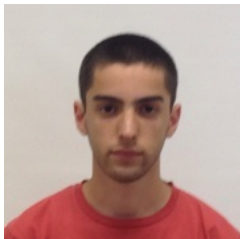




**Universidade do Minho**  
Escola de Engenharia

# Trabalho Prático 3

VIRTUALIZAÇÃO DE REDES  
GRUPO 4



Mário Costa Silva  
A75654



Nuno Gabriel Areal  
A74714

## Introdução

Neste relatório pretendemos expor a nossa resolução do terceiro trabalho prático, realizado no âmbito da disciplina de Virtualização de Redes. Para a melhor compreensão das tecnologias utilizadas serão estudados e apresentados os temas redes SDNS, protocolo *Openflow*, controlador *floodlight* e a sua interação com o *mininet* e por fim serão discutidas as diferentes entre o protocolo de encaminhamento usado no *Floodlight* e o protocolo *spanning tree* usado nas redes tradicionais.

A parte prática deste trabalho baseia-se na criação e gestão de um rede SDN constituída por vários *switches* , 2 servidores DNS , um primário e um *slave* , dois servidores de ficheiros e dois clientes criação de uma imagem *Docker* com o *mininet* , *tcdump* e *wireshark* funcionais. Os passos para a concretização deste projeto são:

- Estudo dos temas Redes SDN, Protocolo *Openflow*, Interação *Floodlight* e *Mininet* , Comparação entre o protocolo de encaminhamento do *Floodlight* e *Spanning Tree*.
- Configuração do controlador *Floodlight* e da rede *mininet* para comunicação de em tanto em *anycast* como em *unicast*.
- Configuração dos servidores DNS
- Desenvolvimento dos programas que serão executados nos servidores de ficheiros de modo a responderam às queries dos clientes.

## Teórica

### Redes SDN

*Software-Defined Networking* (SDN) é uma arquitetura de redes que atribui a propriedade de programabilidade através de aplicações de software. Isto permite uma muito melhor gestão da rede independentemente do das tecnologias de nível mais baixo, por exemplo o hardware, que estão a ser utilizadas pois a comunicação é feita através de API's standard e não através de tecnologias proprietárias promovendo assim a inovação.

A estrutura de uma SDN é composta por 3 partes os controladores, que oferecem uma visão da rede e implementam políticas de *forwarding* de tráfego. *Southbound* APIs , que efetuaram a comunicação entre os controladores e os *routers/switches* , sendo o *Openflow* o primeiro protocolo standard adotado. Por ultimo *Northbound* APIs , que efetuaram a comunicação entre as aplicações de alto nível e os controladores , sendo as principais funcionalidades o *shapping* de tráfego e implementação de serviços nos controladores.

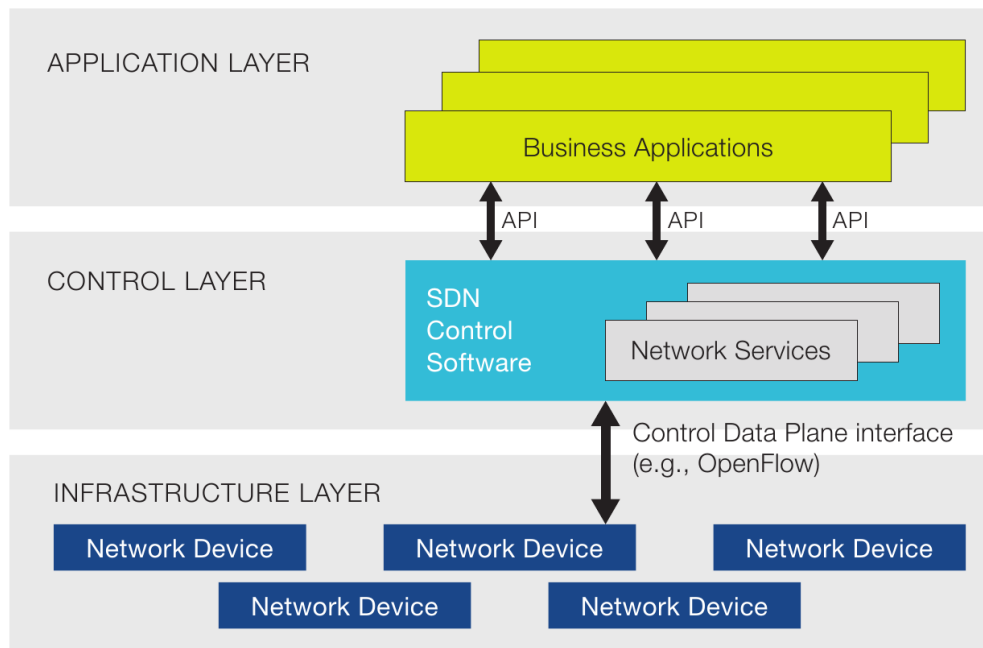
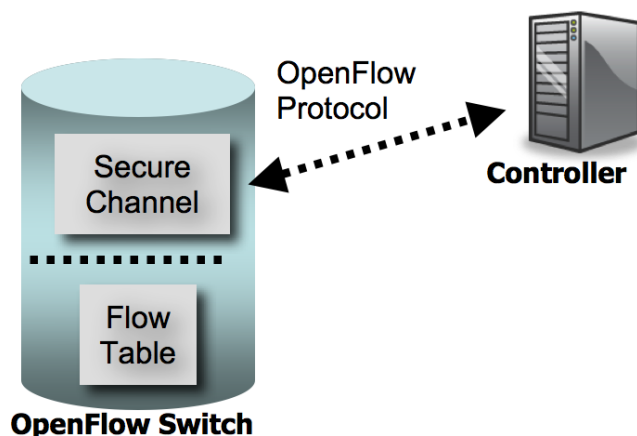


Figura 1

## Protocolo Openflow

O *Openflow* é um protocolo de comunicação entre os controladores e os dispositivos de rede como routers e *switches* pertencendo ao conjunto dos *Southbound APIs* e foi o primeiro standard deste tipo que permanece no topo da popularidade.

Esta tecnologia permitiu a separação do *layer* dados e de controlo, que se encontram juntos num dispositivo tradicional, passando a funcionalidade de controlo para o denominado *OpenFlow switch*. Com o uso deste protocolo é possível modificar as tabelas de *forwarding* de todos os dispositivos simultaneamente, pois numa rede tradicional era necessário utilizar o software para modificar cada tipo de dispositivo individualmente e existiam sempre restrições como funcionalidades implementadas em uns mais não noutros.



## Floodlight e Mininet

O *Floodlight* é o controlador SDN de todos os *switches* da topologia criada. Dessa forma é capaz de capturar e alterar todos os pacotes que passam pelos. É possível que ao receber um pacote de *ARP* num *switch*, modificar o endereço *MAC* do IP pretendido, executando imediatamente um *ARP Reply*. Conseguimos também obter estatísticas de utilização dos links de cada *switch*.

## Comparação entre Floodlight e Spanning Tree

Por defeito o *Floodlight* calcula as rotas através do menor número de saltos, atualizando as tabelas de 500 em 500 milissegundos.

O protocolo de *spanning tree* tem como objetivo criar uma topologia sem *loops*. Este constrói um árvore com base nos links disponíveis, “desativando” os outros para evitar os *loops*, ligando-os quando um dos links previamente ativos falham.

## Prática

### Configuração Floodlight

Toda a configuração foi feita com base no *MACTracker*, modificando depois os parâmetro necessários para obter os resultados.

Na demonstração do trabalho foi observado que o cliente 1 estava a receber pacotes duplicados. O nosso código está bastante restrito quanto aos pacotes em que procede a alterações, efetuando apenas a pacotes destinados a 10.0.0.250 e 10.0.0.251 com o endereço *MAC* ff:ff:ff:ff:ff:ff. A única explicação que encontramos para a sua repetição é o facto de definirmos a porta de saída como `OFPort.Flood`, `OFActionOutput output = myFactory.actions().buildOutput().setPort(OFPort.FLOOD).build();`

Não tínhamos também o controlo de carga a funcionar como esperado pois o controlador não é capaz de enviar pacotes, uma vez que não tem endereço IP, mas conseguiria capturar eventuais pacotes enviados pelos servidores de ficheiros ou então poderíamos implementar nos clientes uma escolha sempre que fosse pretendido enviar pacotes para os servidores.

Para a ativação das estatísticas foi necessário modificar esta linha `net.floodlightcontroller.statistics.StatisticsCollector.enable` para `TRUE` no ficheiro `floodlightdefault.properties` de modo a que fosse possível executar o modulo `StatisticsCollector`.

## Servidores de Ficheiros

Para os servidores de ficheiros foi desenvolvida uma aplicação simples em java que escuta na porta .... (UDP) de pacotes enviados pelos clientes. As funcionalidades implementadas são:

- Download de um ficheiro
- Upload de um ficheiro
- Informar sobre a carga do servidor

Foi escolhida uma diretoria que será partilhada pelos dois servidores com ficheiros exemplo. Esta aplicação está presente na pasta **fileserver**.

## Servidores DNS

A implementação dos servidores foi feita através do Bind9, como proposto no enunciado. Temos um servidor primário e um secundário (*slave* com transferência de zona) com o nome **nemvrelas**. A configuração é básica e apenas têm as modificações necessárias para o seu funcionamento correto. Os ficheiros de configurados estão presentes na pasta **dns**.