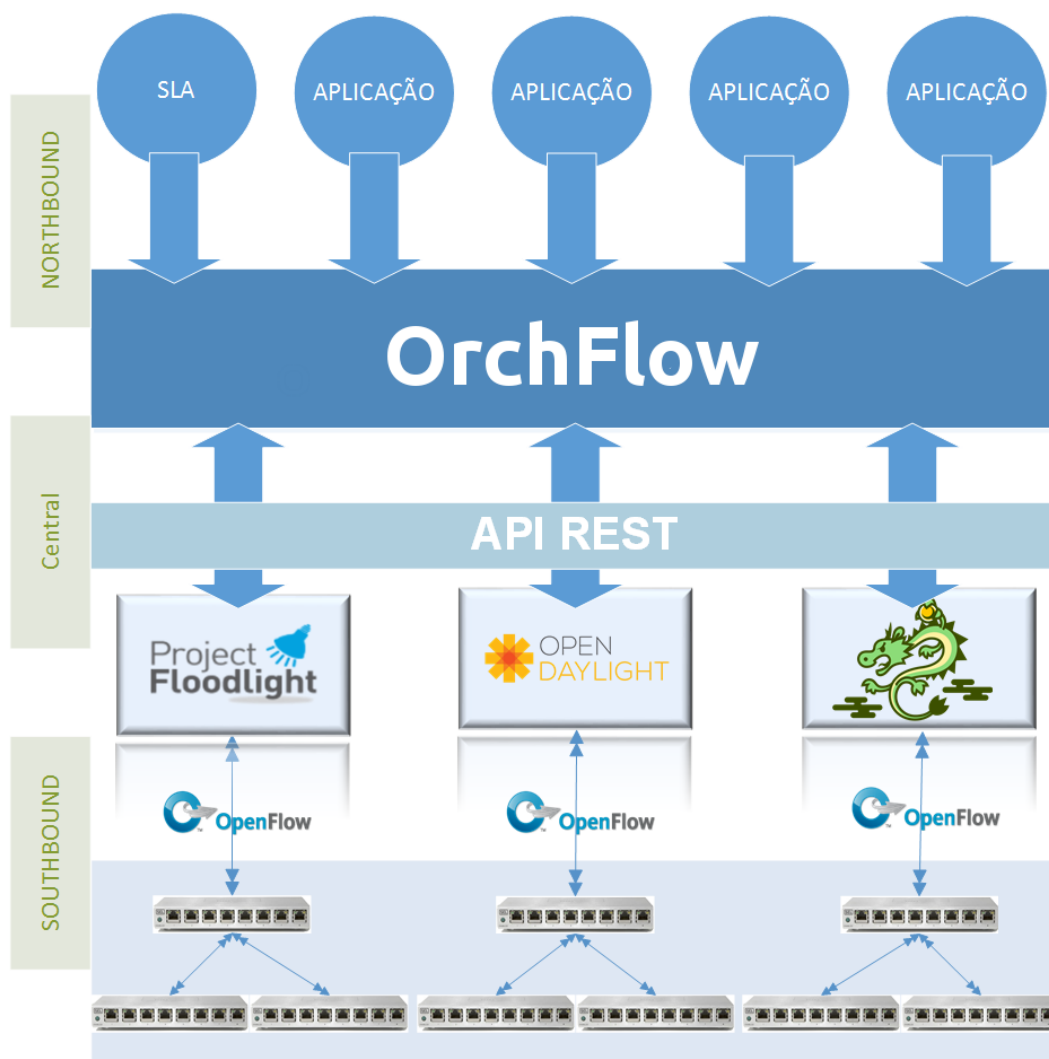


Figura 1.1: Arquitetura utilizada nos testes.

Capítulo 2

Funcionamento do OrchFlow

Na primeira camada da arquitetura estão os switches, formando toda a infraestrutura necessária para a comunicação entre os hosts. Para efeito de testes e uso deste manual, optamos por uma rede emulada através do emulador Mininet [4];

2.1 Detalhes de implementação

A arquitetura de rede proposta, tem para efeito de testes, um domínio administrativo composto por três subdomínios, com um controlador OpenFlow cada. Cada controlador é responsável pelo controle e gerenciamento de um único subdomínio e cabe ao OrchFlow a orquestração, gerencia e a visão completa do Domínio Administrativo.

A Figura 2.1, ilustra a topologia utilizada nos testes e no desenvolvimento do OrchFlow. Utiliza-se aqui quatro máquinas virtuais (VM):

VM2, VM3 e VM4: Em cada uma, um controlador Floodlight já está instalado, configurado, funcionando e pronto para receber as chamadas dos switches das redes instaladas na VM 1;

VM1: Nesta VM, o usuário deverá emular a sua rede. Para facilitar o acesso e uso, deixamos um script pronto de uma rede emulada, contendo três subdomínios interligados em forma de anel, com 7 switches cada em forma de árvore e 4 hosts;

Toda a infraestrutura necessária para a implantação e testes do OrchFlow está baseada na plataforma Linux. Assim, o LERIS, disponibiliza um servidor para acesso remoto com todas as máquinas virtuais, com o banco de dados Neo4j e com o OrchFlow já instalados e configurados, para que todos os testes sejam realizados.

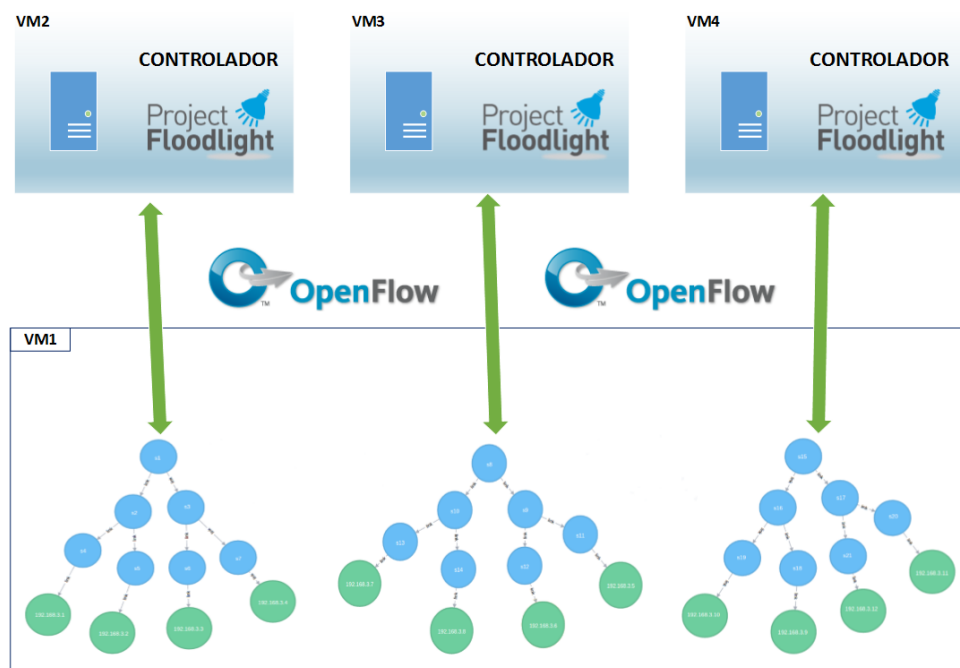


Figura 2.1: Topologia de rede.

Capítulo 3

Requisitos de Software

3.1 Funcionalidades

- Centralização da gestão da rede de computadores;
- Visão global do domínio administrativo;
- Criação de rotas estáticas;
- Criação de rotas dinâmicas em tempo real;
- Administração das rotas criadas;
- Cadastro e manutenção dos controladores da rede;

3.2 Requisitos

- javac 1.8.0_72;
- Banco de Dados Neo4j V2.3;
- Mininet V 2.2.1;
- OpenFlow V 1.3;
- ovs-vsctl (Open vSwitch) 2.0.2;
- 3 Controladores Floodlight;
- Módulo ARPReply instalado nos 3 controladores;
- Módulo Reactive instalado nos 3 controladores;

Capítulo 4

Requisitos de Software

4.1 Funcionalidades

- Centralização da gestão da rede de computadores;
- Visão global do domínio administrativo;
- Criação de rotas estáticas;
- Criação de rotas dinâmicas em tempo real;
- Administração das rotas criadas;
- Cadastro e manutenção dos controladores da rede;

4.2 Requisitos

- javac 1.8.0_72;
- Banco de Dados Neo4j V2.3;
- Mininet V 2.2.1;
- OpenFlow V 1.3;
- ovs-vsctl (Open vSwitch) 2.0.2;
- 3 Controladores Floodlight;
- Módulo ARPReply instalado nos 3 controladores;
- Módulo Reactive instalado nos 3 controladores;

4.3 Inicialização do sistema de testes

Para executar o experimento será necessário o acesso, via SSH, às quatro máquinas virtuais, criadas especialmente para o salão de ferramentas do SBRC. Você poderá acessar este servidor através do endereço IP: 200.133.238.125

Para a realização dos testes, recomendamos usar um terminal de console linux, porém, o acesso poderá ser feito através de programas como o PUTTY em outros sistemas operacionais.

Para tanto, abra um terminal de console linux para cada VM e execute os seguintes comandos:

VM do Controlador 1

Acesso ao servidor OrchFlow

```
ssh -X orchflow@200.133.238.125
```

```
orchflow@200.133.238.125's password: orchflow
```

```
orchflow@ServerLeris:~$
```

Para acesso ao controlador C1:

```
orchflow@ServerLeris:~$ ssh -X mininet@192.168.56.99
```

```
mininet@192.168.56.99's password: mininet
```

```
mininet@Controlador1:~/floodlight$ sudo su
```

```
[sudo] password for mininet: mininet
```

```
root@Controlador1:/home/mininet/# cd floodlight
```

```
root@Controlador1:/home/mininet/floodlight/# java -jar target/floodlight.jar -cf  
src/main/resources/neutron.properties
```

VM do Controlador 2

Acesso ao servidor OrchFlow

```
ssh -X orchflow@200.133.238.125
```

```
orchflow@200.133.238.125's password: orchflow
```

```
orchflow@ServerLeris:~$
```

Para acesso ao controlador C2:

```
orchflow@ServerLeris:~$ ssh -X mininet@192.168.56.100
```

```
mininet@192.168.56.100's password: mininet
```

```
mininet@Controlador2:~/floodlight$ sudo su
```

```
[sudo] password for mininet: mininet
```

```
root@Controlador2:/home/mininet/# cd floodlight
```

```
root@Controlador2:/home/mininet/floodlight/# java -jar target/floodlight.jar -cf  
src/main/resources/neutron.properties
```

VM do Controlador 3

Acesso ao servidor OrchFlow

```
ssh -X orchflow@200.133.238.125
```

```
orchflow@200.133.238.125's password: orchflow
```

```
orchflow@ServerLeris:~$
```

Para acesso ao controlador C3:

```
orchflow@ServerLeris:~$ ssh -X mininet@192.168.56.98
```

```
mininet@192.168.56.98's password: mininet
mininet@Controlador3:~/floodlight$ sudo su
[sudo] password for mininet: mininet
root@Controlador3:/home/mininet/\# cd floodlight
root@Controlador3:/home/mininet/floodlight\# java -jar target/floodlight.jar -cf
src/main/resources/neutron.properties
```

VM do Mininet

```
Acesso ao servidor OrchFlow
ssh -X orchflow@200.133.238.125
orchflow@200.133.238.125's password: orchflow
orchflow@ServerLeris:~$
Para acesso ao Mininet:
orchflow@ServerLeris:~$ ssh -X mininet@192.168.56.101
mininet@192.168.56.101's password: mininet
Pmininet@mn1:~$ cd mininet/
mininet@mn1:~/mininet$ sudo ./custom/topologia.py
```

Por último abra o seu navegador e digite o seguinte endereço:
<http://200.133.238.125:8080/OrchFlow/>

Pronto, você estará na página inicial do OrchFlow, Figura 4.1, aqui basta apertar o botão iniciar.

Na página seguinte, Figura 4.2, você deverá cadastrar os controladores utilizados na sua rede. Para o nosso experimento, os endereços utilizados foram: Controlador 1: 192.168.56.99 na porta 8085 Controlador 2: 192.168.56.100 na porta 8085 Controlador 3: 192.168.56.98 na porta 8085

Note que estes, são os endereços das máquinas virtuais criadas especificamente para este evento, ficam a critério do usuário modificar os endereços podendo são

, desenvolvido e implementado em linguagem Java e um banco de dados não relacional, o Neo4j. As três redes estão definidas e interligadas através de links específicos, denominados aqui de links externos, utilizando ainda switches virtuais OVS-Switch e o protocolo OpenFlow na versão 1.3.

E para a realização dos testes, alguns dos serviços de rede mais conhecidos estão sendo propostos, como os serviços de roteamento fim a fim, como forma de validar esta ferramenta.

Para cada subconjunto de switches, definidos aqui como subdomínio, existe um controlador, responsável pela configuração, por manter as informações da topologia e monitorar o estado da rede. Cada controlador está conectado ao OrchFlow através de uma interface Central.

O OrchFlow, atua como um orquestrador, definindo a programação para todos os controladores do domínio administrativo e possibilita a comunicação entre as diferentes aplicações através de uma interface Norte, uma interface padronizada, capaz de receber solicitações e invocar serviços pré-determinados. Essas interfaces fazem parte de um único sistema WEB, utilizando *Representational State Transfer* (REST) [3], um protocolo que torna possível a troca de informações entre aplicativos e serviços web, pela qual são solicitados todos os recursos necessários para o estabelecimento das rotas fim a fim. Ao receber tais

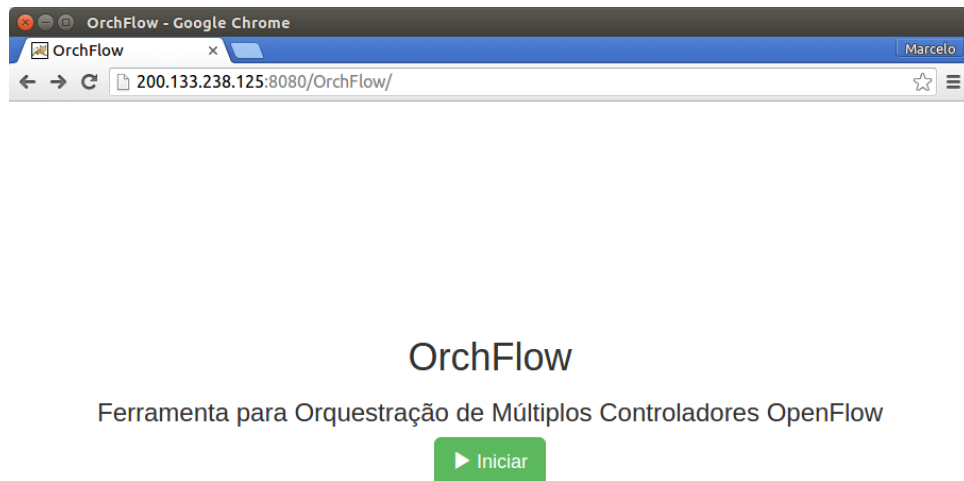


Figura 4.1: Página Inicial.

solicitações, o OrchFlow as processa e de forma orquestrada atuará sobre cada um dos controladores por duas maneiras possíveis, conforme o subdomínio a ser alcançado.

Pró-ativo: A aplicação inicia, via interface Norte, a solicitação de um serviço ao OrchFlow, que faz toda a orquestração necessária aos controladores via interface Central. Neste modo, toda a programação é executada pelo OrchFlow e transferida aos controladores, via interface REST padrão, todas as regras de fluxos necessárias, para que haja a comunicação fim a fim, diretamente aos switches envolvidos;

Reativo: Um novo evento na rede, chegando através da interface Sul, faz com que, um controlador busque em sua programação a correta correspondência e faz a inserção dos fluxos necessários. Para isso, o OrchFlow disponibilizará uma interface de programação central que possibilitará ao administrador da rede, criar a sua programação de forma centralizada e descarregá-las nos controladores. Neste modo, os controladores OpenFlow, respondem diretamente às requisições de seu subdomínio e encaminham para a porta de saída correspondente ao subdomínio vizinho todas as requisições que não pertençam a este subdomínio, possibilitando assim a entrega de pacotes entre subdomínios, independentemente de quantos forem os subdomínios que compõem o domínio administrativo.



The screenshot shows a web browser window titled "OrchFlow - Google Chrome" with the URL "200.133.238.125:8080/OrchFlow/configurar.xhtml". The page has a user name "Marcelo" in the top right corner. The main heading is "Cadastrar Controladores". Below it is the instruction "Adicione os controlador que serão orquestrados pelo OrchFlow". There are three input fields labeled "Nome:", "IP: *", and "Porta: *", each with a corresponding text input box. Below these fields is a button labeled "Adicionar". Below the button is a table with four columns: "Nome", "IP", "Porta", and "Ação". The table contains three rows of data. Below the table are two buttons: "✖ Remover todos" and "✔ Concluir".

Cadastrar Controladores

Adicione os controlador que serão orquestrados pelo OrchFlow

Nome:

IP: *

Porta: *

Nome	IP	Porta	Ação
C1	192.168.56.99	8085	<input type="button" value="✖ Remover"/>
C2	192.168.56.100	8085	<input type="button" value="✖ Remover"/>
C3	192.168.56.98	8085	<input type="button" value="✖ Remover"/>

Figura 4.2: Configurar.

Capítulo 5

Desenvolvimento do Módulo

O desenvolvimento do OrchFlow busca proporcionar a integração, a gerência, a administração e a programação para SDN baseada em múltiplos controladores OpenFlow, para isso ele deverá ser capaz de ler e gravar as diversas programações em cada um dos controladores, de forma a garantir a perfeita comunicação entre quaisquer hosts pertencentes ao mesmo domínio administrativo.

5.1 Banco de Dados - Neo4j

Para que haja a correta leitura e consolidação dos dados é necessário armazená-los de forma eficiente e que os dados estejam bem estruturados. Assim, foi escolhido o banco de dados Neo4j, que como se pode ver em [2], trata-se de um software livre baseado em Java, com banco de dados em grafo, que oferece persistência, alto desempenho, escalabilidade, uma comunidade ativa e uma boa documentação, capaz de receber os dados de cada um dos hosts e switches e representá-los como nós, obter os dados dos links e criar um relacionamento entre eles, estabelecendo um conjunto completo de dados, oferecendo ainda uma interface gráfica, permitindo ao administrador obter a visão completa da rede, de todo o domínio administrativo.

Para que o sistema funcione, é necessário que os controladores reconheçam a topologia da rede, para isso todos os controladores contam com um módulo de reconhecimento de topologia de seu subdomínio, assim, cada elemento de rede, bem como os hosts são identificados e depois esses dados são encaminhados através de uma interface REST para o OrchFlow que fará a composição das informações obtidas de todos os controladores da rede, integrando-as numa só base de dados, o NEO4j, à partir de onde serão realizadas todas as consultas.

Todas as rotas definidas pelo administrador ou por solicitação de alguma aplicação ao OrchFlow, serão criadas tomando por base o algoritmo de Dijkstra [1], que faz uma busca pelo caminho mais curto e determina aos controladores que criem os fluxos seguindo conforme o resultado encontrado por ele. Assim, qualquer rota, mesmo as que passam por dois ou mais subdomínios, serão definidas por seus controladores seguindo conforme o que foi encontrado pelo OrchFlow em sua topologia.

Para efeito de testes e funcionamento exclusivo com o OrchFlow, foram desabilitados

todos os módulos *forward* que permitam o funcionamento como switches legados, assim para todo e qualquer fluxo que chegue à rede, é necessário que haja um correspondente na tabela de fluxos, caso não haja, os controladores devem possuir uma programação prévia, para determinarem a configuração à todos os switches envolvidos na comunicação. Porém, caso um fluxo novo chegue à rede e o mesmo não encontrar correspondência na tabela de fluxos ou não houver uma programação que possa tratá-lo de forma correta, este será descartado.

5.2 Módulo ARPReply

Toda a comunicação na rede, baseada no protocolo TCP-IP exige o uso do endereço IP, que é utilizado para roteamento, ou seja, para a escolha do melhor caminho entre dois hosts. Para isso, é necessário mapear o endereço de nível superior IP para endereço físico Ethernet, como foi proposto pela RFC826, o Address Resolution Protocol (ARP) [6]. O ARP permite que um host encontre o endereço físico de um host destino, tendo apenas o seu endereço IP. Entretanto, numa rede OpenFlow, é necessário que os fluxos ARP sejam tratados previamente, para que os fluxos TCP sejam encaminhados.

Nesse sentido, cada controlador, reconhece a topologia de seu subdomínio e a repassa ao OrchFlow, de posse das informações de todo o domínio administrativo cria-se então uma tabela de conversão MAC/IP e a devolve aos controladores através de uma interface REST, onde um módulo previamente instalado irá possibilitar que os controladores respondam às requisições ARP, permitindo que todos os hosts na rede recebam os endereços MAC destino para cada novo fluxo que solicitarem, assim, mesmo que um host solicite uma comunicação com outro host fora de seu subdomínio, o controlador já terá os dados necessários para devolução da correta mensagem de retorno do protocolo ARP. Essa solução traz um ganho à rede, eliminando o broadcast de mensagens ARP desnecessárias.

5.3 Módulo Reactive

Como vimos na seção 2 a programação dos controladores é realizada de duas maneiras, nos modos proativo e reativo. No modo proativo, um serviço ou ação do administrador da rede, solicita ao OrchFlow que encaminhe os parâmetros aos controladores, para que sejam gravados diretamente na tabela de fluxos dos switches, possibilitando assim, que todos os fluxos sejam encaminhados diretamente às portas de destino.

Entretanto, para que o modo reativo funcione é necessário a instalação do módulo *Reactive* em cada um dos controladores, esse módulo permite às aplicações ou ao administrador da rede que faça a programação de seu domínio administrativo em tempo de execução. Neste modo, um serviço ou ação do administrador da rede, solicita ao OrchFlow que encaminhe aos controladores a programação necessária para que, quando um fluxo chegar e não encontrando nas tabelas de fluxos do primeiro switch um fluxo correspondente, este receba do controlador do subdomínio, o preenchimento de sua tabela, liberando o fluxo a partir deste momento.

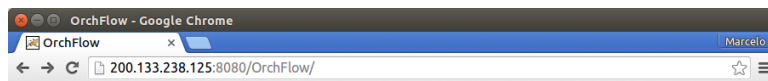
Capítulo 6

Interface

Texto Em construção.....

.
. .

A ferramenta apresenta do lado direito da página a topologia da rede, dentro da interface Neo4j, é possível executar os comandos desse banco de dados, desde uma mudança de cor para os nós e arestas, mudança no tamanho, identificadores, entre outros. É possível realizar também comandos de busca de melhor caminho entre dois hosts, enfim, todos os comandos do Neo4j estão livres para uso.



OrchFlow

Ferramenta para Orquestração de Múltiplos Controladores OpenFlow

▶ Iniciar

Figura 6.1: Interface.

Do lado esquerdo da interface é possível identificar os dois modos de operação do OrchFlow no menu de opções e criar as regras conforme a necessidade da aplicação. Para



Cadastrar Controladores

Adicione os controlador que serão orquestrados pelo OrchFlow

Nome:

IP: *

Porta: *

Nome	IP	Porta	Ação
C1	192.168.56.99	8085	<input type="button" value="✖ Remover"/>
C2	192.168.56.100	8085	<input type="button" value="✖ Remover"/>
C3	192.168.56.98	8085	<input type="button" value="✖ Remover"/>

Figura 6.2: Controladores.

ambos os modos o procedimento é o mesmo, basta preencher os campos necessários para a comunicação.

Antes é preciso cadastrar todos os controladores da rede, colocando nome, endereço IP da interface web e porta http, como por exemplo: Controlador1, 192.168.1.1,8085. Este procedimento é obrigatório, só assim o OrchFlow poderá realizar as consultas e determinar as configurações a cada subdomínio.

Uma vez cadastrado todos os controladores, o sistema está pronto para uso, já é possível realizar todos os tipos de programações possíveis. Para ajudar o leitor, deixaremos um manual de instalação e uso disponível em onde todos os detalhes de implantação e possíveis testes estarão explicados passo a passo, de forma a contribuir com comunidade acadêmica em busca de novas soluções aos problemas de escala em SDN.

Capítulo 7

Conclusão

O OrchFlow, possibilita que as aplicações tirem proveito de todas as funcionalidades dos recursos físicos da infraestrutura de uma rede baseada em software de forma automatizada, espera-se demonstrar que a integração entre controladores diferentes e a sua orquestração responda aos problemas de escala que atualmente existem nas SDN baseadas no protocolo OpenFlow.

O OrchFlow, uma ferramenta capaz de automatizar o processo de programação de múltiplos subdomínios, integrados à um único domínio administrativo, com uma visão global, centralizada, possibilitando um funcionamento híbrido, nos modos proativo e reativo, dando aos fluxos de maior frequência a velocidade necessária e aos fluxos esporádicos, regras com um tempo delimitado. Tais características aproveitam melhor os recursos do hardware, como o uso da memória TCAM, possibilitando maior vida útil aos switches OpenFlow, com uma consequente redução dos custos com a tecnologia.

Bibliografia

- [1] EW Dijkstra. “A note on two problems in connexion with graphs:(numerische mathematik, _1 (1959), p 269-271)”. Em: (1959).
- [2] Emil Eifrem. “Neo4j-the benefits of graph databases”. Em: *no: sql (east)* (2009).
- [3] Roy Thomas Fielding. “Architectural Styles and the Design of Network-based Software Architectures”. Em: *Building* 54 (2000), p. 162. DOI: 10.1.1.91.2433. URL: <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>.
- [4] Bob Lantz, Brandon Heller e N McKeown. “A network in a laptop: rapid prototyping for software-defined networks”. Em: ... *Workshop on Hot Topics in Networks* (2010), pp. 1–6. ISSN: 1450304095. DOI: 10.1145/1868447.1868466. URL: <http://dl.acm.org/citation.cfm?id=1868466>.
- [5] Nick McKeown et al. “OpenFlow”. Em: *ACM SIGCOMM Computer Communication Review* 38.2 (2008), pp. 69–74. ISSN: 01464833. DOI: 10.1145/1355734.1355746.
- [6] David C Plummer. “RFC 826: An ethernet address resolution protocol”. Em: *InterNet Network Working Group* (1982).

Apêndice A

Appendix A name

Here is the first appendix