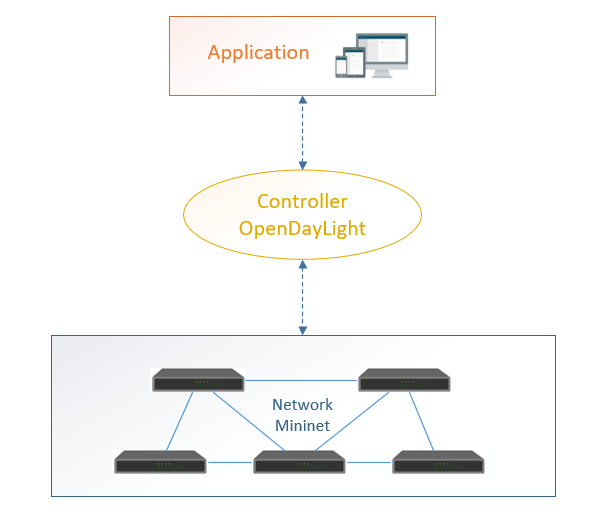
Engenharia Informática

Laboratório de Tecnologias de Informação

***Cloud – Analytics – Service Assurance***



Índice

[Resumo 3](#_Toc510796656)

[1. Introdução 3](#_Toc510796657)

[2. Camada de Aplicação 4](#_Toc510796658)

[3. Controlador SDN 5](#_Toc510796659)

Resumo

O presente relatório descreve todo o trabalho realizado no âmbito da unidade curricular de Laboratório de Tecnologias de Informação, onde desenvolvemos uma aplicação que é capaz de comunicar com um controlador **SDN** que por sua vez consegue gerir um conjunto de equipamentos numa rede virtual.

1. Introdução

Seguindo uma topologia ***Cloud*** – ***Analytics*** – ***Service*** ***Assurance*** o nosso trabalho divide-se nas seguintes camadas.

Camada aplicacional desenvolvida em C#;

Controlador **SDN** OpenDaylight;

Rede virtual Mininet.

1. Camada de Aplicação

Com a necessidade de existir uma interface capaz de interagir com o controlador **SDN** (OpenDaylight), desenvolvemos uma aplicação em C# capaz de fazer pedidos à **API** incorporada no OpenDaylight.

Partindo deste pressuposto a nossa aplicação é capaz:

- Obter informação através de pedidos **GET**;

- Enviar informação através de pedidos **POST**;

- Atualizar informação através de pedidos **PUT**.

De seguida iremos descrever todo o processo de desenvolvimento nesta camada.

Numa primeira fase começamos por estudar o processo de comunicação através de **HttpWebRequest** em C#.

Uma vez garantido a ligação da nossa aplicação ao controlador **SDN**, tivemos de analisar/investigar toda a informação e documentação do OpenDaylight referente aos pedidos **GET**, **POST** e **PUT**.

As principais dificuldades neste ponto foi a escassa informação por parte do OpenDaylight relativa aos pedidos à **API** e no parse do **JSON** recebido através dos pedidos **GET**, ainda assim, conseguimos com mais ou menos esforço atingir os nossos objetivos.

A nossa aplicação consegue com sucesso, ligar-se ao controlador **SDN**, executar os pedidos **GET**, fazer parse do **JSON** recebido e mostrar numa caixa de texto informações como:

- NodeID;

- Node Termination Points;

Relativamente aos pedidos **POST**, a nossa aplicação dispõe de um formulário com campos obrigatórios que devem ser preenchidos pelo utilizador que quando submetidos através de um click no botão **Create** enviam a informação em **XML** para o controlador **SDN**. Este **POST** é responsável por criar um **Flow**.

Os **PUT** atualizam a informação dos **Flow** e os **POST** com delete eliminam os **Flow** existentes.

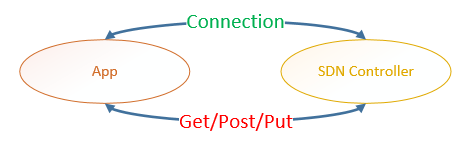


Figura 1 - Interação App c/ SDN Controller

Continuação app….

1. Controlador SDN

O controlador **SDN** que estamos a utilizar no projeto é o OpenDaylight e foi instalado numa máquina ubuntu server 17.10.

No processo de instalação as maiores dificuldades estiveram relacionadas com a instalação do Java e com a necessidade de descobrir quais os pacotes relacionados com o acesso à **API** (odl-restconf-all).

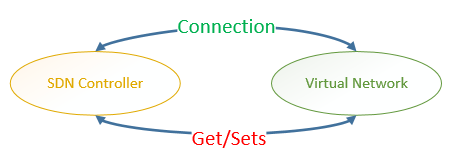


Figura 2 - Interação SDN & Virtual Network

Pontos de instalação:

Obter pacote OpenDaylight para a ubuntu server:

Wget https://nexus.opendaylight.org/content/repositories/public/org/opendaylight/integration/karaf/0.8.0/karaf-0.8.0.tar.gz

Descompactação:

tar -xvf karaf-0.8.0.tar.gz

Iniciar OpenDaylight:

cd karaf-0.8.0

./bin/karaf

Instalar features no OpenDaylight

feature:install odl-restconf odl-l2switch-switch odl-mdsal-apidocs odl-dlux-core odl-dluxappsapplications odl-dluxapps-topology odl-dluxapps-yangutils

O OpenDaylight possui uma interface web onde podemos consultar a topologia de rede criada no Mininet.

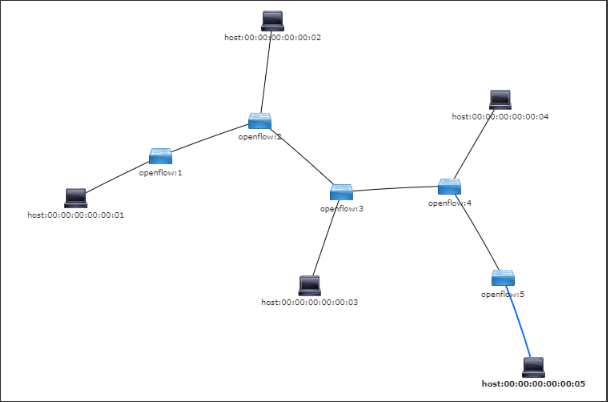


Figura 3 - OpenDaylight Network Topology

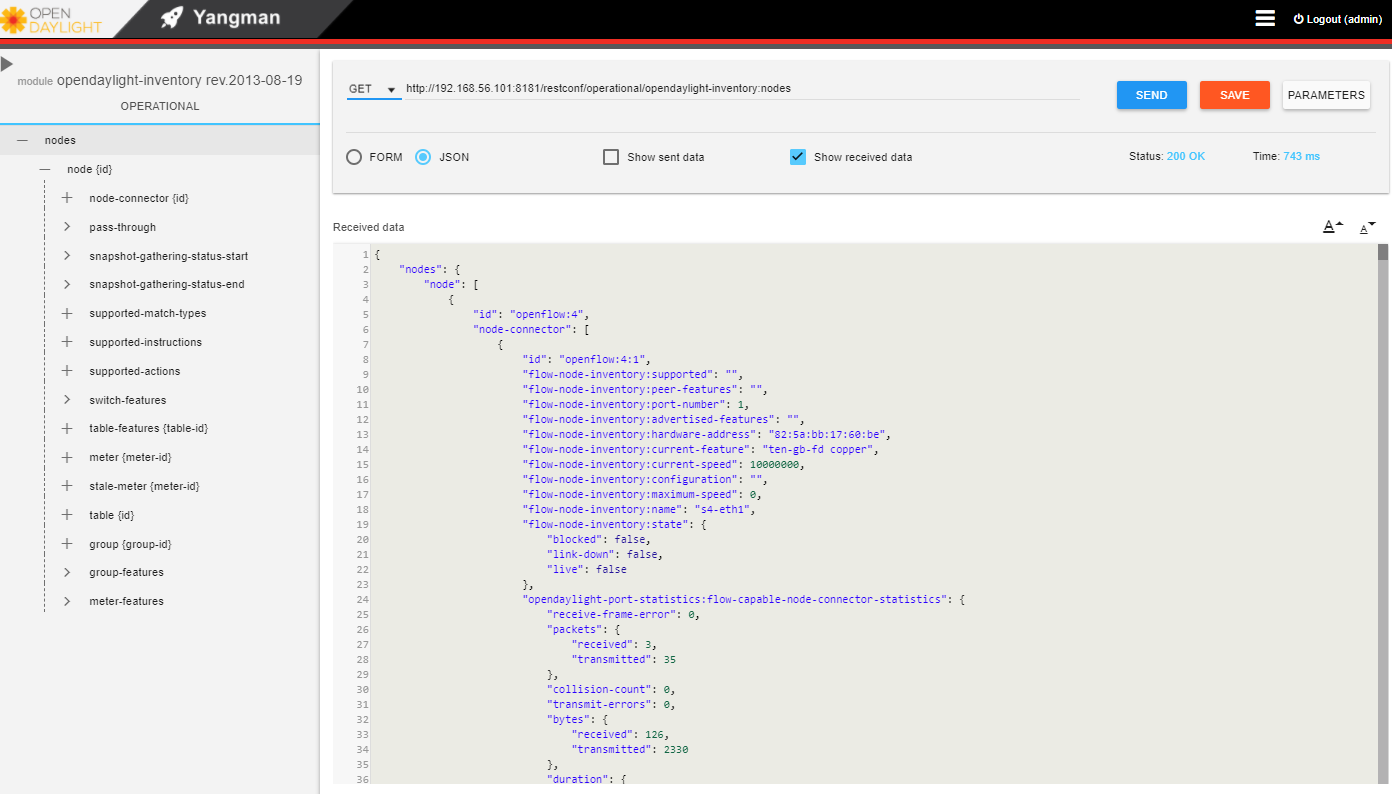


Figura 4 - OpenDaylight Yangman

1. Mininet