À Minha Família

Esta página foi intencionalmente deixada em branco

Agradecimentos

Gostaria de agradecer a todos os que, direta ou indiretamente, me ajudaram a terminar este grande objetivo que representa a conclusão de mais uma etapa da minha vida académica.
Em particular, gostaria de fazer um agradecimento especial aos meus pais, Jaime e Fátima, e à minha avó, Lira, por, ao longo de todos estes anos de estudo, me apoiarem, se sacrificarem por mim e me aturarem em todos os momentos. Obrigado por acreditarem em mim! Estar-vos-ei eternamente grato por tudo!
Quero também agradecer ao meu orientador, Professor Carlos Rabadão, pelas sugestões e pela oportunidade para realizar esta dissertação que muito gosto me deu. Também ao Professor Paulo Valente, por todos os conhecimentos e informações transmitidas, que serviram para valorizar este trabalho e foram muito importantes no esclarecimento de alguns dados.
Quero agradecer ao Instituto Politécnico de Leiria, à Escola Superior de Tecnologia e Gestão e a todos os Professores que comigo se cruzaram nestes anos que aqui estive. Foi deles que tentei absorver o máximo de conhecimentos e valores. Também a eles devo o que eu sou hoje. Um grande obrigado a todos!
Para terminar, um último agradecimento a todos os meus amigos.
Obrigado a todos por tudo!

Esta página foi intencionalmente deixada em branco

Nota prévia

Da execução desta dissertação resultou na escrita de dois artigos científicos submetidos para as seguintes conferências:

- The Twelfth International Conference on Networking and Services (ICNS 2016);
- 11ª Conferência Ibérica de Sistemas e Tecnologias de Informação (CISTI'2016).

No que diz respeito à disponibilização de informação e esclarecimento de dúvidas relativas à sua temática, esta dissertação teve também o contributo do Professor Paulo Valente.

Esta página foi intencionalmente deixada em branco

Resumo

Com a evolução da Informática e com o aparecimento de novos dispositivos com acesso à *Internet*, o número destes dispositivos subiu acentuadamente, logo, o fluxo de tráfego na rede também.

Apesar deste aumento de fluxo, os utilizadores desejam, cada vez mais, que o seu acesso à *Internet* e os conteúdos que utilizam sejam rápidos e que haja inovações nos serviços das operadoras a que estão associados. Por isso, é importante simplificar a rede tradicional.

Esta simplificação leva à programação de redes que, por sua vez, está diretamente relacionada com o SDN. Também a virtualização de redes está relacionada com este tema da programação de redes, visto terem surgido duas abordagens que são convergentes e que se podem complementar: SDN e NFV.

Tendo em conta a vantagem e o potencial do SDN, a sua utilização na gestão de serviços de comunicação veio a revelar-se muito útil para colmatar alguns dos problemas até aí encontrados. Por isso propõe-se a criação de uma solução de apoio ao desenvolvimento e teste de redes e serviços de comunicação antes de estes serem colocados em produção. A automatização de processos ou utilização de ferramentas de apoio à evolução e otimização da rede fazem com que a mesma se torne mais rápida, intuitiva e menos suscetível a falhas. Estes fatores levam a que novos serviços possam ser criados e colocados mais rapidamente em produção, como é desejo de consumidores e operadoras de telecomunicação. Assim, tanto os recursos humanos como os recursos materiais serão muito mais bem aproveitados. Tudo isto, tendencialmente, levará a uma melhoria dos resultados financeiros das operadoras.

Palavras-chave: SDN, fluxo de tráfego, virtualização, programação de redes.

Esta página foi intencionalmente deixada em branco

Abstract

With the evolution of information technology and the emergence of new devices with Internet access, the number of these devices has risen sharply, so the flow of traffic on the network as well.

Despite this flow increase, users want more and more, that their access to the Internet and the content they use are fast and there is innovation in services of which operators they are associated. So it is important to simplify the traditional network.

This simplification leads to programming networks, in turn, is directly related to the SDN. Also virtualization networks is related to this network programming theme, as they have emerged two approaches that are converging and could be complemented: SDN and NFV.

Taking into account the advantages and potential of SDN, its use in the communication service management came to prove very useful to address some of the problems encountered so far. Therefore it is proposed to create a solution to support the development and testing of networks and communication services before they are put into production. The automation of processes or use of the evolution and network optimization support tools make it may become faster, intuitive and less susceptible to failure. These factors mean that new services can be created and placed quickly in production, as is the desire to consumers and telecom operators. Thus, both human resources and material resources will be much better use. All this tends to, will lead to improved financial results of operators.

Key-Words: SDN, flow of traffic, virtualization, network programming

Esta página foi intencionalmente deixada em branco

Índice

Índice de Figuras

Índice de Tabelas

Esta página foi intencionalmente deixada em branco

Lista de Siglas

AAA	Authentication, Authorization e Accounting
ACE	Access Control Engines
ACI	Application Centric Infrastructure
ACL	Access Control Lists
API	Application Programming Interface
APIC	Application Policy Infrastructure Controller
AS	Autonomous System
AVS	Aplication Virtual Switch
CapEx	capital expenditure
CDB	Configuration Database
CLI	Command-Line Interface
CPU	Central Processing Unit
CSR	Cloud Services Routers
CTO	Chief Technical Officer
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DSCP	Differentiated Services Code Point
DSL	Digital Subscriber Line
DSL	Domain-Specific Language
ETSI	European Telecommunications Standards Institute
HTTP	HyperText Transfer Protocol

ICN	Information Centric Networking
IETF	Internet Engineering Task Force
IMS	IP Multimedia Subsystem
IP	Internet Protocol
IPTv	Internet Protocol television
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISG	Industry Specification Group
IT	Information Technology
JSON	JavaScript Object Notation
LAN	Local Area Network
LTE	Long-Term Evolution
NaaS	Networking-as-a-Service
NAT	Network Address Translation
NED	Network Element Driver
NFV	Network Functions Virtualization
NIC	Network Interface Controller
NV	Network Virtualization
OCP	Open Compute Project
OFELIA	OpenFlow in Europe: Linking Infrastructure and Applications
ONF	Open Networking Foundation
OpEx	Operational Expenditure
OSI	Open Systems Interconnection model
OSPF	Open Shortest Path First
OTT	Over-The-Top
OVSBD	Open vSwitch Database Management Protocol
PNF	Physical Network Functions

PT	Portugal Telecom
QoS	Quality of Service
RFC	Request for Comments
RIB	Routing Information Base
RPC	Remote Procedure Call
SDCN	Software Driven Cloud Networking
SDDC	Software-Defined Data Center
SDK	Software Development Kit
SDN	Software-Defined Networking
SO	Sistema Operativo
SP	Service Providers
SSH	Secure Shell
SSL	Secure Socket Layer
TCP	Transmission Control Protocol
Telco	Telecommunications company
ToR	Top-of-Rack
TSL	Transport Security Layer
UCS	Unified Computing System
UML	Unified Modeling Language
VAN	Virtual Application Networks
VLAN	Virtual Local Area Networks
VNF	Virtualized Network Functions
VPN	Virtual Private Network
VRS	Virtual Routing and Switching
VSC	Virtualized Services Controller
VSD	Virtualized Services Directory
VSP	Virtualized Services Platform

VXLAN	Virtual eXtensible LAN
WAN	Wide Area Network
XML	eXtensible Markup Language
XNC	eXtensible Network Controller

Introdução

Com o avançar dos anos, a Informática tem vindo a sofrer muitas evoluções. Também a *Internet* tem vindo a evoluir e o surgimento de novos serviços e utilizações é quase uma constante.

A evolução da *Internet* leva a que surjam muitos e novos utilizadores, que usam os mais diversos aparelhos (*tablets*, *smartphones*, *laptops*, etc.) para se ligarem à rede. Esta evolução levou a uma alteração da rede tradicional como é conhecida.

Com o aumento e facilidade de dispositivos ligados à rede (Cisco Systems, Inc., 2015), surgiram novos problemas e limitações, relativos à própria rede, desconhecidos até então. Por isso, tiveram de se descobrir e criar novas formas de resolver este problema. Uma das alterações/adaptações mais importantes foi o fim de novos endereços *Internet Protocol version 4* (IPv4), passando todos os dispositivos e fabricantes a terem de migrar para o endereçamento *Internet Protocol version 6* (IPv6) (Lawson, 2011). A solução dos problemas teria de estar diretamente relacionada com a escalabilidade das redes, alocação de dados, otimização e virtualização, entre outros (Gouveia, Aparício, Parreira, Sargento, & Carapinha, 2013).

Segundo um estudo realizado (Evans, 2011), em 2020 existirão cerca de 50 mil milhões de dispositivos/aparelhos com acesso à *Internet*. O mesmo estudo apresenta um dado interessante que é o facto de, nesse mesmo ano, se prever que cada pessoa tenha, em média, mais de seis dispositivos pessoais nas mesmas condições (Evans, 2011).

Se observarmos bem, este último dado será facilmente alcançado porque, atualmente, já existem várias pessoas que possuem um *smartphone*, um *tablet* e computador portátil ou *laptop*, ou seja, metade dos dispositivos por utilizador previstos para 2020. Com o aparecimento em massa dos relógios inteligentes ou outros dispositivos com acesso à *Internet*, a tendência apresentada no referido estudo é bastante real.

No Fórum Mundial de Economia, realizado em Davos, Suíça, Eric Schmidt, diretor executivo da Google, disse que "a *Internet* iria desaparecer" (Prigg, 2015). No entanto, o que ele quis dizer foi que a *Internet* como hoje a conhecemos e como hoje ainda muitos a utilizam iria desaparecer, ou seja, em vez de acedermos à *Internet* através de *desktops*, futuramente iremos recorrer à *Internet* através de sensores que, de certo modo, já farão parte do nosso dia-a-dia e iremos então pensar na *Internet* como algo natural. A enorme quantidade de endereços *Internet Protocol* (IP) facilitará bastante a existência de *Internet* em muitos dispositivos. Mesmo assim, a *Internet* continuará a existir, como continuarão a existir os *data centers* como hoje os conhecemos, ou com as devidas evoluções.

Também o aumento de conteúdos *Over-The-Top* (OTT), por exemplo, *streaming*, como é o caso do Netflix (Schaefer, 2015), leva a um aumento natural do fluxo de tráfego na rede. Posto isto, as empresas têm de procurar novas soluções de modo a não perderem utilizadores e a manterem estáveis a sua rede, operações e serviços. Estas empresas podem ser tanto de prestação ou fornecimento de serviços (*Service Providers* (SP)) como empresas ou operadoras de telecomunicações, empresas/fabricantes de *hardware* ou inclusivamente empresas de redes.

1.1 Motivação

Existem vários problemas que tiveram de ser resolvidos na rede e pelas operadoras para que houvesse suporte a um aumento considerável de dispositivos e utilizadores nos últimos anos (Cisco Systems, Inc., 2015) (Simões, 2013).

Todos estes problemas levam a consequências. Por exemplo, o aumento de *hardware* novo (*routers*, *switches*, etc.) na rede leva a que o mesmo tenha de ser configurado manualmente ou recorrendo a *scripts*, num terminal. Este procedimento é, atualmente, efetuado por um humano, o que pode facilitar o aparecimento de erros. Multiplicar isto por milhares ou milhões de dispositivos leva a outros problemas que podem surgir nas configurações dos dispositivos, como a complexidade em que a rede se torna, o aborrecimento, que leva ao cansaço por parte de quem aplica as configurações e a limitação de potencial que a rede podia ter[1].

Mais alguns problemas são os custos elevados a que estas alterações obrigam e o espaço físico para novas máquinas que são necessárias. Outro dos problemas é a falta de

[1] Why SDN...Software-defined Networking? <https://www.youtube.com/watch?v=b5JNMDWt4lA>

escalabilidade das redes, o que leva a que estas estejam a estrangular os serviços, isto é, caso não se possa aumentar e deixar crescer a rede, a criação e inovação de serviços nas redes não irá evoluir como se pretende.

Posto isto, as empresas tiveram de criar soluções e têm de as estudar e continuar a aprofundar para que cheguem ao objetivo pretendido. O objetivo principal é que a configuração das redes seja o mais facilitado e simples possível para o administrador da rede, de modo a que este se possa dedicar mais ao desenvolvimento de novos serviços, entre outros pontos, ao invés de se dedicar quase que apenas e exclusivamente à configuração de redes.

As principais soluções apresentadas até ao momento são o *Software-Defined Networking* (SDN) e o *Network Functions Virtualization* (NFV).

A partir do aparecimento do SDN, várias empresas desenvolveram as suas próprias soluções e criaram parcerias de modo a apresentar soluções SDN à sua medida.

1.2 Objetivos

Um dos primeiros objetivos a atingir consiste em compreender bem os conceitos fundamentais do tema abordado nesta dissertação, descritos no próximo capítulo, e que são a base da mesma.

Para isso, é importante fazer uma análise da rede como hoje é utilizada e como hoje ela existe, ou seja, deve-se fazer uma análise à rede tradicional existente. Depois, é importante analisar-se qual o caminho que a rede tende a seguir e para onde é previsível que siga a evolução da mesma. Naturalmente que este caminho poderá não ser exatamente o que agora se imagina que venha a ser utilizado no futuro, mas deverá ser bastante próximo, de acordo com o desenvolvimento feito pela comunidade científica. Um dos caminhos que já está em desenvolvimento por parte dessa comunidade, e que apresenta melhorias no que toca à gestão de redes de comunicação, é o SDN. Tendo em conta este conceito, já apresentado publicamente, é importante fazer um estudo bastante consistente e aprofundado do SDN, bem como das tecnologias a ele associadas. Além do conceito SDN, surgiu também o conceito NFV que, tal como o primeiro, também terá de se analisar, de modo a que se conclua as vantagens e desvantagens de um e de outro conceito.

Com a análise destes conceitos, e com as conclusões obtidas tanto por parte dos especialistas como da análise feita, poder-se-á fazer uma previsão primordial do caminho que a gestão de redes de comunicação possa vir a tomar.

Concluída a análise dos conceitos, é importante fazer o levantamento e estudo do que as empresas da área de redes já desenvolveram, ou mesmo o que ainda estão a desenvolver até ao momento, nomeadamente soluções já apresentadas pelas mesmas. É importante ter um espectro alargado do desenvolvimento efetuado, de maneira a que seja possível tirar conclusões importantes das suas soluções e ver quais os pontos semelhantes e distintos de cada uma dessas soluções apresentadas pelas empresas.

Depois de feita a análise dos conceitos e de algumas soluções, o objetivo principal desta dissertação é, além dos já referidos até aqui, o desenvolvimento de um protótipo de uma ferramenta *Open Source* que permita realizar a gestão de uma rede e respetivos serviços de um modo mais centralizado. Este protótipo pode, entre outras funções, servir para testar redes e serviços de um modo mais simples e facilitado, antes de estes serem colocados em produção. Por forma a verificar o bom funcionamento do protótipo desenvolvido, é importante proceder a algumas validações e testes à ferramenta para que se possam tirar algumas conclusões da sua utilização.

Nesta parte final da apresentação dos objetivos propostos para esta dissertação, é fundamental que se tire algumas conclusões. De modo a poder fazer evoluir o protótipo desenvolvido, deve-se apresentar algumas propostas que possam vir a servir como trabalho futuro e que possam também ajudar à evolução dos conceitos estudados.

1.3 Estrutura da dissertação

De seguida será apresentada a estrutura desta dissertação:

- No presente capítulo, é feita a introdução do tema a aprofundar. É descrita a motivação e é feita a introdução dos objetivos a alcançar. Por fim é referida a estrutura;
- No Capítulo 2 são descritos o enquadramento e os assuntos relacionados com o tema;
- No Capítulo 3 é apresentada a arquitetura definida;
- No Capítulo 4 é apresentado o protótipo desenvolvido e que acaba por ser o resultado do objetivo proposto;
- Por último, no Capítulo 5 apresenta-se a conclusão e o trabalho futuro.

Revisão da literatura

Neste capítulo serão apresentados os conceitos base para uma correta compreensão desta dissertação. Inicialmente são apresentadas as redes e a sua evolução e, posteriormente, aprofunda-se o tema, mais propriamente dito, e apresenta-se alguns exemplos de soluções empresariais já existentes.

2.1 Redes tradicionais

A topologia de rede é a ligação e relacionamento entre vários elementos da rede. Existem dois elementos básicos na topologia da rede: os nós e os *links*.

Os nós representam *routers, switches*, servidores e telefones, entre outros, tal como demonstra a Figura 1.

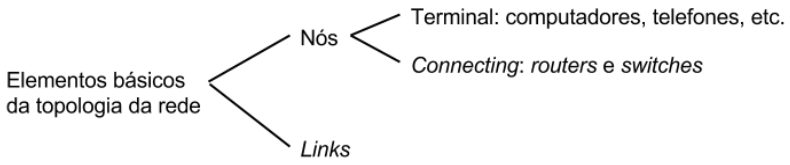


Figura 1 - Elementos básicos da topologia da rede

Além dos elementos básicos, apresentados na Figura 1, existe a estrutura da topologia da rede, que consiste na ligação dos dois elementos básicos. Essa ligação pode ser física ou lógica.

A ligação:

- Física: é a rede real, o que se vê, ou seja, a cablagem física ou fibra que se encontra nos nós;
- Lógica: são as *interfaces* virtuais ou caminhos da rede construídos no topo da infraestrutura física, como por exemplo, a camada 3 do modelo *Open Systems Interconnection* (OSI) (camada de rede).

Relativamente às topologias propriamente ditas, existem pelo menos sete, que estão listadas de seguida e representadas na Figura 2:

- Em anel - *ring* ou circular;
- Em malha (Amaral, 2010) – *mesh*;
- Em estrela – *star*;
- Em malha completa (Amaral, 2010) – *fully connected*;
- Em encadeamento – *line*;
- Em árvore – *tree*;
- Em barramento (Amaral, 2010) – *bus*.

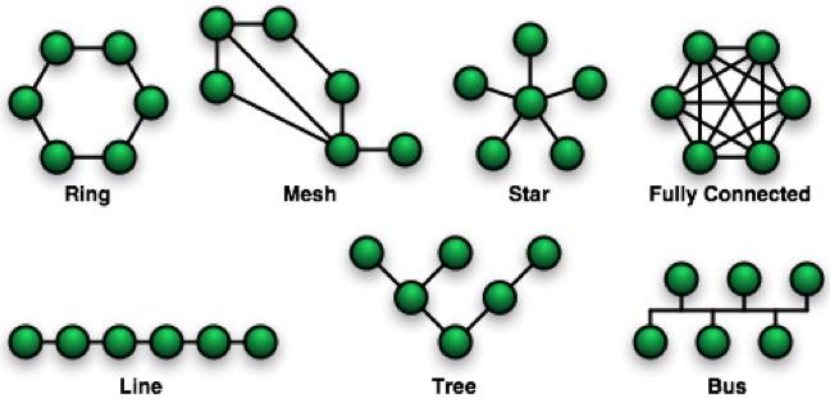


Figura 2 - Sete topologias de rede

A partir da Figura 2 (Nadeau & Gray, 2013) pode ser feita uma análise mais aprofundada das topologias.

Sabe-se que as redes estão cada vez mais complexas (HP Enterprise Business, 2014) e, consequentemente, a sua configuração também se torna complexa e suscetível a falhas. Isto leva a que as configurações manuais de uma rede possam levar horas ou mesmo dias a ser implementadas. Quando um novo elemento básico da rede é integrado, por exemplo um *router*, ele tem de ser configurado para que toda a rede o conheça. Se se pensar em mais equipamentos, facilmente a rede se torna bastante complexa e de grandes dimensões. No relatório (Silva & Santos, 2014), refere-se que a necessidade de novos equipamentos aumenta 50% por ano.

As redes tradicionais ainda são dependentes de *hardware*, têm baixa eficiência e estão muito sujeitas a erros. A evolução destas redes levará a que a configuração se torne mais abstrata, passando a ser mais automatizada com recurso a *software* (Naguib, 2013).

2.2 Evolução das redes

Na sequência do aumento do fluxo de dados e da necessidade, por parte das *Telecommunications company* (Telco), ou empresas/operadoras de telecomunicações, de apresentarem inovações, ou seja, novos serviços e soluções mais eficazes para serem apresentados aos clientes, surgiram aspetos que tiveram de ser melhorados ou facilitados relativamente à configuração de redes informáticas.

Naturalmente que não há uma solução ideal e que algumas Telco seguiram um caminho e outras outro. Apesar disto, tem-se notado uma convergência de duas tendências que evoluíram bastante: *Network* e *Information Technology* (IT), tal como demonstra a Figura 3 (Altice Labs, 2015) apresentada num *webinar*[2] da Portugal Telecom (PT) Inovação, atual Altice Labs, promovido pela Alcatel-Lucent[3].

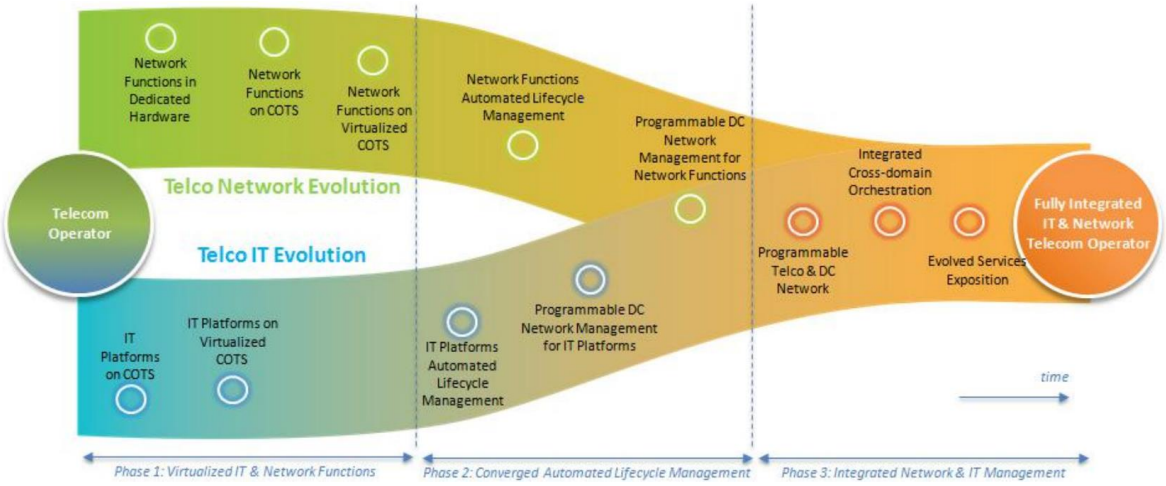


Figura 3 - Evolução do SDN/NFV, segundo as Telco

Relativamente à gestão de redes de comunicação, inicialmente existia uma completa separação por parte das Telco entre a evolução do *Network* e a evolução do IT mas, com a evolução da tecnologia, foi-se observando que, se existisse convergência de ambas as partes, isso poderia ser benéfico para as redes e para a inovação das mesmas. Esta convergência pode ser bem notada na Figura 3.

Se, por um lado, a tendência do *Network* era sustentada pela utilização de protocolos já existentes, por outro o IT está mais relacionado aos *webservices* e virtualização. Tal como se

[2] *Webinar*: "webconferência" ou seminário *online.*

[3] "An NFV/SDN Enabled Service Providers: A New Generation of Digital Services" <https://event.on24.com/eventRegistration/EventLobbyServlet?target=lobby20.jsp&eventid=916706&sessionid=1&key=FA3AFACD50EB5318A25AE9558AF2255E&eventuserid=111164770>

nota pela Figura 3, a junção destas duas tendências traz benefícios e apresenta-se como uma solução exequível.

Da temática de redes programáveis surgiu uma solução principal intitulada *Software-Defined Networking*, mas que ainda está a ser estudada e aprofundada por algumas Telco. Do avançar desta solução, e tendo em conta que várias Telco ainda estão a fazer a sua análise do SDN, surgiu uma outra solução "irmã" (Alexander, 2014) intitulada *Network Functions Virtualization*. Esta solução de virtualização surgiu por parte de sete empresas líderes mundiais da área de telecomunicações, entre elas, AT&T, Telecom Italia e Orange (6WIND, 2013).

Estas duas soluções serão aprofundadas de seguida.

2.2.1 Software-Defined Networking

Tendo em conta que atualmente a rede ainda é bastante estática, um dos paradigmas encontrados para resolver isso é o SDN (Gouveia, Aparício, Parreira, Sargento, & Carapinha, 2013).

Apesar de o termo SDN não ser novo, e de já em 1995 (Dryden, 1995) haver registos do termo *Software-Defined Network* no jornal "Computer World", só há relativamente poucos anos é que o SDN tem vindo a fazer progressos e a ser mais estudado, aprofundado e desenvolvido.

Segundo consta, a ideia da criação do SDN surgiu por parte de investigadores que, aborrecidos de alterar o *software* cada vez que queriam fazer uma alteração na rede, se lembraram: "Por que não criar redes programáveis?" (Pate, 2013).

Avançaram então com a ideia de criar redes programáveis que seriam controladas por um elemento central.

Esta solução tem como principal objetivo ajudar os administradores de redes a gerirem as suas redes do modo mais superficial/genérico possível, evitando assim que os mesmos tenham de as configurar de um modo mais manual ou de baixo nível (Open Networking Foundation, 2012). O facto de se poder implementar serviços rapidamente também é uma vantagem do SDN.

Com o aumento de troca de dados na rede, o aumento do nível de suporte de dados e a diminuição do espaço e das condições necessárias de funcionamento, aliados ao surgimento

do termo "*cloud computing*", a área seguinte de sucesso para o SDN foi o *cloud data center* (Pate, 2013). Em suma, pode-se dizer que o SDN simplifica o trabalho do administrador de redes, bastando para isso que ele utilize um programa do género de um painel de controlo para proceder às configurações (Gouveia, Aparício, Parreira, Sargento, & Carapinha, 2013) (Pinote & Martins, 2013).

Na Figura 4 apresenta-se uma imagem comparativa entre as arquiteturas de rede tradicional e a arquitetura genérica do SDN, baseada na figura apresentada por Frank Yang (Yang, 2013).

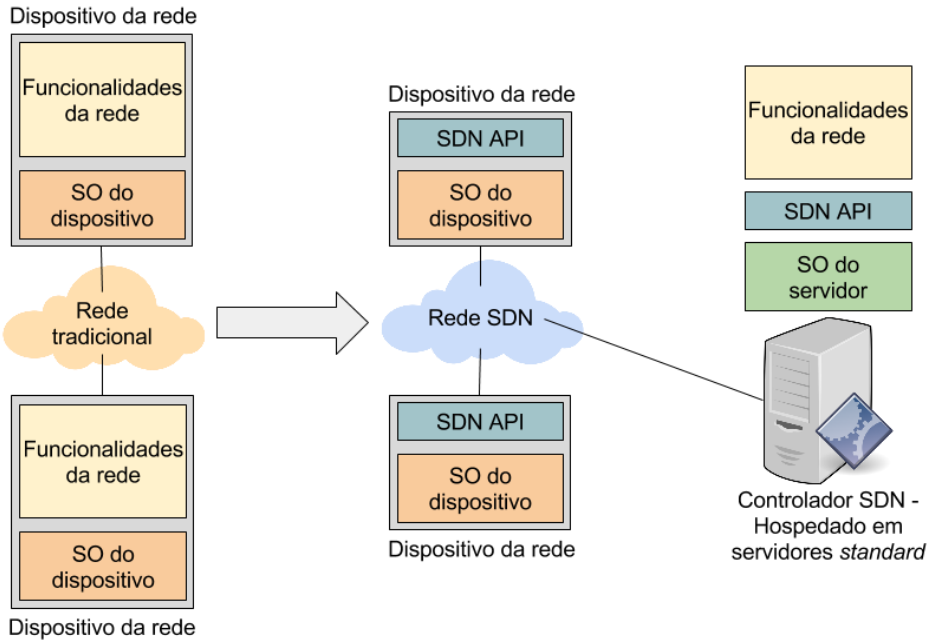


Figura 4 - Comparação entre a rede tradicional e a rede SDN

Na Figura 4, pode-se notar que, utilizando o conceito de SDN, as funcionalidades da rede passam a estar concentradas num ponto central, contrariamente ao que acontece nas redes tradicionais.

Segundo William Stallings (Stallings, 2013), cada vez que se cria uma máquina virtual, ou que a mesma é movida para uma empresa, a sua configuração pode levar dias a ser feita pelos gestores de redes. Por isto, a arquitetura SDN faz a separação entre as funções de controlo e o acesso a servidores de controlo. Assim, a rede é tratada como uma entidade lógica, havendo abstração entre as aplicações e os serviços da rede (Stallings, 2013).

2.2.1.1 Estrutura

O SDN divide-se em três partes, duas delas principais. Essas três partes são: *Application plane*, *Control Plane* e *Data plane*.

A Figura 5 apresenta um esquema genérico do SDN.

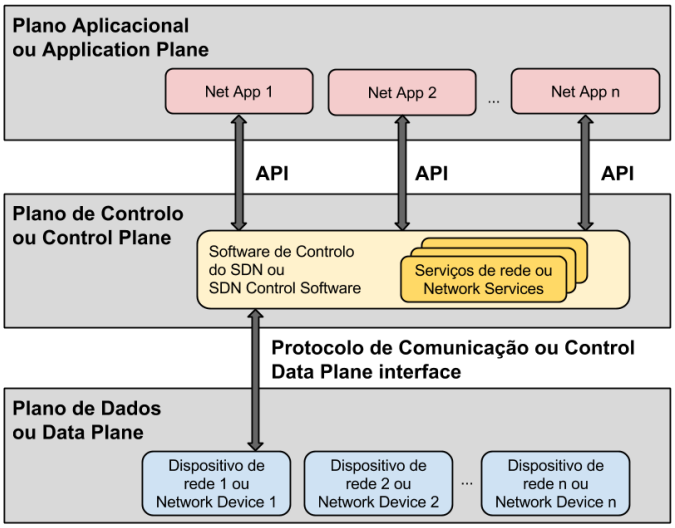


Figura 5 - Esquema genérico do SDN

As duas partes principais são: o plano de dados, ou *Data plane*, e o plano de controlo, ou *Control plane*. Estas podem ser visualizadas na Figura 5, que é baseada na arquitetura *Software-Defined Networking* das referências (Open Networking Foundation, 2012) (Kreutz, et al., 2014). O SDN faz a separação entre estas duas funções das redes convencionais (Pinote & Martins, 2013):

- *Data plane*: serve para analisar os cabeçalhos dos pacotes recebidos e encaminhar esses pacotes para o seu destino final, dependendo das tabelas de encaminhamento e comutação;
- *Control plane*: serve para implementar todos os protocolos de coordenação que sejam necessários para o correto funcionamento do *Data plane*.

Esta separação permite diminuir a complexidade, aumentando a inteligência e tornando a rede mais programável, dinâmica e eficiente. Assim, os *switches* executam funções de *Data plane* e os servidores (controladores da rede) executam funções de *Control plane* (Pinote & Martins, 2013).

No SDN existe ainda um elemento controlador, o *SDN Controller*, ou *SDN Control Software*, que define o fluxo de dados que ocorre no *SDN Data Plane*. Cada fluxo que chega à rede deve primeiro obter permissão do controlador, que verifica que a comunicação é permitida pelo policiamento da rede. Se o controlador permitir o fluxo, é indicado o caminho a seguir e é adicionada uma entrada para esse fluxo em cada um dos *switches* por onde vá passar. Assim, os *switches* apenas necessitam de gerir a tabela de fluxos (Stallings, 2013).

A ligar o *Control plane* e o *Data plane* existe um protocolo de comunicação diretamente associado à arquitetura SDN e que une as duas funções. Para ligar o *Application plane* e o *Control plane* utiliza-se uma *Application Programming Interface* (API) (Stallings, 2013).

Com a separação entre o plano de controlo e de dados, o SDN permite que as aplicações lidem com um único dispositivo de rede, não tendo de se preocupar com mais detalhes de funcionamento (Stallings, 2013).

Uma das principais vantagens do SDN é a fácil e rápida implementação de serviços que esta solução permite (Barry, 2014).

As três principais razões para a utilização de domínios SDN são (Stallings, 2013):

- Escalabilidade: possibilidade de se adicionar controladores na rede consoante a mesma vá crescendo;
- Privacidade: utilização de várias políticas de privacidade em diferentes domínios SDN;
- Desenvolvimento incremental: possibilidade de separar a rede em secções havendo, assim, a hipótese de criar uma rede de testes que não perturbe a secção/rede principal.

Especificando e exemplificando as tecnologias SDN existentes, pode-se chegar a uma figura semelhante à Figura 6, que é baseada na que foi apresentada por William Stallings (Stallings, 2013).

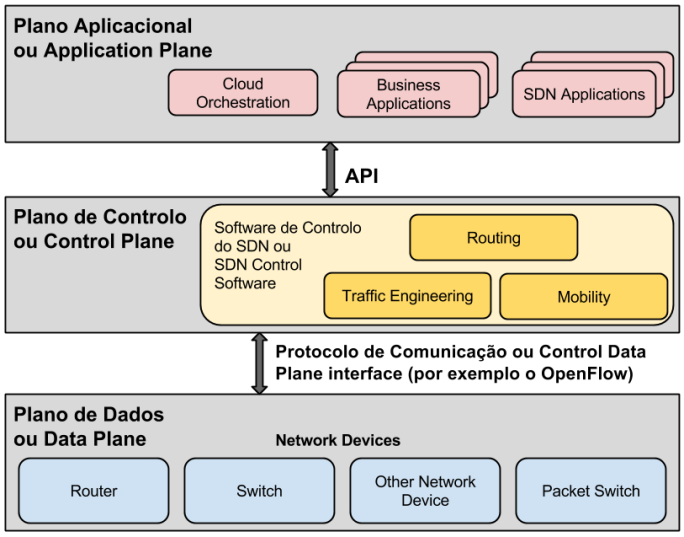


Figura 6 - Estrutura lógica do SDN

A grande diferença da Figura 5 para a Figura 6 é que na 6 são especificados alguns campos que na 5 estavam mais genéricos.

No plano aplicacional referem-se algumas *net apps* como, por exemplo, aplicações de orquestração, de negócio e aplicações SDN.

O *SDN Controller*, que se encontra no *Control plane*, define o fluxo de dados que ocorre no *SDN Data Plane*. Cada fluxo que chega à rede deve, primeiro, obter permissão do controlador, que verifica que a comunicação é permitida pelo policiamento da rede. Se o controlador permitir o fluxo, é indicado o caminho a seguir e é adicionada uma entrada para esse fluxo em cada um dos *switches* por onde vá passar. Assim, estes dispositivos apenas necessitam de gerir a tabela de fluxos (Stallings, 2013). O *SDN Controller* é responsável por fazer a manutenção de todas as regras da rede e disponibiliza as instruções necessárias para a correta manipulação do tráfego (SDxCentral, 2014). A utilização do *SDN Controller* é muito útil para:

- a gestão e distribuição do estado da rede, podendo isto envolver bases de dados, que servem de repositórios de informações da rede;
- uma sessão segura de controlo *Transmission Control Protocol* (TCP) entre o controlador e os agentes associados nos elementos da rede;
- um dispositivo, topologia e mecanismo de serviços de descoberta, entre outras funções (Nadeau & Gray, 2013)

Dois dos protocolos mais conhecidos e que são usados por *SDN Controller* para comunicar com *switches*/*routers* são o *OpenFlow* e o *Open vSwitch Database Management Protocol* (OVSDB), da VMware, Inc. (Rouse, 2013), que é um *software Open Source* para *switches* que os torna *vSwitches* (*virtual switches*) em ambiente de servidores virtuais (Pfaff & Davie, 2013). Outro protocolo que pode ser usado pelos controladores SDN é o NETCONF/*YANG*.

Alguns exemplos de *SDN Controller* são:

- *NOX*: primeiro *SDN Controller*, que foi lançado pela Nicira Networks (SDxCentral, 2014);
- *Floodlight*: é uma ramificação do Beacon (controlador *OpenFlow* baseado em *Java*) (SDxCentral, 2014). É um controlador SDN oferecido pela Big Switch Networks e é utilizado com o intuito de orquestrar o fluxo de tráfego no ambiente SDN. Uma das vantagens da utilização do *Floodlight*, especialmente para os desenvolvedores, é a implementação da ferramenta ser em *Java*, o que facilita a adaptação e desenvolvimento de novas aplicações utilizando este *SDN Controller*. O *Floodlight* funciona tanto em ambiente físico como em ambiente virtual, sendo compatível com

switches que usem *OpenFlow*. Este controlador pode correr como *back-end* da rede para o OpenStack, usando o *Neutron*, que será abordado mais à frente neste documento (SDxCentral, 2014). Por fim é importante referir que o *Floodlight* pode operar com qualquer agente que utilize *OpenFlow*. Na Figura 7 está representado um esquema do *Floodlight*;

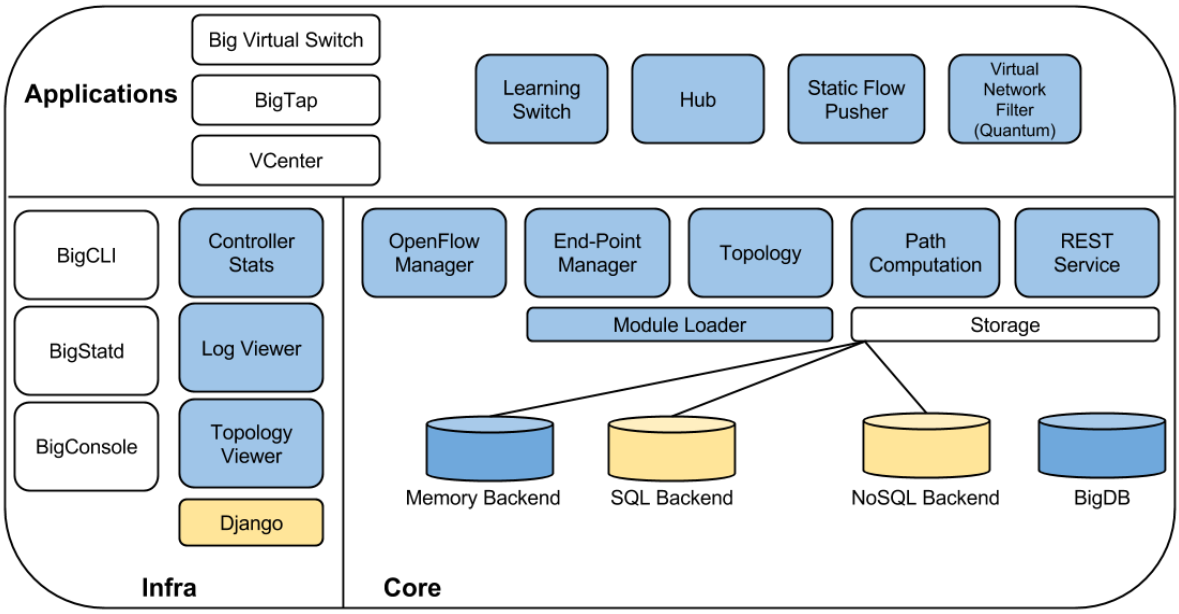


Figura 7 - Esquema representativo do *Floodlight* (Nadeau & Gray, 2013)

- *OpenDaylight SDN Controllers*: controlador SDN da fundação Open Source OpenDaylight, que faz parte do Linux Foundation. O controlador desenvolvido é baseado em *Java* e suporta tanto *OpenFlow* como outros *southbound API* (API usados para comunicar entre o *SDN Controller* e os *switches*/*routers* da rede (SDxCentral, 2014)), por exemplo, o *Cisco OpFlex*. A primeira versão lançada pelo OpenDaylight Project chamava-se *Hydrogen*. É possível observar, na Figura 8, que a sua estrutura tem pontos comuns com a estrutura lógica do SDN, anteriormente apresentada. Os referidos pontos são a apresentação das 3 camadas base do *Software-Defined Networking*, ou seja, *Application plane*, *Control plane* e *Data plane*. Em setembro de 2014, o OpenDaylight Project lançou a sua segunda versão intitulada *Helium*. Este controlador é uma implementação pura de *software*, mas que pode ter várias variantes no ambiente de redes.

  Tanto assim é que empresas como a Cisco ou a Brocade desenvolveram a sua própria solução de *OpenDaylight Controller – Cisco Hydrogen-based Extensible Network*

Controller e *Brocade Helium-based Vyatta Controller*, respetivamente (SDxCentral, 2014).

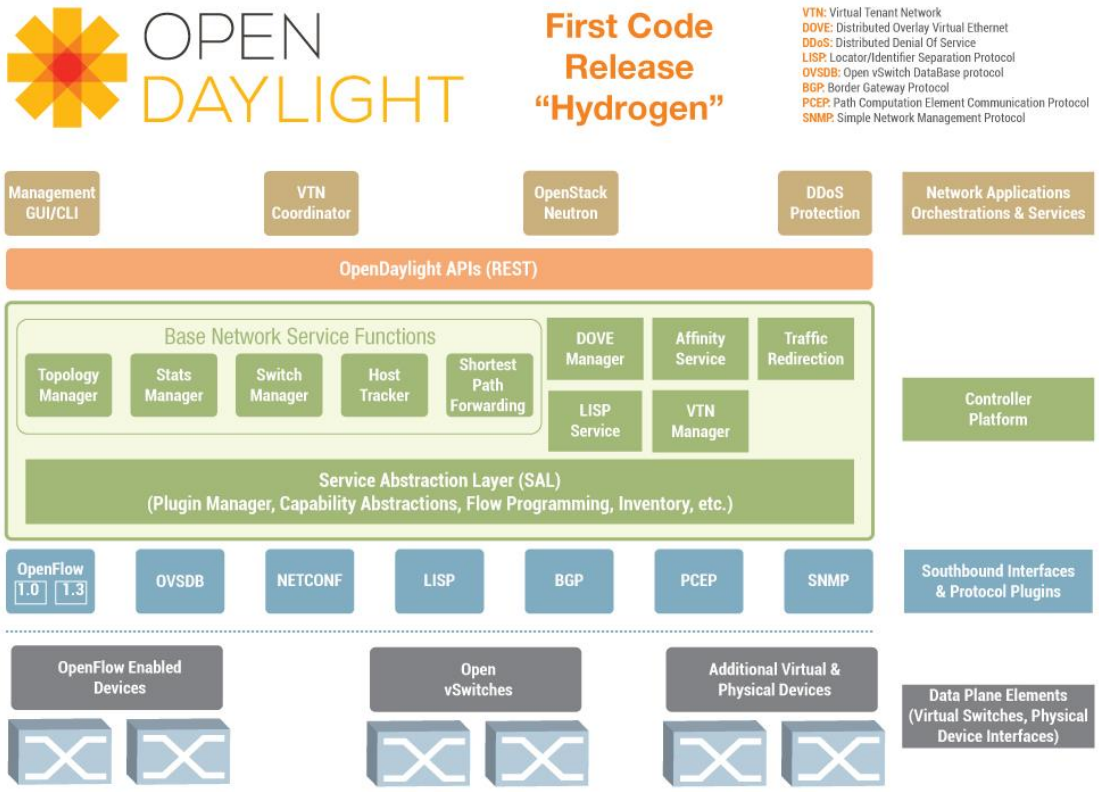


Figura 8 - Estrutura do *Hydrogen* (SDxCentral, 2014)

Relativamente à ligação entre o *Control Plane* e o *Data plane*, ou entre o controlador e os *switches*, esta é sempre feita recorrendo a um protocolo de comunicação para SDN como, por exemplo, o *OpenFlow*, e a uma API (Stallings, 2013). Este protocolo costuma ser utilizado por abordagens de diferentes empresas, como é o caso da Google, que utiliza exclusivamente para a função o *OpenFlow* (Pepelnjak, 2012). Este protocolo funciona muito bem em conjunto com o SDN por, de certo modo, se abstrair do *hardware* e de outros elementos utilizados na rede (Gouveia, Aparício, Parreira, Sargento, & Carapinha, 2013).

O *OpenFlow* também é utilizado por permitir ao controlador programar os *switches*, recorrendo a regras de encaminhamento constituídas por um descritor de um conjunto de pacotes e pelas ações a aplicar aos pacotes. Serve, também, para organizar o fluxo de tráfego (Pinote & Martins, 2013).

Existem já trabalhos publicados que têm em atenção a Qualidade de Serviço (QoS) associada a soluções de gestão de SDN. Estes trabalhos referem ter poucas perdas de QoS e utilizam a virtualização para os fundamentar. É importante também notar que estes trabalhos utilizam o protocolo *OpenFlow* (Pinote & Martins, 2013).

Tudo isto faz com que o SDN seja flexível e com que os *switches* e a camada de controladores SDN possam ser implementados em *switches Ethernet* (layer 2), *routers* (*layer* 3), camada de transporte (*layer* 4) ou mesmo camada de aplicação (*layer* 7) (Stallings, 2013).

No final da estrutura lógica do SDN aparece o *Data plane*, que é o conjunto de todos os dispositivos relativos diretamente à rede como, por exemplo, *routers* e *switches*.

Na secção seguinte estão mencionadas e explicadas algumas tecnologias associadas ao conceito do SDN.

2.2.1.2 Tecnologias

Neste ponto serão apresentados alguns dos elementos constituintes do SDN. Na Figura 9, que se baseia em informação recolhida num seminário da Cisco Systems, Inc. (Cisco Systems, Inc., 2014), está representado o enquadramento de algumas tecnologias associadas ao SDN.

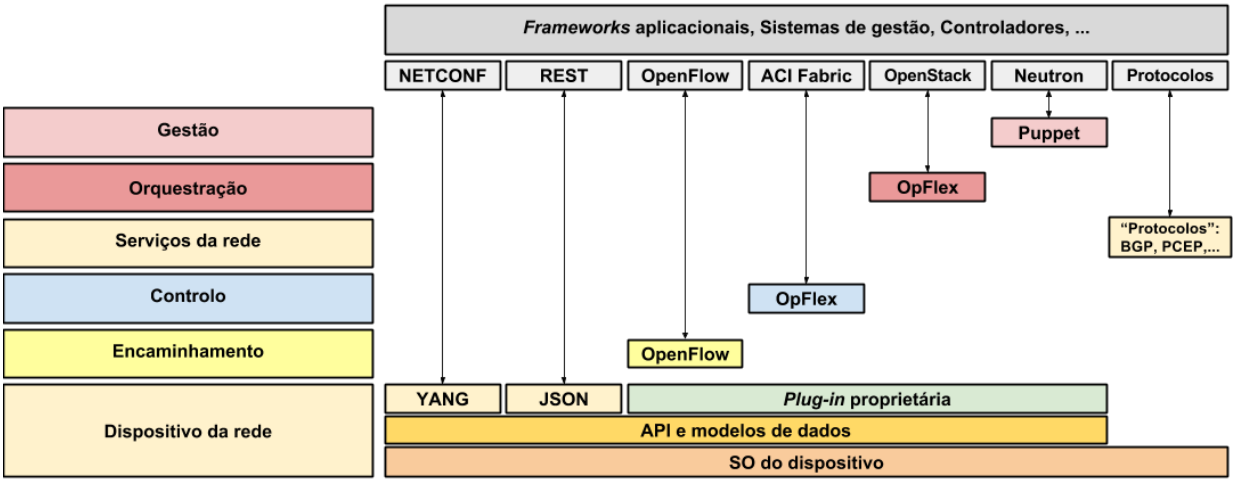


Figura 9 - Enquadramento de algumas tecnologias utilizadas pelo SDN

Na Figura 9, é importante notar que o *YANG* e o JSON estão no patamar de API e modelos de dados a serem utilizados sobre os dispositivos de rede (Cisco Systems, Inc., 2014). Ainda na mesma figura, é referido que, de acordo com a solução apresentada, o *OpenFlow*, o *OpFlex*, o *Neutron* e o *Puppet* estão associados a um *plug-in* proprietário (Cisco Systems, Inc., 2014).

Segue-se agora a apresentação de algumas tecnologias presentes na Figura 9 e que estão relacionadas com o conceito de SDN.

NETCONF

O *Network Configuration Protocol* (NETCONF), genericamente, serve para fazer a gestão de configurações em dispositivos de rede e é baseado na codificação em *eXtensible Markup Language* (XML) (Enns, Bjorklund, Schoenwaelder, & Bierman, 2011). Este protocolo define operações básicas que são equivalentes a comandos a serem executados num terminal ou numa de linha de comandos (CLI). Como este protocolo suporta a utilização de API, um dispositivo pode ser configurado mais facilmente através, por exemplo, de uma aplicação de gestão, reduzindo os custos de implementação e permitindo acessos temporários para criação de novas funcionalidades. O NETCONF utiliza o paradigma de *Remote Procedure Call* (RPC), fazendo com que através de um servidor, utilizando uma ligação segura, se troquem informações XML, para posteriormente os equipamentos ficarem corretamente configurados. O protocolo NETCONF pode-se tornar um importante pilar num sistema de configurações automatizadas (Enns, Bjorklund, Schoenwaelder, & Bierman, 2011).

Tal como no XML, o NETCONF também recorre a *tags*. Um dos fabricantes que utiliza NETCONF nos seus dispositivos é a Juniper Networks (Juniper Networks, Inc., 2015).

YANG

O *YANG* é uma linguagem de modelação de dados usada para o modelo de configuração e para a manipulação de estado dos dados. Esta linguagem é usada pelo protocolo NETCONF e está publicada na *Request for Comments* (RFC) 6020 de setembro de 2010. O *YANG* está relacionado com o conteúdo e operações nas camadas do NETCONF[4].

Tendo em conta a rápida adoção do NETCONF por parte da indústria, teve de se criar uma nova linguagem de modelação de dados porque as que já existiam, como o *Unified Modeling Language* (UML) e o XML *Scheme*, não eram direcionadas para a gestão de configurações. Por isso, os investigadores criaram o *YANG*, que é uma linguagem de fácil leitura e compreensão para os utilizadores, com hierarquia na configuração do modelo de dados e com suporte ao desenvolvimento, entre outras caraterísticas[4].

Aproveitando todas essas caraterísticas positivas, a *Open Networking Foundation* (ONF) estabeleceu o *YANG* como linguagem de modelação de dados para as configurações de dispositivos que suportem *OpenFlow*, logo, para utilização no SDN. Esta aposta foi feita a

[4] What is YANG? <http://www.tail-f.com/education/what-is-yang/>

pensar no desenvolvimento rápido para o mercado e para a criação de ferramentas que automatizem o desenvolvimento de soluções para a *cloud* (Ferro, 2012).

RESTful

O *REST* é baseado em tecnologias que muitos programadores conhecem, particularmente o *HyperText Transfer Protocol* (HTTP), para transportar pedidos e respostas entre clientes e serviços e XML e JSON para descrever os parâmetros de um pedido e de uma resposta. Devido à sua simplicidade, o *REST* é muito utilizado e útil no SDN para a programação de fluxos na rede (Dürr, 2014).

A plataforma *OpenDaylight SDN Controller* é um dos projetos que utiliza a *REST API*. Com o *REST API* torna-se mais simples aceder à base de dados da rede, isto é, ter acesso a dados relativos à configuração do controlador, por exemplo entradas da tabela de fluxos estatísticos, ter acesso a dados para dinamicamente descobrir entidades da rede, como *routers* ou *switches* e, por fim, ter acesso a estatísticas e dados de registo (OpenDaylight, 2013).

Uma outra solução útil para a utilização de *REST* no SDN é implementar uma configuração *RESTful API* diretamente em *switches*, como mostra a Figura 10 (RESTful control of switches, 2013).

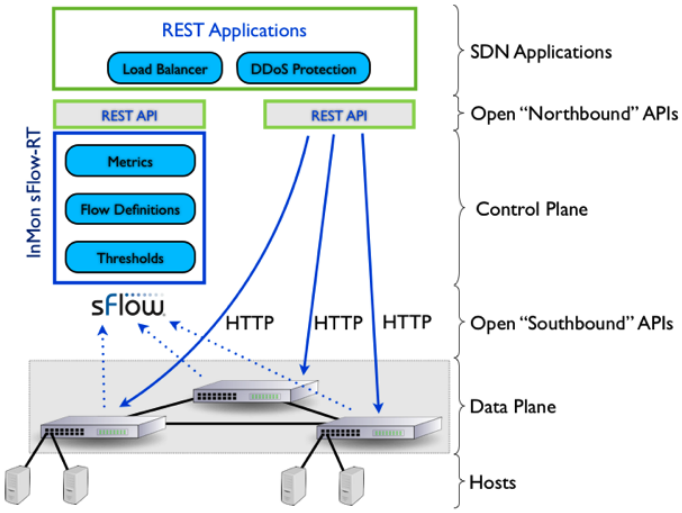


Figura 10 - Controlador de desempenho em SDN

De notar que esta solução não utiliza um controlador *OpenFlow* (RESTful control of switches, 2013), tecnologia esta que será apresentada de seguida.

OpenFlow

Neste ponto irá apresentar-se o *OpenFlow.*

O OpenFlow

No artigo original do *OpenFlow* (McKeown, et al., 2008) refere-se que este é um novo caminho para os investigadores testarem/correrem protocolos experimentais nas redes que usam diariamente. O *OpenFlow* é baseado num *switch Ethernet*, com uma tabela de fluxos interna e um *interface standard* para adicionar fluxos na tabela. O objetivo do *OpenFlow* é que ele possa ser utilizado seja qual for o fabricante dos dispositivos da rede.

Quando os investigadores escreveram o artigo do *OpenFlow* (McKeown, et al., 2008) já previam o aumento do tráfego de dados na rede, o que aumentaria a complexidade de configuração da mesma. Daí o surgimento da necessidade de haver redes programáveis. O início da programação de redes foi apelidada de "programação de *switches* e *routers*" que, recorrendo à virtualização, podiam processar pacotes simultaneamente para múltiplas redes isoladas.

Apesar das muitas dificuldades com que se depararam inicialmente como, por exemplo, o facto de existirem poucas plataformas de *software* que fossem ao encontro do que pretendiam e de as poucas que existiam não terem a performance que pretendiam, os investigadores idealizaram o *switch OpenFlow* (McKeown, et al., 2008) que está representado na Figura 11.

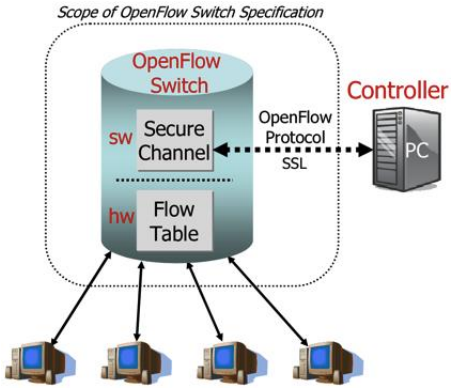


Figura 11 - Idealização do *switch OpenFlow*

Como se observa na Figura 11, alguns pontos, como o ponto central ou controlador (*Controller*), mantiveram-se desde o início até à atualidade. Também se mantém o facto de continuar a ser o protocolo *OpenFlow* quem faz a ponte entre esse controlador e o *switch*.

A ideia inicial dos investigadores foi desenvolver um *switch* que tivesse implementadas as funções do *OpenFlow.*

Na conclusão do artigo (McKeown, et al., 2008) os investigadores referem que, em caso de sucesso no desenvolvimento do *OpenFlow*, a sua ideia seria implementar, gradualmente, nos campos universitários este protocolo, criando "ilhas". De seguida, a comunicação entre universidades seria facilitada recorrendo a túneis e à sobreposição de redes.

O OpenFlow no SDN

Este protocolo de comunicação, que permite que os investigadores desenvolvam e testem protocolos, é o primeiro *interface* de comunicações *standard* entre as camadas de controlo e encaminhamento de uma estrutura SDN. O *OpenFlow* permite o acesso direto e a manipulação do plano de encaminhamento de dispositivos da rede, como *switches* e *routers*, quer eles sejam físicos ou virtuais (*hypervisor-based*). Em 2012 dizia-se que um protocolo como o *OpenFlow* era essencial para que fosse possível mover o controlo da rede para fora dos *switches* para que, assim, a lógica das redes estivesse centralizada, sendo esta controlada por *software* (Open Networking Foundation, 2012). O *OpenFlow* pode ser comparado às instruções dadas pelo *Central Processing Unit* (CPU), num computador.

O *OpenFlow* é implementado tanto nos dispositivos da infraestrutura da rede como no *software* de controlo do SDN. Até ao momento, este é o único protocolo *standard* para SDN que aplica manipulações diretas ao plano de encaminhamento dos dispositivos da rede. Este protocolo está a ser muito adotado por parte dos fabricantes que o estão a implementar ou através do *firmware* dos dispositivos ou recorrendo a *upgrades* de *software* nos mesmos.

Alguns dos benefícios do *OpenFlow* são (Open Networking Foundation, 2012):

- Controlo centralizado em ambientes com dispositivos de várias marcas;
- Redução da complexidade através da automação;
- Grande possibilidade de inovação;
- Aumento da confiança e da segurança na rede;
- Controlo mais granular ou detalhado da rede;
- Melhor escalabilidade para quem utiliza a rede.

A utilização de SDN em redes alargadas já é benéfico. Juntando ao SDN o *OpenFlow*, para o encaminhamento na rede, esta torna-se mais estável e bem definida.

O futuro das redes irá assentar cada vez mais no *software* como ferramenta de automatização na programação e configuração das redes.

Diretamente relacionado com o *OpenFlow* existe um projeto europeu intitulado *OpenFlow in Europe: Linking Infrastructure and Applications* (OFELIA). O projeto OFELIA tem como intuito disponibilizar aos investigadores uma ferramenta não só para testar a ferramenta desenvolvida numa rede, mas também controlar e alargar a própria rede de modo preciso e dinâmico.

O *software* de controlo de *framework* é gratuito de modo a dar ao utilizador/investigador a possibilidade de criar uma versão própria, apelidada de "ilha". O facto de este protocolo ser aberto leva a que possa haver comunicação entre diferentes "ilhas" criadas por outros utilizadores que também estão a testar este projeto (OFELIA, 2012).

O projeto OFELIA permite o desenvolvimento e teste de um sistema *Information Centric Networking* (ICN) (OFELIA, 2012).

O ICN é uma abordagem para fazer evoluir a infraestrutura da *Internet* introduzindo um nome único nos dados como um princípio base da *Internet*. Estes dados são independentes da localização e das aplicações e os benefícios da utilização do ICN são o aumento de eficiência e melhor escalabilidade, relativamente à informação/largura de banda e melhor robustez em cenários de comunicações concorrentes[5].

OpFlex

Atualmente os *data centers* crescem dinamicamente e espera-se que as redes se tornem ágeis e flexíveis sem comprometer a *performance*, a segurança, a escalabilidade e a estabilidade da rede (Cisco Systems, Inc., 2015).

Já se começa a notar que a virtualização das redes está a entrar nas soluções SDN, mas estas soluções aumentam de complexidade separando os domínios virtuais e físicos oferecendo, em contrapartida, pouca visibilidade entre ambos (Cisco Systems, Inc., 2015).

O *OpFlex* é um protocolo de policiamento pensado para troca de políticas de rede abstratas entre controladores da rede e dispositivos inteligentes para executarem essas mesmas políticas (Cisco Systems, Inc., 2014).

O *OpFlex* foi anunciado pela Cisco e publicado como *draft*, no *Internet Engineering Task Force* (IETF), no dia 2 de abril de 2014 (Smith, et al., 2014). Após a proposta da Cisco, a

[5] Information-Centric Networking Research Group ICNRG <https://irtf.org/icnrg>

empresa continua a trabalhar com a comunidade *Open Source* permitindo, assim, que ele possa ser utilizado por mais empresas. O objetivo da Cisco é fazer com que o *OpFlex* possa ser utilizado em *data centers*, na camada de acesso e em *Wide Area Network* (WAN) (Cisco Systems, Inc., 2014).

Acredita-se que com o *OpFlex* a inovação em produtos de redes independentes de fabricantes será bastante notada nos *data centers* e que as redes *cloud* se direcionam para a simplicidade, para o baixo custo e para o aumento de agilidade nas operações (Kiran, 2014).

Atualmente, para se gerir uma rede, tem que se descrever todas as configurações e passos para que um dispositivo comunique com outro. (Cisco Systems, Inc., 2015).

Num sistema de gestão de controlo declarativo, os objetos apenas devem indicar ao sistema controlador o seu estado para, em caso de alterações, o controlador efetuar as devidas correções na rede. Assim, reduz-se a carga e a complexidade do sistema de controlo, permitindo uma maior escalabilidade (Cisco Systems, Inc., 2015) (Avramov & Portolani, 2015).

O facto de o *OpFlex* poder ser aplicado em ambientes de vários fabricantes, de traduzir e definir um mapa de políticas nas estruturas e de poder ser aplicado em *switches*, *routers* e das camadas 4 à 7 em serviços de redes são alguns dos pontos positivos da sua utilização. (Cisco Systems, Inc., 2015).

O *OpFlex* permite a troca de dados entre dispositivos de gestão e, além de suportar XML e JSON, usa mecanismos *standard* de RPC como o JSON-RPC sobre TCP. Para que a comunicação se torne mais segura, ela deve ser feita através de *Secure Socket Layer* (SSL) e *Transport Security Layer* (TSL) (Cisco Systems, Inc., 2015).

À primeira vista, o *OpFlex* pode ter semelhanças com o *OpenFlow*, mas o *OpFlex* tem outras capacidades. Por exemplo, enquanto o *OpenFlow* se centra nas funções de controlo da rede, o *OpFlex*, segundo os seus criadores, foca-se mais nas políticas evitando, assim, o afunilamento da rede. As principais características do *OpFlex* são a capacidade de ultrapassar as adversidades, a disponibilidade e a escalabilidade. Esta última característica justifica-se pelo facto de utilizar protocolos de rede já existentes.

Uma das organizações que têm projetos relacionados com o *OpFlex* é a OpenDaylight, cujo objetivo é disponibilizar uma API para policiamento-base e que sirva de modelo *standard* a implementações *OpFlex* (SDxCentral, 2014).

Empresas como a Microsoft, IBM e Citrix são algumas das envolvidas ativamente no processo de tornar o *OpFlex* um *standard* (SDxCentral, 2014).

Em suma, a Cisco, juntamente com outras empresas e organizações, pretende que o *OpFlex* seja livre para que, assim, este utilitário possa ser disponibilizado e reutilizado em qualquer plataforma, assumindo que o mapeamento apropriado esteja criado. O *OpFlex* disponibiliza uma ferramenta poderosa de gestão na infraestrutura recorrendo a controlos declarativos.

Neutron

O *Neutron* é um projeto da OpenStack que disponibiliza *Networking-as-a-Service* (NaaS) entre dispositivos de *interface* geridos por serviços OpenStack. O *Neutron* veio substituir o *Quantum*, que era a API de rede original. Este projeto é mais direcionado para ambientes *cloud*[6] (SDxCentral, 2014).

Com o aumento da complexidade da rede e com o aparecimento de novas tecnologias, incluindo o SDN e NFV, a topologia e endereçamento também se foi complicando, o que dificultou o avanço no desenvolvimento de serviços para as redes (SDxCentral, 2014).

O *Neutron* é o componente que possibilita a virtualização de redes, no OpenStack.

Referiu-se que o *Neutron* era mais direcionado para o ambiente *cloud*, porque permite libertar o *stress* que se dá na rede neste ambiente, facilitando o processo NaaS na *cloud*. O *Neutron* permite também criar múltiplas redes privadas e controlar o endereçamento IP. A utilização da API leva a que as organizações possam controlar melhor a segurança, as políticas, o QoS e a monitorização e também analisar erros que possam acontecer na rede, bem como desenvolver serviços de rede avançados como *firewalls*, deteção de intrusões ou *Virtual Private Network* (VPN).

Puppet

Quando se tem muitos servidores, configurá-los "à mão" será bastante aborrecido. Pode-se sempre recorrer a *shell scripts*, sendo essa uma solução útil, mas de certa maneira limitada. Para facilitar este processo de configuração surge o *Puppet* como uma opção mais consistente para gerir a configuração de servidores (Vervloesem, 2010).

[6] Neutron <https://wiki.openstack.org/wiki/Neutron>

O *Puppet* é um utilitário ou ferramenta *Open Source*, criada em 2005, para gerir configurações. Este utilitário pode correr tanto em Windows como em vários sistemas Unix, ajudando os administradores de sistemas a automatizar tarefas, fazendo assim com que poupem o tempo e a mente para campos que sejam mais úteis e mais valorizados[7].

O *Puppet* foi desenvolvido em *Ruby* e pode ser usado recorrendo à linguagem declarativa própria ou a *Ruby Domain-Specific Language* (DSL) (Bode, 2010). Recorrendo ao *Puppet*, a gestão pode ser feita tanto em servidores físicos como em virtuais. A informação relativa às configurações é armazenada nos ficheiros "*Puppet manifests*", que podem ser criados pelo próprio utilizador. É utilizado o paradigma cliente-servidor, usando um *REST* API para distribuir as configurações pretendidas pelos sistemas definidos, sendo esta comunicação feita pelo *Puppet* mestre e pelo agente *Puppet*[8] [9].

O *Puppet* cobre todos os passos, dos dispositivos até à orquestração, desde o início do desenvolvimento do código, passando pelos testes até que estes equipamentos sejam colocados em produção ou sejam lançadas atualizações. Isto faz com que haja uma melhor colaboração entre administradores de sistemas e programadores, permitindo que o código seja mais eficiente e limpo e mais bem estruturado assegurando, assim, que haja consistência, confiança e estabilidade[10].

Utilizando este utilitário existe completa abstração, dando a possibilidade aos administradores de descreverem a configuração recorrendo a termos de alto-nível como, por exemplo, *packages* ou *users*, isto sem usar comandos específicos dos sistemas operativos como, por exemplo, *apt* ou *yum*, havendo apenas a necessidade de ter conhecimentos básicos de programação[11] [12].

Todas as alterações que se pretenda efetuar na rede são feitas num sistema central, semelhante a um repositório central. Esse repositório envia as alterações aos vários servidores (Vervloesem, 2010).

---

[7] What is Puppet? <http://puppetlabs.com/puppet/what-is-puppet>

[8] The Puppet Language: Basics <https://docs.puppetlabs.com/puppet/latest/reference/lang_summary.html>

[9] HTTP API <https://docs.puppetlabs.com/guides/rest_api.html>

[10] What is Puppet? <http://puppetlabs.com/puppet/what-is-puppet>

[11] The Puppet Language: Resources <https://docs.puppetlabs.com/learning/ral.html>

[12] The Puppet RAL: An Introduction <http://somethingsinistral.net/blog/puppet-ral-an-introduction/>

Inicialmente, quando se instala um sistema operativo, caso ele seja configurado automaticamente, sabe-se com relativa facilidade as configurações mas, caso mais tarde seja necessário efetuar alterações e as configurações iniciais não sejam documentadas, este processo complica-se. O resultado é simples: tempo despendido e tentativa de correção de erros causados pelas alterações, o que pode provocar ainda mais erros.

Caso se concentre as configurações num repositório central como se utiliza no *Puppet*, quando se reinstala um sistema operativo basta recarregar as configurações do repositório e as mesmas ficam definidas. Com o *Puppet* é possível manter pacotes, ficheiros, serviços, utilizadores e grupos independentemente do sistema operativo (Red Hat, CentOS, Ubuntu LTS, Debian, Solaris, Microsoft Windows e Mac OS X, entre outros)[13].

Utilizar um gestor de configurações com um sistema de controlo de versões, como o *Git* ou o *Subversion*, pode ser muito útil para que as alterações fiquem registadas. Caso se pretenda reverter as configurações para um estado anterior, a utilização de um sistema de um controlo de versões também é muito útil (Vervloesem, 2010).

Assim que o *Puppet* é instalado, todos os nós (incluindo servidores físicos, máquinas virtuais ou dispositivos da rede) na sua infraestrutura ficam com um agente *Puppet* instalado.

Os utilizadores mais sonantes do Puppet são: AT&T, CERN, Cisco, Citrix Systems, Intel, McAfee, Motorola, NASA, PayPal, Sun Oracle, Symantec, Verizon e VMware[14].

2.2.2 Network Functions Virtualization

O NFV surgiu em sequência do SDN, não sendo dependente do mesmo (SDxCentral, 2014). Esta abordagem é aplicada à *Network Virtualization* (NV); por isso, apesar de não se referir sempre NV, é importante chamar a atenção que as funções do NFV são aplicadas às *Network Virtualization*.

O NFV vem tentar colmatar alguns pontos que os operadores julgam que o SDN não resolve. Por exemplo, enquanto o SDN tende a criar redes abstratas, para que mais rapidamente se possam criar inovações na mesma, o NFV, como se apresenta mais à frente, tem outros objetivos, como a redução do *CapEx* (capital financeiro despendido) e do *OpEx* (gastos

[13] Installing Puppet Enterprise: System Requirements and Pre-Installation <https://docs.puppetlabs.com/pe/latest/install_system_requirements.html>

[14] Customers <http://puppetlabs.com/about/customers>

materiais e em manutenção, por exemplo), que serão abordados ainda nesta secção, entre outros fatores (Pate, 2013).

De notar que, tal como referido anteriormente, enquanto o SDN foi desenvolvido por investigadores e arquitetos de *data centers* (Pate, 2013), o NFV foi criado a partir de um consórcio de empresas e fornecedores de serviços (*Service Providers*) (Pate, 2013) no *European Telecommunications Standards Institute* (ETSI) *Industry Specification Group* (ISG) *for Network Functions Virtualization* (NFV), entre elas, AT&T, Orange, Telecom Italia, Verizon, Alcatel-Lucent, Cisco Systems Belgium, Ericsson, Huawei, Juniper Networks, Vodafone Group Services e PT Portugal SGPS, SA[15] (Pate, 2013) (SDxCentral, 2013).

Segundo o artigo original do NFV (Chiosi, et al., 2012), ele informa que é necessário espaço e energia para ser lançado um novo serviço de rede, além de investimento de capital. Isto leva a que o *hardware* aplicado se torne rapidamente obsoleto (Pate, 2013).

O NFV ajuda a resolver os problemas anteriormente referidos levando para as redes a tecnologia de virtualização das Tecnologias de Informação, bastando para isso que haja equipamento nos *data centers*, nos nós das redes e nos utilizadores finais. O NFV pode ser aplicado a qualquer pacote do processo *Control plane* e às infraestruturas de redes móveis (SDxCentral, 2013).

Esta virtualização de redes oferece uma nova maneira de projetar, desenvolver e gerir os serviços da rede de modo virtual. Esses serviços podem ser o *Domain Name System* (DNS) e *Dynamic Host Configuration Protocol* (DHCP), por exemplo. O NFV divide as funções da rede do *hardware* proprietário para que essas funções (*Network Address Translation* (NAT), DNS e *firewalling*, entre outras) possam ser executadas através de *software*. O NFV, um pouco à semelhança do SDN, recorre à tecnologia de informação *standard*, ou seja, à virtualização, ajudando assim a melhorar e acelerar alguns serviços (SDxCentral, 2014) (SDxCentral, 2013).

Em suma, alguns dos benefícios do NFV são (SDxCentral, 2013):

- Reduzir o *CapEx* – reduzir o capital necessário para o investimento numa nova plataforma;

[15] Network Functions Virtualisation: List of members <https://portal.etsi.org/TBSiteMap/NFV/NFVMembership.aspx>

- Reduzir o *OpEx* – reduzir o espaço, energia, custos de operação e de manutenção e requisitos de refrigeração para equipamentos físicos podendo, assim, dar lugar a mais máquinas virtuais;
- Agilizar e flexibilizar a implementação desta arquitetura de modo a responder à procura por parte das empresas/clientes.

2.2.3 SDN vs NFV

O *Software-Defined Networking* e o *Network Functions Virtualization* são abordagens complementares porque cada uma oferece um caminho para projetar e gerir a rede e respetivos serviços (SDxCentral, 2014).

O SDN separa o plano de controlo (cérebro) do plano de encaminhamento (músculo) da rede e foca a sua atenção na distribuição da rede, de modo a que ela seja mais eficiente e automatizada (SDxCentral, 2014).

Já o NFV desassocia várias soluções de rede como, por exemplo, a obrigatoriedade/dependência que o DNS tem de *hardware*; assim, as soluções da rede podem ser executadas através de *software* de modo a que possam ser melhoradas e acelerados alguns dos seus serviços.

Apesar das diferenças existentes nas duas abordagens (SDN e NFV), ambas têm pontos em comum (SDxCentral, 2014):

- Mover funcionalidades para *software*;
- Utilizar servidores e *switches* sobre aplicações proprietárias;
- Utilizar API;
- Suportar uma maior eficiência na orquestração, virtualização e automatização dos serviços da rede.

Em suma, o SDN e o NFV funcionam bem em conjunto mas não são dependentes um do outro. O SDN contribui para a automatização, que possibilita políticas de decisão para indicar o melhor caminho ao tráfego da rede, enquanto o NFV se foca mais nos serviços, apesar de não ser possível a virtualização de todos os serviços.

Os avanços de toda a tecnologia são a chave para envolver a rede e os dispositivos das pessoas e para os manter ligados entre si (SDxCentral, 2014).

No ponto seguinte serão apresentadas as tendências do SDN e do NFV para os próximos anos.

2.2.3.1 Tendências

Neste tópico serão apresentadas as tendências do SDN e do NFV.

Previa-se que no ano de 2014 se chegasse a algumas conclusões concretas ou definitivas sobre alguns assuntos tecnológicos relacionados com o SDN, NFV e até com "*Internet* das Coisas", mas tal não se verificou (Alexander, 2014). Verificou-se sim, a continuação da discussão e dos debates destes assuntos.

Na área da virtualização de redes, tanto as *cloud service providers*, como os *data centers* de empresas são quem está a indicar o caminho. Empresas como a Amazon, na área de serviços em *data centers*, e como o Facebook, que necessita de computação em larga escala, fazem com que a virtualização tivesse sido um tema importante em 2014 e nos anos seguintes (Palmer, 2014).

A otimização de custos é outro ponto importante e o facto de a VMware disponibilizar uma solução própria de SDN, o VMware *NSX*, leva a que muitas empresas a apliquem nas suas infraestruturas, isto porque ela também utiliza virtualização de redes (Palmer, 2014).

A própria Cisco tem vindo a alterar a ideia que tinha há uns tempos e já está a apostar um pouco na virtualização, visto ter adquirido duas empresas (Cisco Systems, Inc., 2013) (Cisco Systems, Inc., 2014) que estão relacionadas com a virtualização: Insieme Networks (possível concorrente do VMware *NSX*) e a Tail-f (para aumentar o portefólio da *Cisco Cloud Virtualization*).

Se o ano de 2012 foi de abordagem inicial mais séria sobre o que era o SDN e o ano de 2013 foi já para compreensão do SDN, o ano de 2014 era previsto ser o da prova de conceito (Palmer, 2014), mas isso não significa que esta prova de conceito não avance nos anos seguintes. Os clientes do SDN e do NFV querem reduzir o número de elementos de segurança física mas mantendo, simultaneamente, o mesmo nível de segurança na rede virtual que existia na rede física (Palmer, 2014).

As abordagens SDN e NFV terão de ser úteis tanto para empresas e operadoras de maior dimensão (Google, Facebook, Amazon, Microsoft, Yahoo, AT&T, Verizon, entre outras) como para empresas de uma dimensão mais reduzida do que as referidas (Palmer, 2014).

Um estudo da Infonetics Research intitulado "*2014 SDN Stategies: North American Enterprise*" (Grossner, 2014) apresenta um dado importante: em 2016, 87% das empresas norte-americanas pretendem já ter soluções SDN implementadas nas suas infraestruturas. Steve Alexander (Alexander, 2014), *Chief Technical Officer* (CTO) da Ciena Corporation, previa que, em 2015, se começaria a ver as primeiras implementações de SDN em redes de telecomunicações um pouco por todo o mundo. Ainda segundo Alexander, as discussões havidas no ano de 2014 seriam semelhantes às discussões de 2015, mas tendo por base o NFV em vez do SDN, ou seja, em 2015 ter-se-á começado a analisar as vantagens de substituir funções de *hardware* por funções virtualizadas (Alexander, 2014).

Para apoiar a ideia de que 2015 tenha sido o *mainstream* do SDN, ou seja, o ano zero do SDN, o relatório "*Carrier SDN and NFV Hardware and Software Market Size and Forecast*" (Howard, 2014) prevê que, em 2018, os mercados de SDN e NFV atinjam valores na casa dos 11 mil milhões de dólares em todo o mundo (Alexander, 2014).

Assim se vê que o SDN e o NFV são um mercado em expansão, com futuro e bastante atrativo.

Na secção seguinte irão ser listadas as soluções de algumas empresas que estão a trabalhar no SDN.

2.3 Soluções SDN de algumas empresas

De seguida serão referidas algumas soluções ou propostas SDN de algumas empresas e depois serão explicadas as soluções de sete empresas. Apesar de se referir soluções SDN, pela pesquisa efetuada conclui-se que algumas delas são bastante direcionadas à virtualização.

A gigante das redes de informática, Cisco Systems, Inc., tem como solução SDN a "*Cisco Network Service Orchestrator enabled by Tail-f*" (NSO), que estava previsto ser lançada comercialmente no final de 2015 (Cisco Systems, Inc., 2015). (Cisco Systems, Inc., Tail-f Systems, 2015). Já a Huawei apelidou a sua solução de *SoftCOM* (Huawei Technologies Co., Ltd., 2013). A Nuage Networks, empresa interna da Alcatel-Lucent, tem como abordagem de SDN a Nuage Networks *Virtualized Services Platform* (VSP)[16] (Nuage Networks, 2014). Esta solução pode ter algum interesse para o panorama das redes em Portugal, visto a Alcatel-Lucent ser uma parceira da Altice, antigamente nomeada de PT (Casa dos Bits,

[16] Nuage Networks: Products <http://www.nuagenetworks.net/products/>

2010)(Portugal Telecom, 2010). A empresa VMware, Inc., conhecida por disponibilizar *software* de virtualização, em parceria com a HP disponibiliza uma solução naturalmente mais direcionada para a virtualização. Esta solução chama-se VMware *NSX* [17]. Nesta parceria, a VMware aplica os conhecimentos que tem a nível da virtualização e a HP aplica agilidade empresarial, monitorização, *troubleshooting* e gestão num único ponto e centralizado no que se refere ao controlo da rede (Shaikh, 2013).

Todas as soluções SDN referidas até aqui neste subtópico serão analisadas nas secções seguintes.

Outras empresas, apesar de não terem soluções próprias, estão a trabalhar em conjunto em soluções SDN com outras empresas como é o caso da IBM, que apresenta no seu *site* como parceiras de soluções SDN a Citrix, Juniper Networks, Palo Alto Networks, Plexxi, entre outras. Mais algumas soluções SDN são das seguintes empresas:

- Juniper Networks com a solução Contrail[18];
- Big Switch Networks que apelidou a sua proposta de *Big Network Controller*[19];
- IBM que, apesar de não ter uma solução própria, tem parcerias com diversas empresas tais como Citrix, Juniper Networks, Palo Alto Networks e Plexxi (IBM, 2014), onde disponibiliza uma arquitetura SDN padrão. Esta abordagem da IBM leva os benefícios das redes virtuais ao ambiente *cloud* (IBM, 2014);
- Arista EOS que tem como solução a *Arista Software Driven Cloud Networking* (SDCN)[20];
- Citrix com a *Citrix NetScaler*. De referir que esta solução se integra com a Cisco ACI (Cisco Systems, Inc., 2013);
- Google com o *Andromeda*. O *Andromeda* é a solução SDN da Google para os serviços de virtualização de redes da empresa (Vahdat, 2014). Ainda segundo Amin Vahdat, engenheiro distinguido pela Google, a empresa beneficia por utilizar programação de redes em toda a sua rede, desde o nível mais baixo de *hardware*, até ao nível mais

[17] VMware Products: NSX <http://www.vmware.com/products/nsx>

[18] Contrail <http://www.juniper.net/us/en/products-services/sdn/contrail/>

[19] Big Network Controller <http://bigswitch.com/products/SDN-Controller>

[20] Software Driven Cloud Networking <http://www.arista.com/en/products/software-driven-cloud-networking>

elevado de elementos de *software*. O *Andromeda* é a base da distribuição dos serviços de redes da Google, o *Cloud Platform*, com alta performance, disponibilidade, isolamento e segurança. Através da API *Andromeda* é possível implementar regras de *firewall*, *routing* e encaminhamento na rede. É também importante referir que a solução *Andromeda* utiliza *OpenFlow*, tal como muitas soluções SDN, mas Vahdat refere que o SDN não requer obrigatoriamente o *OpenFlow* (Higginbotham, 2014). Atualmente, esta solução da Google é apenas utilizada internamente;

- Nos últimos anos, o Facebook também tem estado a investir para o SDN[21], nomeadamente com a fundação do *Open Compute Project* (OCP), um esforço colaborativo da indústria para acomodar *hardware* tanto na computação como no armazenamento e nas áreas de rede de fornecedores de serviços, fazendo assim com que este *hardware* possa ser mais programável através de *software*. Este projeto é mais direcionado para *data centers* (Swan, 2014). Estes esforços têm feito com que a empresa tenha poupado bastantes recursos, direta ou indiretamente, a todos os níveis[23].

No Anexo 1, estão listadas algumas das mais relevantes propostas SDN de empresas de referência, na área das redes informáticas.

2.3.1 Características de algumas soluções

Neste tópico serão apresentadas as pesquisas efetuadas relativamente às soluções SDN de sete empresas, 5 fabricantes de soluções e conceituadas empresas na área das redes Informáticas e as duas últimas operadoras de telecomunicações: Cisco Sytems, Inc., Huawei, Alcatel-Lucent (Nuage Networks), HP, VMware, Inc., Telecom Italia e PT Inovação, atualmente denominada de Altice Labs.

2.3.1.1 Cisco Application Centric Infrastructure

A *Cisco Application Centric Infrastructure* (ACI) é uma solução SDN direcionada para redes *cloud* e *data center*.

A Cisco ACI consiste num novo conceito de uma arquitetura de *data centers* de acordo com a rede tradicional hoje existente (Cisco Systems, Inc., 2014).

[21] About OCP <http://www.opencompute.org/about/>

Esta solução SDN da Cisco é, por isso, baseada numa combinação de protocolos tradicionais e novos desenvolvimentos centralizados na gestão, programação e automatização de tarefas ou processos.

A centralização cria uma alta disponibilidade, flexibilidade, resiliência e, por fim, alta velocidade em infraestrutura de uma rede com alta velocidade, caso haja interligação entre todas as camadas.

Cisco ACI Fabric

A *Cisco ACI Fabric*, resultado em produção da *Cisco ACI*, assenta em 3 componentes principais sendo um deles a *Application Policy Infrastructure Controller* (APIC), que será abordada a seguir.

Explicando um dos conceitos referidos na frase anterior, o *Cisco ACI Fabric* disponibiliza uma funcionalidade de autodescoberta de IP na rede, caso se adicione um dispositivo, seja ele real ou virtual, e se pretenda que ele se integre na rede (Citrix, 2015).

A *Cisco APIC* disponibiliza uma gestão centralizada, programável, automatizada e simplificada.

A configuração da *Cisco ACI Fabric* é partilhada e armazenada em múltiplas cópias ao longo do *cluster* APIC, podendo assim fazer-se restauro de alguma configuração, caso se pretenda.

Na Figura 12, apresenta-se a estrutura e integração da *Cisco APIC* na *Cisco ACI Fabric* (Cisco Systems, Inc., 2014).

Nessa figura é também possível notar-se a existência das três camadas fundamentais do SDN que, como já foi referido, são:

- *Application plane* – no topo da figura, onde são apresentados, entre outros, a gestão do sistema e de armazenamento;
- *Control plane* – a meio da figura, realçado pelo APIC;
- *Data plane* – na zona inferior da figura, representado por *routers*, entre outros.

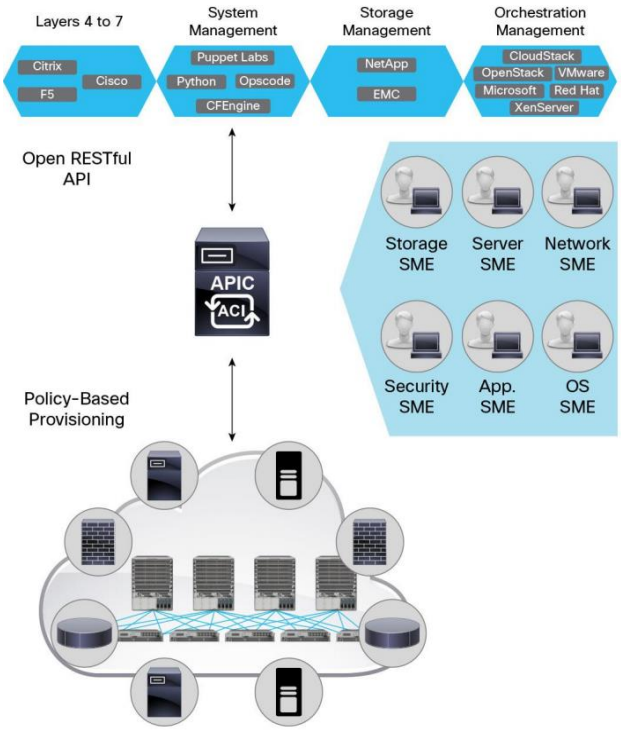


Figura 12 - *Cisco APIC* na *Cisco ACI Fabric*

Na Figura 12 surgem alguns conceitos que serão aprofundados no decorrer desta dissertação e que serão importantes como, por exemplo, a gestão de orquestração (*orchestration management*).

Os principais objetivos da *Cisco ACI Fabric* são (Cisco Systems, Inc., 2014):

- Escalabilidade: de modo a tornar a rede mais escalável, todos os dispositivos estão ligado à *Fabric*. Com os nós diretamente ligados a um ponto, a escalabilidade da rede simplifica-se. O administrador da rede apenas tem de adicionar nós de ligação à *Fabric*. Assim, a flexibilidade da rede faz com que a mesma comece com pequenas dimensões mas possa vir a aumentar gradualmente até uma escalabilidade de um milhão de *endpoints*, ou seja, um milhão de rotas ligadas a um milhão de elementos (Citrix, 2015);
- Extensibilidade: a *Cisco ACI Fabric* é altamente extensível. O administrador da *Fabric* pode integrar tanto redes virtuais, como serviços, desde a camada 4 (Transporte) até à camada 7 (Aplicação), do modelo OSI como, por exemplo, *firewalls* e balanceamento de carga;
- Simplicidade: o importante é que exista conetividade em todo o lado. A *Cisco ACI Fabric* foi desenvolvida tendo por base a sua utilização em *data centers*. Por isso, só o

que é realmente importante é que foi mantido/aproveitado para esta solução SDN da Cisco, evitando assim que a rede se torne complexa;

- Flexibilidade: a *Cisco ACI Fabric* tem capacidade, nativamente, para permitir que o utilizador acrescente ou ligue um *host* em qualquer sítio da *Fabric*. A *Cisco ACI Fabric* pode normalizar múltiplos tipos de encapsulamento por parte dos *hosts* e respetivos *hypervisors*. Posto isto, a *Cisco ACI* suporta, tal como já mencionado anteriormente, tanto dispositivos físicos como virtuais;
- Eficiência: um benefício da utilização da *Cisco ACI Fabric* é que todos os *hosts* estão exatamente a 2 saltos de distância uns dos outros na *Fabric*. Devido a este facto, para a transação de muitos dados, a *Cisco ACI Fabric* dispõe de baixa latência. Isto faz com que seja eficiente a troca de informações entre *hosts* nas aplicações *data centers*.

2.3.1.2 Huawei SoftCOM

Segundo o documento da Huawei (Huawei Technologies Co., Ltd., 2013), atualmente as empresas e operadoras de telecomunicações têm de arranjar soluções para resolver o aumento de largura de banda provocado pelo cada vez maior número de utilizadores e dispositivos e que leva a um crescimento considerável do tráfego.

A solução SDN proposta pela Huawei é o *Huawei SoftCOM* (Huawei Technologies Co., Ltd., 2013), baseado na *cloud*.

Esta solução assenta em quatro pontos:

- *Equipment-Level Cloud-Lization*: divisão entre *hardware* e *software*. A divisão entre o *hardware* e as funções abstratas leva a que os dispositivos de rede não dependam de *hardware* específico, ou seja, os dispositivos irão partilhar uma plataforma unificada, partilhando também a flexibilidade e as funções definidas por *software*;
- *Network-Level Cloud-Lization*: divisão entre encaminhamento e controlador. Dividindo o plano de controlo do plano de encaminhamento, isto é, ignorando os recursos da rede, a gestão desta será feita a partir de um nível superior, mais superficial, simples e com maior eficiência. Isto torna a arquitetura da rede mais inteligente e mais ágil a transformações;
- *IT System Cloud-Lization*: transformação de infraestrutura de IT tradicional para a infraestrutura baseada na *cloud*. A infraestrutura IT atual deve ser movida para uma

plataforma na *cloud*, levando a um aumento da capacidade a baixo custo devido à distribuição de armazenamento e escalabilidade;

- *Internetized Operation*: transformação de sistemas orientados a telecomunicações para sistemas baseados na *Internet*. A gestão das operações, feita pelas operadoras, deve ser mais orientada para a *Internet*. Esta mudança deve ser paralela à transformação da infraestrutura IT tradicional para a baseada na *cloud* (*IT System Cloud-Lization*). Esta transformação deve ser visível para as empresas através da disponibilização de uma *framework web* que permite *Business Intelligence* da análise de muitos dados, isto é, todos os serviços de clientes, bem como aplicações para diversos dispositivos e plataformas, serão disponibilizados *online*.

Segundo a empresa, a utilização do *SoftCOM* leva à redução de custos e à criação de novos modelos de negócio inovadores para as operadoras. Isto está diretamente relacionado com o *OpEx* e *CapEx*, ou seja, gastos operacionais e gastos capitais.

No futuro, a Huawei prevê que se faça diversas modificações tanto na rede como na solução SDN que propõe. Algumas dessas modificações poderão ser a divisão entre o plano de controlo e o plano de encaminhamento, serviços de rede que passarão a ser definidos por *software* e serviços de operadoras, como o *IP Multimedia Subsystem* (IMS) e o *Internet Protocol Television* (IPTv), que serão movidos para a *cloud* (Huawei Technologies Co., Ltd., 2013).

Para finalizar, a ideia da Huawei é aplicar os conceitos de *cloud* e virtualização às redes, de maneira a que elas se adaptem ao futuro (Huawei Technologies Co., Ltd., 2013).

2.3.1.3 Virtualized Services Platform

A Nuage Networks é uma empresa de redes, *startup* interna fundada pela Alcatel-Lucent (Dignan, 2013), que anunciou a sua solução SDN apelidando-a de *Virtualized Services Platform* (VSP) (Dignan, 2013).

O VSP é um pacote de virtualização de redes que surge como alternativa a soluções de outras empresas, nomeadamente a VMware (*VMware NSX*) e a Cisco (*Cisco ACI*) e consiste em três elementos ou funções-chave, que serão explicados de seguida (Dignan, 2013) (Ferro, 2013):

- *Virtualized Services Directory* (VSD) - é o motor de policiamento de toda a plataforma. Serve para criar *templates* que definem o funcionamento da rede para

determinados recursos e utilizadores. Quando uma nova máquina virtual ou aplicação é criada, ela associa-se a um *template* que disponibiliza os recursos necessários na rede;

- *Virtualized Services Controller* (VSC) - comunica com outro elemento principal e *software* de distribuição que é o *Virtual Routing and Switching* (VRS). Programa a rede através de *OpenFlow*;
- *Virtual Routing and Switching* (VRS) - é o agente da rede que se instala no *hypervisor*[22], ou hipervisor, para o *software* da rede. É uma versão do *Open vSwitch Database Management Protocol* (OVSDB), da VMware, Inc. (Rouse, 2013), que a empresa diz integrar com vários hipervisores como o *ESXi* - virtualizados de computadores -, por exemplo. A Nuage Networks utiliza o *Virtual Extensible LAN* (VXLAN) para ligar os agentes VRS à *Local Area Network* (LAN) do *data center*. A arquitetura de *tunneling* da Nuage contrasta com soluções de outras empresas, como Cisco, NEC e Big Switch, onde o controlador programa diretamente nos *switches*.

Na Figura 13, baseada na apresentada pela Nuage Networks (Nuage Networks, 2014), estão sintetizados os três pontos principais da solução VSP da Nuage Networks.

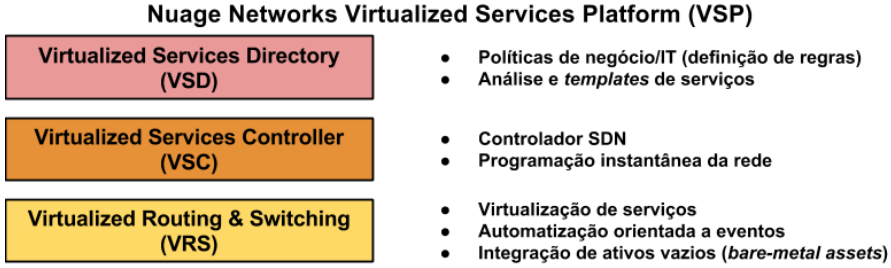


Figura 13 - Pontos principais dos elementos do VSP

A Nuage informa que são necessárias poucas máquinas virtuais para implementar a solução: duas para as aplicações VSD e VSC e, pelo menos, mais uma para os agentes VRS (Ferro, 2013). Estes 3 pontos estão referidos na Figura 13.

Tal como outras soluções *Open vSwitch*, por exemplo a *NSX* da VMware, os serviços de rede estão localizados ou no *hypervisor* ou no sistema operativo. A Nuage está a implementar um conceito de distribuição de *layer* 3 através de túneis, ou seja, cada VRS tem um túnel que se liga a outros agentes VRS na rede.

[22] *Hypervisor*, ou gestor de máquina virtual, é um *software* que gere vários sistemas operativos num único equipamento. < http://www.webopedia.com/TERM/H/hypervisor.html >

Relativamente ao policiamento no VSD, ele é baseado no que já é usado hoje nos *routers Digital Subscriber Line* (DSL) e *Long-Term Evolution* (LTE) da Alcatel-Lucent, ou seja, é auto instanciado e o controlo de acesso aos mecanismos é refinado (Ferro, 2013) (Nuage Networks, 2013).

Em relação às configurações, em vez de elas serem movidas entre agentes VRS, o VSD pode receber *triggers*, modificações no *template* e reajustar as configurações VRS antes das entregas do VSC, modificando os fluxos para o VRS.

A Nuage refere que pode dar suporte a vários *data centers* através de uma integração muito próxima do WAN, pelo facto de os dados sobrepostos na rede passarem diretamente para esta rede de longa distância.

Segundo consta (Ferro, 2013), veem-se muitas semelhanças entre a solução da Alcatel-Lucent (Nuage Networks), da VMware e da Juniper Networks (*Contrail Networks*), mas a solução da Nuage destaca-se pelas diversas funcionalidades que a sua solução traz e mostra-se mais direcionada a uma rede de grandes dimensões.

No entanto, será difícil vender a sua solução SDN a empresas, apesar de a Nuage já ter um largo segmento de requisitos de rede no *core* do SDN (Ferro, 2013).

2.3.1.4 HP SDN

A HP é outra das gigantes da informática que também desenvolveu a sua própria solução de SDN. A HP congratula-se por ser a primeira empresa a apresentar uma loja de aplicações exclusivamente para o SDN[23]. Na sua loja, apresenta diversas soluções SDN, desde aplicações que permitem a avaliação automática da rede e segurança em tempo real até aplicações que tratam automaticamente de políticas e qualidade de serviço da rede[24]. Esta aposta nas aplicações parece fazer sentido, pois um relatório apresentado pela IDC, e referido pela HP (Mehra & Casemore, 2013), indica que, em 2016, dos 3,7 mil milhões de dólares que prevê que o SDN valha, 670 milhões virão de aplicações de SDN.

Relativamente à sua solução propriamente dita, esta intitula-se *HP SDN*.

[23] SDN Dev Center <http://www8.hp.com/us/en/networking/sdn/devcenter-index.html>

[24] Introducing HPE SDN App Store <https://hpn.hpwsportal.com/catalog.html#/Home/Show>

Algumas características da *HP SDN* são utilizar o *OpenFlow* e ser um sistema de código aberto através da disponibilização de um *kit* de desenvolvimento (SDK) para o SDN, por parte da HP. Através da *App Store* e do SDK, em conjunto ainda com a comunidade de desenvolvedores, a HP desenvolve a sua solução SDN no seu próprio ecossistema, tal como se pode observar na Figura 14[25].

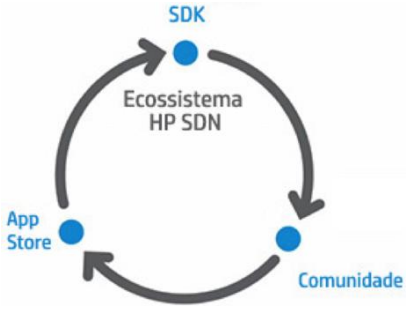


Figura 14 - Ecossistema *HP SDN*

Aprofundando um pouco mais a Figura 14, em relação à arquitetura SDN, a solução da HP assenta em quatro pontos essenciais: infraestrutura, controlo, aplicações e gestão. Na Figura 15 (Hewlett-Packard Development Company, L.P., 2013) pode-se observar a arquitetura SDN, da solução da HP.

Figura 15 - Arquitetura SDN da HP

Como se pode analisar pela Figura 15, na infraestrutura estão todos os equipamentos necessários que a solução da HP suporta e, se se observar bem, o *OpenFlow* será utilizado nestes mesmos equipamentos como protocolo *standard* da indústria e também utilizado na arquitetura genérica do SDN. Na camada acima da infraestrutura está o controlo que recorre a uma solução própria da HP – o controlador de SDN da HP apelidado de *Virtual Application Networks* (VAN).

[25] Software-defined Networking: Programmable network aligned to business applications delivers agility <http://h17007.www1.hp.com/br/pt/networking/solutions/technology/SDN/#tab=TAB1>

O *HP VAN SDN Controller Software* disponibiliza um ponto único de controlo, simplificando assim a gestão e a orquestração. A utilização deste controlador permite que os programadores desenvolvam soluções próprias, inovadoras e que se adaptem dinamicamente às empresas. O resultado destas soluções pode ser um programa em *Java* ou um controlo de *interfaces RESTful* (Hewlett-Packard Development Company, L.P., 2013).

De seguida encontra-se a camada aplicacional onde, tal como o nome faz prever, se encontra todas as aplicações desenvolvidas pela HP exclusivamente para o mercado do SDN e, tal como referido antes, todas as aplicações disponibilizadas na sua loja de aplicações de SDN.

Ao nível da camada de gestão existem quatro elementos - *VAN SDN Manager*, V*AN Network Resource Automation, VAN Server Connect* e *Intelligent Management Center* -, sendo que o último elemento engloba todas as camadas anteriores da arquitetura de SDN.

Segundo Andre Kindness (Shaikh, 2013), analista sénior na Cambridge, Mass.-based Forrester Research, Inc., é importante haver interação entre a parte física e a parte virtual das redes porque não é possível eliminar ou fazer desaparecer milhões de *Virtual Local Area Networks* (VLAN) que existem. Na sua opinião, o primeiro passo é integrar a área do *Top-of-Rack* (ToR) com os lados físicos e virtuais. É necessário haver interação entre o mundo físico e virtual. Para Kindness, esta parceria é ótima porque a HP é mestre no que se refere à orquestração (Shaikh, 2013).

2.3.1.5 VMware NSX

Apesar de no seu *site*[26] não haver qualquer referência ao SDN, é consensual em vários *sites* da *Internet* (Banks, 2014) (Townsend, 2014) (Ferro, 2013) (Ferro, 2014) (Townsend, 2013) a associação do NSX ao SDN.

A solução *NSX*, da VMware, é uma plataforma segura de virtualização de redes específica para data centers, ou *Software-Defined Data Center* (SDDC)[27] (Townsend, 2013), e surgiu após a aquisição da empresa Nicira, por parte da VMware, Inc. (Townsend, 2014).

Com esta solução, o objetivo da VMware é trazer o modelo operacional de uma máquina virtual para os *data centers* e, assim, economizar a rede e manter a segurança nas operações[27].

[26] VMware NSX <http://www.vmware.com/products/nsx >

[27] NSX: Features <http://www.vmware.com/products/nsx/features.html>

Tal como outras soluções SDN, a base da criação do *NSX* reside na complexidade em que as redes se tornaram, nas imensas configurações e reconfigurações de *routers*, *switches*, *firewalls* e no tempo perdido na correção de erros feitos nas configurações.

Segundo o vídeo introdutório do *NSX*[28], artigos do eBay, AT&T e Rackspace indicam que o tempo de configuração de dispositivos na rede, recorrendo à virtualização na mesma, diminui drasticamente de dias para segundos. Ainda segundo é explicado na demonstração, a rede está no *software*, através da instalação de hipervisores que possuem *vSwitches* na rede física. Posteriormente, é feita a gestão da rede através de aplicações. Recorrendo ao NSX, não terão de ser feitas nem alterações a VLAN, *Access Control List* (ACL) e *firewalls*, nem em *hardware*, porque os serviços de redes, desde a camada 2 até à camada 7, são reproduzidos em *software*, o que simplifica a rede[29].

Na Figura 16, apresenta-se um esquema básico da solução *NSX*, da VMware.

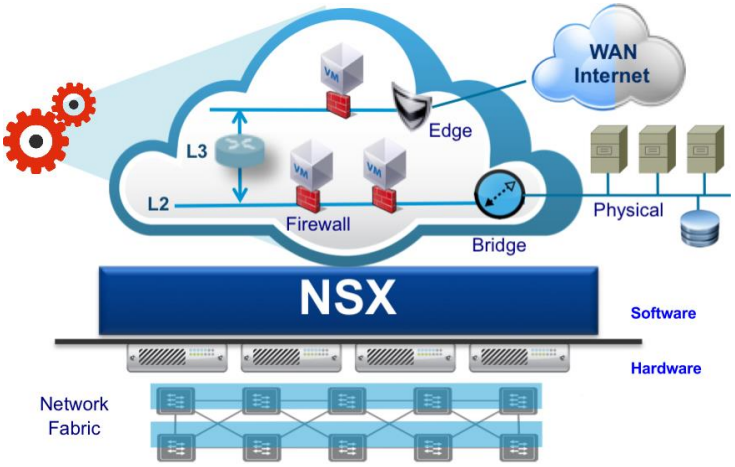


Figura 16 - Esquema básico do NSX

Tal como se observa na Figura 16, é colocada uma camada de *software* sobre o hardware já existente, não havendo necessidade de se efetuar modificações físicas nos *data centers*.

Assim, os responsáveis pela rede apenas terão de se preocupar com pontos mais importantes da rede como, por exemplo, definição da arquitetura da rede e fluxo de tráfego, em vez de se preocuparem com conceitos mais técnicos e de baixo-nível.

Contudo, caso haja falhas na rede, os erros apresentados serão bastante mais detalhados e concisos, ajudando os responsáveis a corrigir muito mais fácil e rapidamente esses mesmos erros.

[28] NSX Overview <http://bcove.me/74djrhqn>

[29] NSX: Features <http://www.vmware.com/products/nsx/features.html>

A utilização do *NSX* tem também em atenção o *OpEx*, o *CapEx*, o tempo de mercado[30] e a segurança da rede.

Além da parceria com a HP, a VMware tem parcerias com outras empresas[31] como, por exemplo, Arista, Brocade, Dell, HP, Juniper Networks, Intel Security, Palo Alto Networks, Symantec, Citrix e F5. Estas parcerias são úteis para, aproveitando os conhecimentos nas áreas em que essas empresas são dominantes, tornar a sua solução mais sustentável e completa.

Tudo isto resulta na arquitetura do NSX, apresentada na Figura 17 (Ferro, 2013).

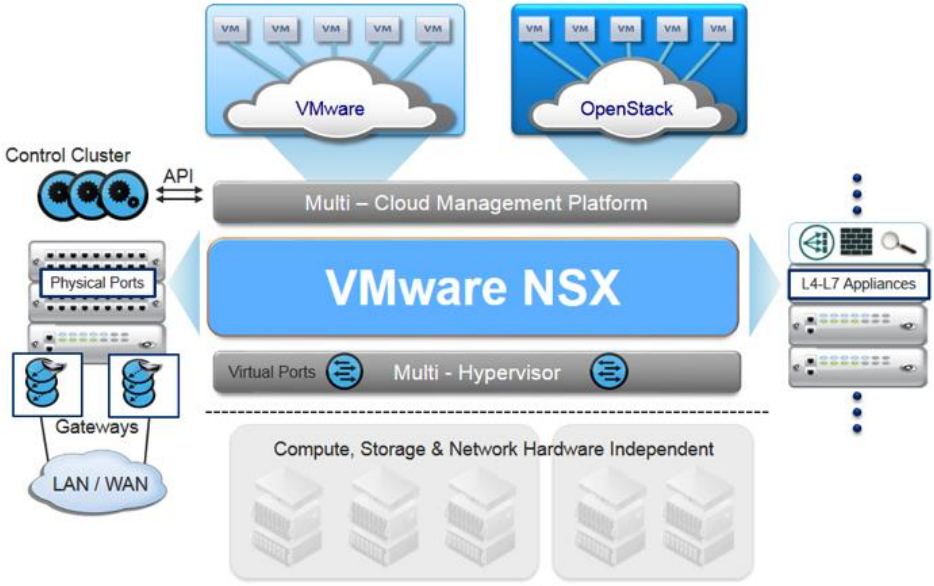


Figura 17 - Arquitetura do NSX

Na Figura 17, pode observar-se a utilização do OpenStack como *framework* aberta para a construção de *clouds* (Behera & Desmidt, 2013).

2.3.1.6 Telecom Italia

Segundo o artigo "*Telecom Italia Creates SDN Network Testbed with Five Universities*" (Schroeder, 2014), publicado no *blog* da Cisco, atualmente, a Telecom Italia ainda não sabe exatamente como irá usar o SDN. Apesar disto, o facto de o SDN permitir rápidas implementações de novos serviços é um ponto extremamente favorável à sua utilização.

[30] NSX: Features <http://www.vmware.com/products/nsx/features.html>

[31] NSX: Resources <http://www.vmware.com/products/nsx/resources.html>

A Telecom Italia já está a trabalhar em conjunto com cinco universidades italianas para se familiarizar com a tecnologia de programação de redes e gestão automática de configurações e já está a estudar e a aprofundar conhecimentos neste campo para, quando a tecnologia estiver mais madura e consistente, a Telco não perder muito tempo a implementá-la na sua infraestrutura e a estudar o SDN tarde de mais. Esta questão está relacionada com *time-to-market*. Os testes e experiências com o SDN continuarão até 2017.

A Telecom Italia está empenhada no SDN também devido às alterações nas configurações poderem ser feitas em tempo real sem ter de haver, obrigatoriamente, intervenção humana.

De entre três empresas/vendedores, a Cisco foi a opção escolhida para se juntar ao consórcio (Telecom Italia/universidades/Cisco). As razões para a sua escolha foram várias, mas as principais foram o suporte e equipamento disponibilizado pela Cisco.

Relativamente às questões técnicas é importante referir que (Schroeder, 2014):

- Os *switches* utilizados já trazem suporte a *OpenFlow*;
- O controlador *Open Daylight* disponibilizado pela Cisco é o *eXtensible Network Controller* (XNC) e está alojado nos laboratórios na Cisco *Unified Computing System* (UCS) *Blade Server*;
- Em cada uma das cinco universidades existe um par de *switches Catalyst 3850*, controlados pelo XNC;
- Cada universidade tem as suas próprias *UCS C220 Series Rack Server* para alojar as aplicações;
- Para poupar recursos foram instalados Cisco *Cloud Services Routers* (CSR) 1000V, que interligam as cinco universidades e que por sua vez se ligam ao laboratório na Cisco, através de túneis *layer* 2 (*Data Link*).

2.3.1.7 Altice Labs

No que diz respeito ao tema do SDN/NFV no panorama do território português, e mais especificamente na Altice, antiga PT, esta empresa tem uma parceria com a Alcatel-Lucent, já referida anteriormente nesta dissertação (Casa dos Bits, 2010) (Portugal Telecom, 2010). No decorrer de uma série de *webinares* promovidos pela Alcatel-Lucent, um dos participantes foi a Altice.

Essa sessão teve a participação de dois elementos da Altice Labs, na altura do *webinar* ainda com o nome de PT Inovação, que esteve representada pelo Eng. Pedro Neves e pelo Eng. Miguel Santos, respetivamente arquiteto SDN/NFV e arquiteto de Soluções e Serviços da Altice Labs. Ambos apresentaram a rede desenvolvida pela Altice e que tem como base a utilização dos conceitos de SDN e NFV. Como se prevê pelo cargo de cada um dos intervenientes, o Eng. Pedro Neves é reponsável pela colocação em prática dos referidos conceitos, enquanto o Eng. Miguel Santos é responsável pelo desenvolvimento e definição dos conceitos a serem implementados na rede.

Esta conferência teve como tema "*An NFV/SDN Enabled Service Providers: A New Generation of Digital Services*"[32]. Nesta apresentação, os participantes explicaram como estão a aplicar e como já está aplicada esta abordagem SDN/NFV na arquitetura da Altice. Naturalmente que, consequentemente, isto beneficia os clientes da empresa da qual fazem parte.

No *webinar*, referiram algo bastante interessante que é a tendência para a convergência de soluções tanto de Telco IT, como de Telco *Network*, como visto e representado na figura no ponto 2.2 deste documento.

A Altice Labs recorreu a máquinas virtuais para que pudessem testar a rede dinamicamente e a nível de escalabilidade.

Apresentaram ainda a arquitetura da Altice, que pode ser consultada na Figura 18, no que diz respeito ao SDN/NFV.

[32] "An NFV/SDN Enabled Service Providers: A New Generation of Digital Services" <https://event.on24.com/eventRegistration/EventLobbyServlet?target=lobby20.jsp&eventid=916706&sessionid=1&key=FA3AFACD50EB5318A25AE9558AF2255E&eventuserid=111164770>

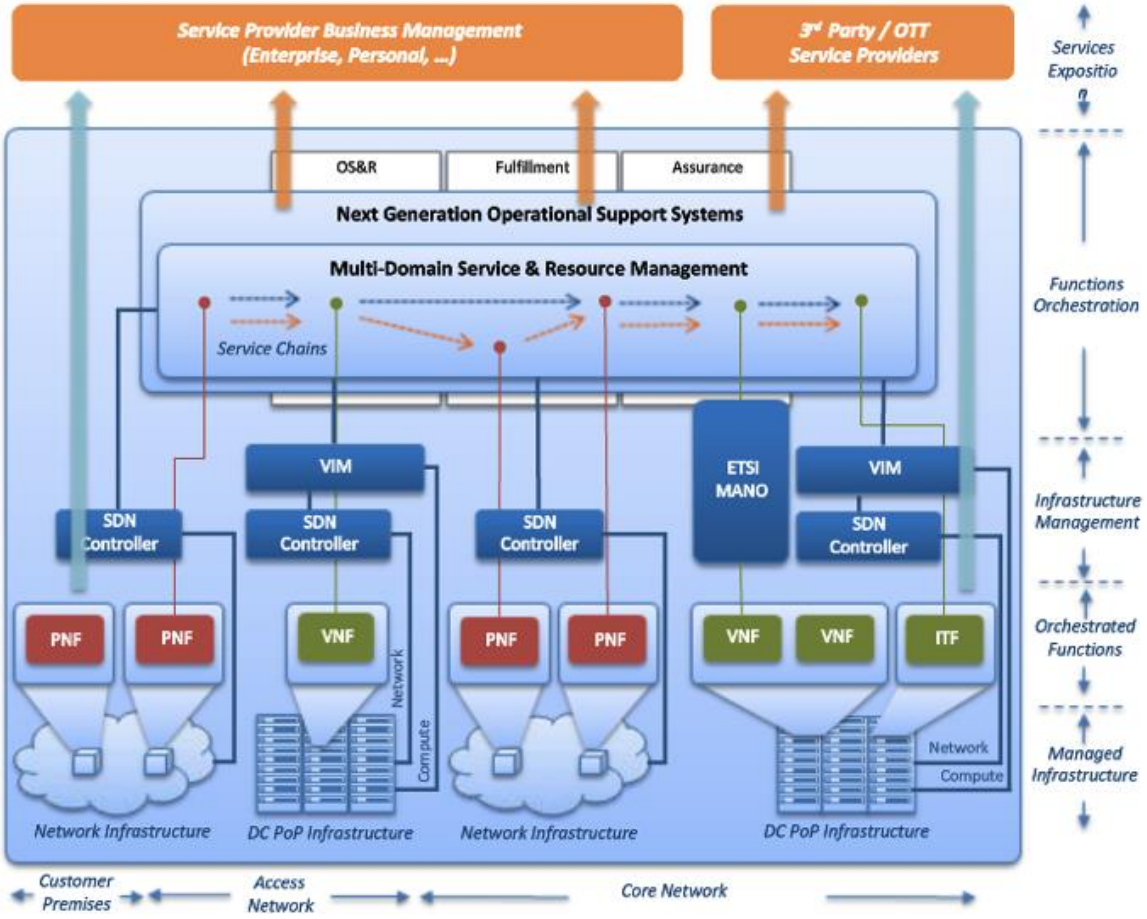


Figura 18 - Arquitetura SDN/NFV da Altice Labs

Relativamente à arquitetura apresentada na Figura 18, os intervenientes chamaram a atenção para o facto de a orquestração ser uma combinação de várias funções, entre elas, *Virtualized Network Functions* (VNF) e *Physical Network Functions* (PNF) e que a arquitetura assenta nos *data centers*.

Apesar de já oferecerem virtualização (não dependem de *hardware*) e arquitetura modular (maior ou menor escalabilidade - *scale in*/*scale out*), estes foram dois dos maiores desafios, sendo que outro é a elasticidade (orquestração), que ainda está em desenvolvimento.

Em suma, alguns dos objetivos da Altice Labs são:

- Gerir a integração de componentes virtuais e de equipamentos físicos;
- Reduzir o *OpEx* e *CapEx*;
- Criar e acelerar o desenvolvimento de serviços;
- Criar parcerias com outras empresas.

Os dados relativos à Altice Labs também podem ser consultados no artigo disponibilizado pela empresa (Altice Labs, 2015).

2.4 Síntese

Em jeito de conclusão deste capítulo, foi aqui que se fez o ponto de situação das redes como hoje as conhecemos e a apresentação das tendências que se prevê que elas sigam, nomeadamente no que diz respeito ao SDN e ao NFV. Foi também neste capítulo que se fez a introdução dessa nova abordagem, bem como a estrutura desta e as tecnologias a ela associadas. Por fim, mencionou-se as soluções desenvolvidas ou que ainda estão em desenvolvimento e estudo por parte de algumas Telco, quer sejam IT, quer sejam *Network*.

No capítulo seguinte irá explicar-se e aprofundar-se a arquitetura projetada, no que diz respeito ao trabalho a realizar relativo a esta dissertação.

Arquitetura

Neste capítulo irá apresentar-se a arquitetura definida para esta dissertação.

Anteriormente, aprofundou-se a estrutura lógica do SDN, que se recorda agora na Figura 19 (Stallings, 2013).

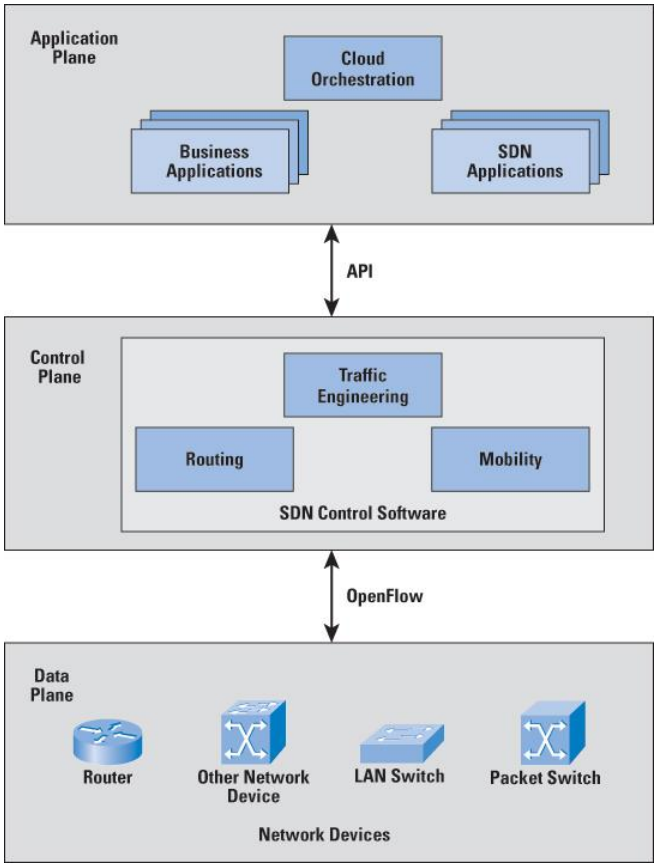


Figura 19 - Estrutura lógica do SDN, segundo Stallings

Facilmente se pode observar na figura que esta estrutura assenta em três pontos. Fazendo uma retrospetiva, relembra-se que todas as soluções SDN empresariais apresentadas tinham, pelo menos, 3 pontos em comum.

Assim, para que se entenda a arquitetura a ser utilizada, primeiro faz-se a análise de cada camada da estrutura lógica do SDN. As três camadas estão listadas de seguida:

- *Application Plane*: no plano aplicacional refere-se algumas *net apps* como, por exemplo, aplicações de orquestração. Estas aplicações irão, explícita, direta e programaticamente comunicar com a rede e tratar do seu comportamento, ou seja, a ordem dada a cada equipamento virá desta camada aplicacional (Open Networking Foundation, 2013);
- *Control Plane*: tem como objetivo implementar todos os protocolos de coordenação que sejam necessários para o correto funcionamento do *Data Plane*. Esta camada é uma entidade centralizada do ponto de vista lógico. É o *Control Plane* que possibilita a abstração das aplicações SDN, independentemente da rede (Open Networking Foundation, 2013);
- *Data Plane*: serve para analisar os cabeçalhos dos pacotes recebidos e encaminhar esses pacotes para o seu destino final, dependendo das tabelas de encaminhamento e comutação. O *Data Plane* não é mais do que a camada onde se encontram os equipamentos da rede, ou seja, é um dispositivo de rede lógica. A representação lógica pode conter toda a rede definida ou apenas parte dela (Open Networking Foundation, 2013).

Consequentemente, a arquitetura definida para esta dissertação também terá as três camadas que se acabou de analisar.

Especificando e aprofundado um pouco mais a estrutura lógica do SDN, chega-se à arquitetura genérica que se pretende implementar e que é muito semelhante a outras que existem. Esta arquitetura genérica está representada na Figura 20.

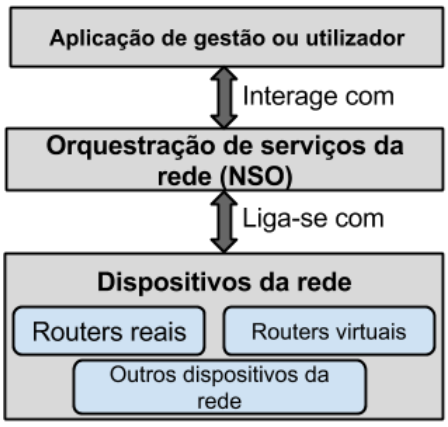


Figura 20 - Arquitetura genérica da implementação realizada

Explicando a Figura 20, e contextualizando-a de acordo com a ferramenta que se pretende desenvolver, apresenta-se as 3 partes seguintes:

- Aplicação de gestão ou utilizador (*Management Application or User*): esta será a camada em que o utilizador irá interagir, abstratamente, com a rede. É onde se prevê que ele ocupe a maior parte do seu tempo. É esta camada, tal como o nome sugere, que serve de ponte, por exemplo, a uma aplicação de gestão já existente, tendo apenas de se fazer as devidas adaptações;
- Orquestração de serviços da rede (*Network Service Orchestration*): esta é a camada "inteligente" da arquitetura que se está a apresentar. É nesta camada que todo o processo se irá desenrolar. O *Network Service Orchestration* irá interpretar os *inputs* do utilizador e transformá-los para que possam ser postos em prática e terem o efeito desejado na rede, que é a próxima e última camada a ser apresentada;
- Dispositivos da rede (*Network devices*): esta última camada provavelmente situa-se no *core* da rede ou onde se pretenda fazer a gestão dos serviços que se pretenda implementar, ou seja, a rede é onde se encontram os dispositivos mais visíveis que existem como, por exemplo, os *routers*.

Conclui-se que a arquitetura genérica apresentada segue a linha das arquiteturas anteriores e a própria arquitetura do SDN. Prevê-se, assim, que a arquitetura utilizada na prática siga também as mesmas ideias. Esta arquitetura está apresentada de seguida.

3.1 Arquitetura utilizada para a implementação

Neste subcapítulo apresenta-se a arquitetura utilizada e que é consequência de uma análise mais esmiuçada da arquitetura genérica anteriormente apresentada.

Na Figura 21 está representada essa análise mais profunda do que a apresentada na Figura 20. De notar que, naturalmente, se teve em atenção o conceito SDN, bem como a estrutura lógica do mesmo, apresentada, por exemplo, por William Stallings num dos seus artigos.

Enquanto na Figura 20 são apresentadas referências um pouco genéricas, na Figura 21, essas referências foram mais especificadas, sugerindo-se uma representação mais prática do sistema.

A arquitetura utilizada serviu de base ao desenvolvimento da ferramenta proposta e que será apresentada no decorrer deste documento.

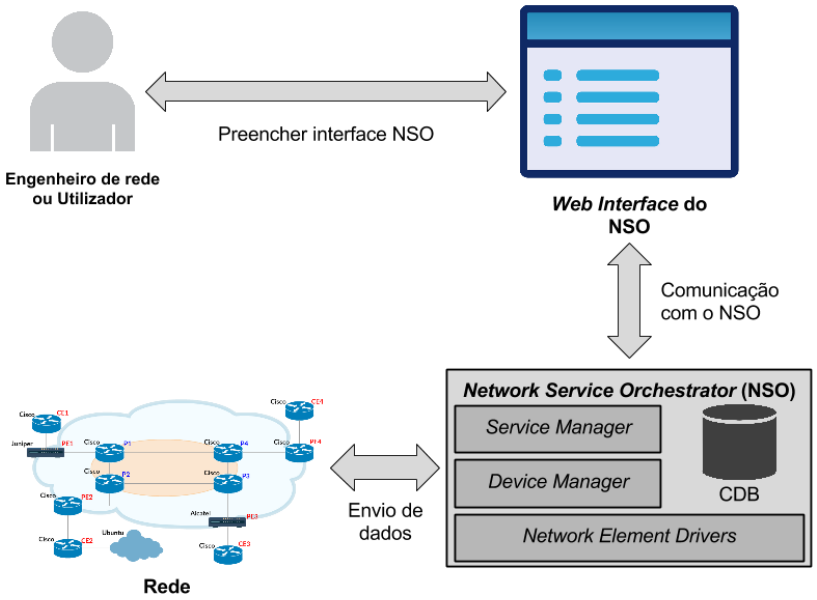


Figura 21 - Arquitetura utilizada na proposta

Na Figura 21 pode-se notar 4 fases principais:

- Engenheiro/Administrador da rede ou Utilizador;
- *Web Interface* de acesso ao *Network Service Orchestrator* (NSO);
- *Network Service Orchestrator*;
- Rede.

Após a definição e respetiva implementação, é importante realçar que é apenas na fase inicial, ou seja, preenchimento da *web interface*, que se prevê que o utilizador interaja com a ferramenta proposta. Tudo o resto deverá ficar a cargo do responsável pela configuração mais profunda e técnica da rede como, por exemplo, o administrador da rede ou dos serviços. Deste modo, o utilizador poderá abstrair-se quase por completo da topologia da rede e de toda a configuração mais repetitiva que normalmente surge no início, como a configuração do endereçamento e do encaminhamento, processo este que deve ser feito logo no começo, pois apenas posteriormente a gestão de serviços pode ser realizada.

Um dos objetivos da ferramenta é que um utilizador com baixo nível técnico possa proceder à configuração básica da rede e dos serviços. Assim, é possível aproveitar tempo que, anteriormente, era despendido de outra maneira em configurações mais repetitivas, como na

otimização de serviços ou em outras tarefas. Este processo será completamente analisado e explicado mais à frente nesta dissertação.

Alguns passos importantes que se dá na aplicação desta arquitetura são:

- Preenchimento da *interface* NSO: depois de o utilizador preencher o formulário do NSO e confirmar esse preenchimento, o *interface* comunica com o NSO. O *interface* já verifica alguns dados básicos como, por exemplo, a validação do IPv4;
- Comunicação com o NSO: o *interface* comunica com o NSO, enviando-lhe os dados recebidos por parte do *interface* e consequentemente introduzidos pelo utilizador. É no NSO que está todo o processo inteligente do sistema e vários passos que serão apresentados ainda nesta secção e que são feitos no próprio NSO;
- Envio de dados: é neste último ponto que se dá o envio da informação do NSO para a rede, ou seja, neste passo as configurações já deverão estar esquematizadas e corretamente configuradas para serem executadas e interpretadas por parte dos equipamentos da rede como, por exemplo, os *routers*. Como descrito à frente nesta dissertação, se necessário terá de haver conversão de comandos/configurações da rede.

De notar que o processo é bidirecional, pois tem de haver comunicação em ambos os sentidos. Isto será mais percetível na apresentação do protótipo mas, para que se entenda a razão deste facto, é importante referir que, por exemplo, no caso de se estar a testar uma rede recorrendo à ferramenta utilizada ou a uma semelhante, toda a alteração feita nessa mesma rede por um utilizador deve ser visível por qualquer outro. Daí a importância desta bidirecionalidade para a correta utilização e configuração numa rede.

Assim, nota-se que a configuração, como hoje se conhece, se torna um pouco abstrata para o utilizador. Observa-se, também, que o ponto fundamental e mais importante desta arquitetura é o NSO. Por isso, é importante apresentar e explicar as partes que o constituem, e que são (Cisco Systems, Inc., 2014):

- *Service Manager*: é neste ponto que está a inteligência da ferramenta NSO e que se possibilita ao utilizador a gestão, em alto-nível, de aspetos da rede que não sejam suportados nos dispositivos que estejam diretamente ligados. Definindo-se previamente os serviços que estejam a ser executados no *Service Manager*, o utilizador pode configurar um serviço existente num único local. Posto isto, o *Service Manager* tratará do restante processo de gerir (criar, editar ou eliminar) os serviços da

rede, sendo este processo abstrato e transparente para quem esteja encarregue da configuração da rede;

- *Device Manager*: tem como função fazer a gestão da configuração dos dispositivos de forma transacional, suportando recursos como a sincronização de configurações bidirecional e mudanças refinadas em tempo real nos dispositivos;
- Base de dados do NSO ou *Configuration Database* (CDB): aqui é armazenada toda a informação referente às configurações dos dispositivos, havendo assim a sincronização dos dados devido à existência deste ponto. É no CDB que há sincronização, consistência e reconciliação no que diz respeito à configuração entre os serviços e os dispositivos;
- *Network Element Drivers* (NED): são os responsáveis pela ligação entre o NSO e os dispositivos de rede. O NED utiliza o conceito de atomicidade, ou seja, na execução de um comando ou ele está correto e é executado, ou basta que um parâmetro do comando esteja errado e nada será executado. O NSO, de acordo com o dispositivo que se pretende configurar, informa o NED do tipo do dispositivo (*device-type*), ficando este último a saber o que deve fazer, independentemente da sua marca ou fabricante. O *interface* do dispositivo é modelado em ficheiros, recorrendo ao *YANG*, fazendo assim com que cada ficheiro esteja modelado com os comandos – que podem ser atualizados – do respetivo dispositivo. A filosofia dos NED varia de dispositivo para dispositivo pois, enquanto para a Cisco e para a Alcatel os comandos são convertidos para a linha de comandos (CLI) para serem executados no terminal do dispositivo, para os equipamentos Juniper, que já utilizam NETCONF – tecnologia, usada pelo SDN, baseada na codificação em XML e já apresentada nesta dissertação –, a filosofia é diferente, não necessitando de converter configurações.

Por fim, na arquitetura definida, está presente a rede como usualmente é conhecida. É de realçar um fator importante na utilização do NSO: a rede, já existente ou criada de raiz, pode ser real ou virtual. No caso de se pretender testar uma rede antes de ela ser colocada em produção, o recurso a uma rede virtual pode ser algo bastante útil e muito mais prático de ser usado. De notar que os erros ou falhas em redes de teste virtuais, certamente não terão um impacto negativo e os seus testes e as suas experiências podem ser tão ou mais esmiuçados do que num equipamento real. Mas obviamente que, testá-las num equipamento real antes de serem colocadas em produção, é algo naturalmente fundamental. Esta referência serve apenas

para identificar que, inicialmente e com maior rapidez, se pode utilizar um simples computador para se testar, por exemplo, uma rede virtual.

Estes equipamentos podem ser de diversos modelos/fabricantes, desde que o NSO tenha suporte para estes equipamentos. A rede é dada a conhecer ao NSO através do protocolo de comunicação *Secure Shell* (SSH).

De seguida são apresentadas as operações disponibilizadas ao utilizador, de modo a que ele possa usufruir da ferramenta na gestão da rede.

3.2 Operações disponibilizadas na ferramenta proposta

Antes de se partir para o desenvolvimento da ferramenta proposta, definiu-se alguns serviços a serem testados, bem como as operações que se previa serem disponibilizadas por parte da ferramenta proposta.

Os serviços definidos e que foram testados na ferramenta foram:

- L3VPN: Qualidade de Serviço (QoS) e *Virtual Private Network* (VPN) - a este serviço estão associadas as classes e as políticas que podem ser definidas na rede e que estão associadas ao QoS. Também se definiu a gestão de serviço de VPN onde se pode fazer a gestão da VPN e respetivos *endpoints* associados à VPN;
- *Virtual Local Area Network* (VLAN): serviço simples para gestão de VLAN, que está associado aos equipamentos da rede e respetiva *interface* que se pretenda configurar;
- *Hostname*: serviço básico para gestão de *hostname* dos equipamentos, tal como o nome indica. Este serviço foi desenvolvido como prova de conceito para demonstrar e testar o funcionamento da ferramenta proposta.

É importar notar que os serviços implementados serviram de teste à ferramenta. Naturalmente que, desde que corretamente implementado, qualquer serviço pode ser criado e utilizado na ferramenta proposta.

Depois de apresentados os serviços, que correspondem às páginas *web* da ferramenta proposta, sumarizam-se, de seguida, todas as ações que podem ser realizadas pelo utilizador:

- Adicionar, editar e remover classes de QoS: dar a possibilidade ao utilizador de gerir classes de QoS, tendo associados os valores de *Differentiated Services Code Point*

(DSCP), o tipo de tráfego de uma classe, endereço e porto de origem e destino e, por último, o protocolo associado àquela classe de QoS;

- Adicionar, editar e remover políticas de QoS: dar a possibilidade de associar cada classe, previamente definida, a uma política de QoS. Define-se, também, a largura de banda e a prioridade para essa política;
- Adicionar, editar e remover VPN: possibilitar a gestão de uma VPN. A essa VPN está associado o número *Autonomous System* (AS), os *endpoints*, os equipamentos da rede e a *interface* onde se pretende configurar a VPN e a largura de banda que se pretende definir para um *endpoint*;
- Adicionar, editar e remover VLAN: possibilitar a gestão de um *link-VLAN*, associado a uma VLAN e do equipamento e respetiva *interface* que também se pretenda associar a uma VLAN;
- Editar *hostname*: possibilitar a edição do *hostname* de um equipamento da rede. Este serviço é o mais simples de todos os implementados e será analisado no próximo capítulo;
- Atualizar a base de dados: sincronizar a base de dados quando o utilizador desejar. Assim, há troca de informação entre a rede e a base de dados, onde estão armazenados os dados a serem apresentados nos formulários ao utilizador. Assim, todos os pontos têm as configurações mais recentes existentes na rede;
- Exportar as configurações do NSO: dar a possibilidade ao utilizador de analisar toda a configuração que o NSO executou na rede, através de um simples ficheiro de texto;
- Atualizar configurações de serviços e dispositivos da rede;
- Adicionar dispositivos à rede: possibilitar a adição de um dispositivo à rede, dando-se a conhecer à ferramenta esse novo equipamento. Esta definição tem exatamente esse propósito.

Com a apresentação das operações disponibilizadas pela ferramenta proposta ao utilizador, conclui-se este capítulo da Arquitetura.

3.3 Síntese

Em suma, neste capítulo apresentou-se a arquitetura desenvolvida, bem como a apresentação de alguns conceitos necessários para a correta compreensão da arquitetura proposta.

Além disto, foram apresentadas as funcionalidades disponibilizadas pela ferramenta proposta, bem como a referência aos serviços utilizados.

Concluída a apresentação da arquitetura proposta, analisa-se no capítulo seguinte o protótipo desenvolvido.

Esta página foi intencionalmente deixada em branco

Protótipo

Neste capítulo irá apresentar-se o protótipo implementado. Esta solução tem por base a arquitetura anteriormente definida. Como resultado final, apresenta-se uma ferramenta simples e que se pretende que seja capaz de fazer a gestão de serviços de uma rede.

4.1 Implementação do protótipo

Tendo por base o conceito de SDN, a implementação que se vai apresentar é de uma ferramenta, *Open Source*, desenvolvida para servir de apoio a testes a serem realizados numa rede, que pode ser real ou virtual e pode já existir ou ainda não. Além disto, também será possível realizar testes à gestão de serviços. Tudo isto serve para se experimentar todas as funcionalidades antes que a rede seja colocada em produção. A utilização do conceito de SDN é muito importante neste processo, pois a gestão da rede passa a ser feita num ponto centralizado da rede, facilitando bastante a configuração da mesma. Assim, de uma maneira mais automatizada trata-se da configuração da rede, bastando o utilizador usar um único ponto de acesso.

Do ponto de vista puramente visual, a ferramenta desenvolvida não é mais do que um *interface* gráfico ou *WebUI* com várias páginas *web* para a gestão de uma rede. Sobretudo este *interface* baseia-se em formulários.

Depois de definidos e apresentados os serviços implementados, foi tempo de passar da definição e da "teoria" do que se tinha previsto implementar para a "prática", ou seja, implementação da ferramenta pensada.

De seguida será explicado o processo de implementação da ferramenta proposta desde a fase inicial de implementação do cenário até à fase final de desenvolvimento da solução.

A proposta assenta em 3 fases principais:

- Cenário/topologia da rede – onde estão presentes os equipamentos da rede;

- Desenvolvimento da camada intermédia – camada que vai fazer a ligação entre a configuração e os equipamentos da rede e que é transparente para o utilizador. A base do desenvolvimento é a utilização da plataforma "*Cisco Network Service Orchestrator enabled by Tail-f*" (NSO), que acaba por ser a que liga a topologia da rede ao *interface* gráfico. Esta fase será o *back-end* para o utilizador;
- *Interface* gráfico – local principal de interação entre o utilizador e a rede, seja ela física ou virtual. *Front-end* para o utilizador.

É importante, então, apresentar as fases de implementação. Nos subtópicos seguintes será apresentada cada uma dessas fases anteriormente referidas.

4.1.1 Cenário/Topologia da rede

Para a conceção da rede optou-se por criar uma virtualização/simulação de rede para que esta fosse mais prática e funcional. Para isso, criou-se uma máquina virtual em Linux Ubuntu a correr a versão 14.04. Nessa máquina desenvolveu-se uma rede no *software* de simulação de redes, GNS3[33], onde foram configurados equipamentos de 3 fabricantes diferentes: Alcatel-Lucent, Cisco e Juniper.

Para esta simulação foram utilizados os seguintes modelos dos routers apresentados na Tabela 1.

FABRICANTE	MODELO
ALCATEL-LUCENT	ALCATEL SR-OS 7750 TiMOS-B-12.0.R6
CISCO	Cisco IOS Software, 3700 Software (C3725-ADVENTERPRISEK9-M), Version 12.4(25d)
JUNIPER	firefly-perimeter, JUNOS Software Release [12.1X47-D15.4]

Tabela 1 - *Routers* simulados na implementação

Relativamente à configuração destes equipamentos, esta é muito simples, bastando ter-se feito a configuração do endereçamento e do encaminhamento onde seria necessário.

O encaminhamento usado na topologia foi o *Open Shortest Path First* (OSPF) e o protocolo de comunicação utilizado para fazer a ligação entre a rede e a ferramenta foi o SSH.

Na Figura 22 apresenta-se uma imagem genérica da topologia implementada.

[33] GNS3 Technologies Inc.: What is GNS3? <https://www.gns3.com/software>

Uma figura mais desenvolvida e completa pode ser consultada no Anexo 2. Nessa figura é possível consultar o endereçamento definido na topologia de rede realizada.

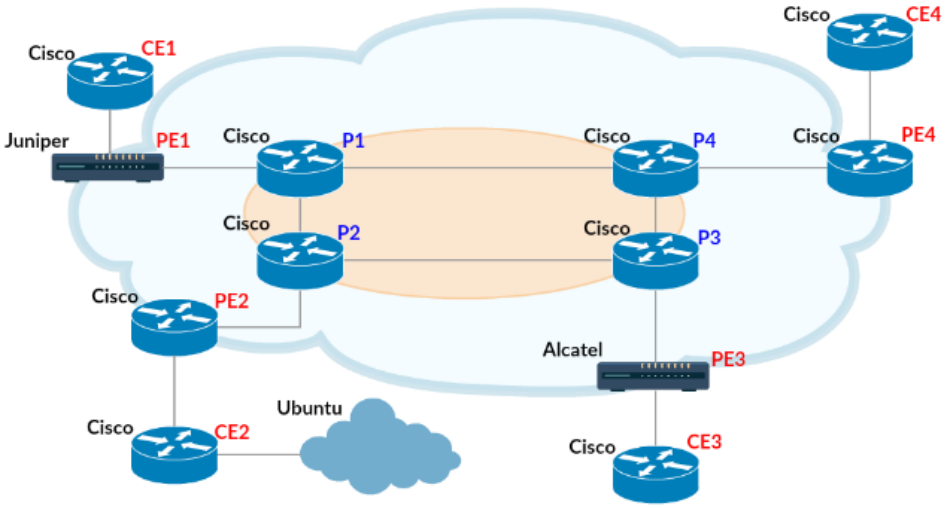


Figura 22 - Topologia da rede definida no GNS3

A fazer a ligação entre a topologia da rede definida no GNS3 e a ferramenta desenvolvida utiliza-se a solução NSO, da Cisco/Tail-f, tal como referido antes, e que será analisada em pormenor a seguir.

4.1.2 Desenvolvimento da camada intermédia

Depois de a topologia e respetiva configuração dos equipamentos estar concluída, definiu-se alguns serviços a serem geridos e testados na rede. Pretende-se que esta gestão seja feita utilizando a ferramenta desenvolvida, ou seja, a *WebUI*. Os serviços testados foram *Quality of Service*, ou QoS, e *Virtual Private Network*, ou VPN. Mais tarde adicionou-se dois serviços simples para configuração de *Virtual Local Area Network*, ou VLAN, e para configuração do *Hostname* dos equipamentos. Este último viria a servir como prova de conceito da ferramenta desenvolvida. Após saber-se quais os serviços a serem implementados, definiu-se os parâmetros dos mesmos. De modo a criar uma sincronização dos parâmetros definidos no *interface* gráfico desenvolvido e no NSO, optou-se por utilizar uma base de dados que iria proceder à troca de informação entre a *WebUI* e o NSO, tendo por objetivo que os dados apresentados ao utilizador, na ferramenta gráfica, fossem consistentes em ambos os lados. Chega-se assim às tabelas de base de dados utilizadas na ferramenta e que estão apresentadas no Anexo 3.

Estando definida a base de dados e os parâmetros dos serviços, criou-se um "esqueleto" (*template skeleton*) de cada serviço a ser criado. Nesse "esqueleto" existem diversos ficheiros, entre eles o de modelação de serviços.

Para fazer a modelação de serviços, o NSO recorre ao *YANG*. Como já referido anteriormente, o *YANG* é uma linguagem de modelação de dados usada para um modelo de configuração de estado dos dados. Esta linguagem é usada pelo protocolo NETCONF e está publicada na RFC 6020, de setembro de 2010. O *YANG* está relacionado com o conteúdo e operações nas camadas do NETCONF[34].

É nos ficheiros *YANG* que são definidos os campos ou parâmetros a serem pedidos para a correta implementação dos serviços na rede. Na Figura 23 é apresentada parte de um exemplo de um ficheiro *YANG* (*hostname.yang*) para a implementação do serviço de *Hostname*, que tem por objetivo a alteração do *hostname* no dispositivo pretendido. Este serviço, tal como referido previamente, serve para demonstrar a implementação feita e terá reflexo nos dispositivos da rede de teste.

```
module hostname {
  namespace "http://com/example/hostname";
  prefix hostname;
  import tailf-ncs {
     prefix ncs;
  }
  container host {
     list hostname {
           description "Configure hostname";
           key name;
           uses ncs:service-data;
           ncs:servicepoint "hostname";
           leaf name{
                 type string;
           }

             leaf device {
                 type leafref {
                   path "/ncs:devices/ncs:device/ncs:name";
                 }
             }
             leaf changeto {
                 type string;
             }
      }
    }
  }
```

Figura 23 - Ficheiro *YANG* para modelação de um serviço simples: *Hostname* (*hostname.yang*)

Na Figura 23 pode-se observar os parâmetros definidos e que servirão de base aos dados a serem preenchidos no NSO. Pode-se verificar que o *YANG* modela o nome do dispositivo

[34] What is YANG?<http://www.tail-f.com/education/what-is-yang/>

cujo *hostname* se pretende alterar e o seu próprio *hostname*. Caso se execute o comando para a criação do serviço de *Hostname*, ele obedece, mas não terá, para já, efeito nos dispositivos da rede, pois apenas após a definição do mapeamento ele começará a ter o referido efeito na rede. Na Figura 24, pode-se observar a sequência de como o NSO é usado para manipular a abstração de serviços, ou seja, não se consegue verificar para que modelo ou fabricante esta configuração é feita.

```
admin@ncs(config)# host hostname troca device pe1 changeto pe1device
admin@ncs(config-hostname-troca)# top
admin@ncs(config)# show configuration
host hostname troca
 device   pe1
 changeto pe1device
!
```

Figura 24 - Abstração na execução de comandos

Em termos de abstração, o resultado dos comandos executados no terminal do NSO e apresentado na Figura 24 é um facto que passa despercebido ao utilizador. O que ele pretende é proceder às configurações independentemente do que exista na rede e abstraindo-se dela, isto é, mesmo sem a conhecer pormenorizadamente.

Depois da alteração do ficheiro *YANG*, deve-se definir o mapeamento do serviço para que, assim, o comando seja executado e o serviço criado. Relativamente à definição do mapeamento, isto não é mais do que a alteração do *template* (*hostname.xml*) que é gerado quando se cria o serviço no NSO. Assim, e para o exemplo do serviço *Hostname*, o resultado do *template* poderá ser o apresentado na Figura 25.

```
<config-template xmlns=http://tail-f.com/ns/config/1.0
servicepoint="hostname">
  <devices xmlns="http://tail-f.com/ns/ncs">
     <device>
           <name>{/device}</name>
           <config>
                 <hostname xmlns="urn:ios">{/changeto}</hostname>
                 <configuration
xmlns="http://xml.juniper.net/xnm/1.1/xnm">
                       <system>
                             <host-name>{/changeto}</host-name>
                       </system>
                 </configuration>
           </config>
     </device>
  </devices>
</config-template>
```

Figura 25 - *Template* do serviço *Hostname* (*hostname.xml*)

Na Figura 25 pode-se ainda observar que o *template* já segue a configuração do *hostname*, seja para o *router* Cisco, identificado pelo seu sistema operativo (IOS), seja para o router da Juniper, identificado pelo seu sistema operativo (JunOS). Caso se pretendesse utilizar mais

equipamentos, teria de se proceder a algumas adaptações ou alterações no *template* do serviço.

O passo seguinte é a execução, propriamente dita, do serviço. Na Figura 26, apresenta-se um comando que serve como demonstração à configuração do serviço *Hostname*, bem como de exemplo à execução prática de alteração do *hostname* de um dispositivo, neste caso do *router p0*.

```
admin-ncs(config)# host hostname troca device p0 changeto p0cisco
admin-ncs(config)# commit
```

Figura 26 - Comando a ser executado no terminal NSO

Depois da execução do comando e do *commit*, o serviço começa a correr na rede. O resultado desta execução será apresentado no subcapítulo "4.2 Teste ao funcionamento da ferramenta".

Concluída a apresentação da camada intermédia, passa-se a apresentar no subtópico seguinte a ferramenta gráfica que serve de suporte ao objetivo proposto.

4.1.3 Ferramenta gráfica desenvolvida

A fase final de execução do protótipo realizado resultou no desenvolvimento de um *interface* gráfico que é onde se prevê que o utilizador interaja maior parte do tempo, no que toca à gestão de serviços.

O *interface* gráfico foi criado a partir do tema "*Big Impresa by Iografica Themes*", em *WordPress* e é muito simples, baseando-se sobretudo em formulários, como se pode observar pela Figura 27.

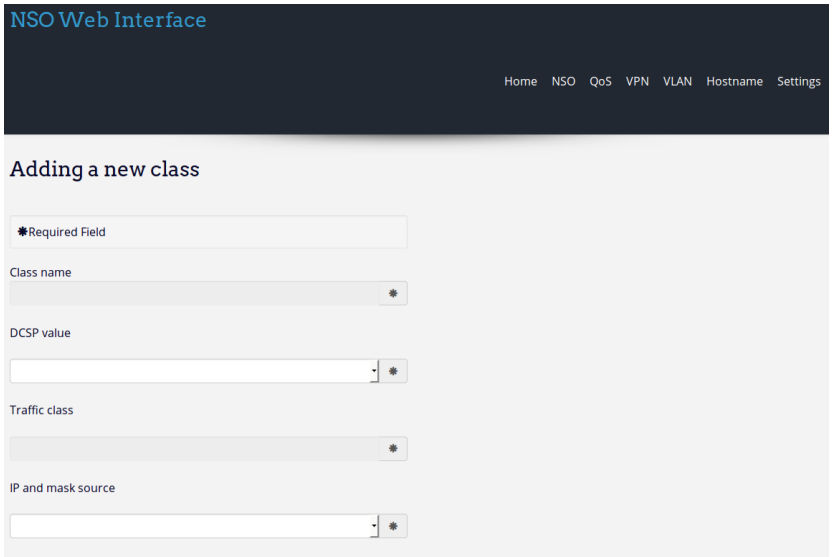


Figura 27 - Parte do *interface* gráfico desenvolvido na proposta

Observada a Figura 27, é importante relembrar que o objetivo principal não foi a implementação de uma *web interface* de alto nível mas sim o desenvolvimento de uma solução simples que possa servir a uma fase prévia de testes à rede ou aos serviços desenvolvidos.

De modo a enquadrar este *interface*, a Figura 28 mostra uma sequência dos dados utilizados no desenvolvimento.

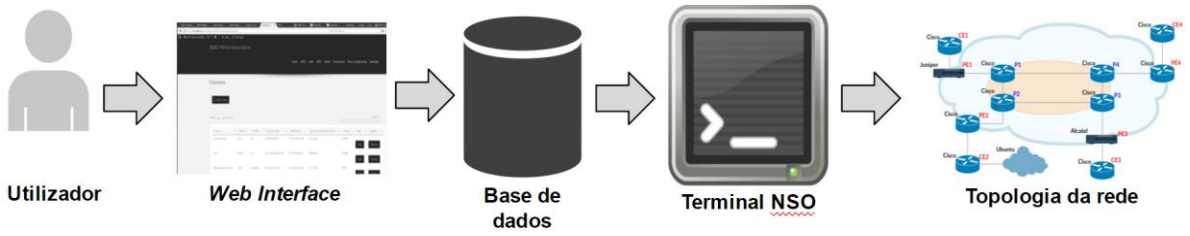


Figura 28 - Esquema de utilização do *interface* gráfico

Na Figura 28 está representado o esquema de interação entre o utilizador e a ferramenta de configuração, que posteriormente se irá refletir na rede existente. A implementação da *WebUI* dividiu-se em duas partes: a visível (*front-end*) e a não visível (*back-end*), onde se está a executar os processos mais importantes. O *front-end* é muito simples e baseia-se, sobretudo, em botões e formulários de preenchimento. O *back-end* é onde se faz a leitura dos dados que foram previamente preenchidos nos formulários pelo utilizador. Essa leitura segue a sequência apresentada na Figura 29.

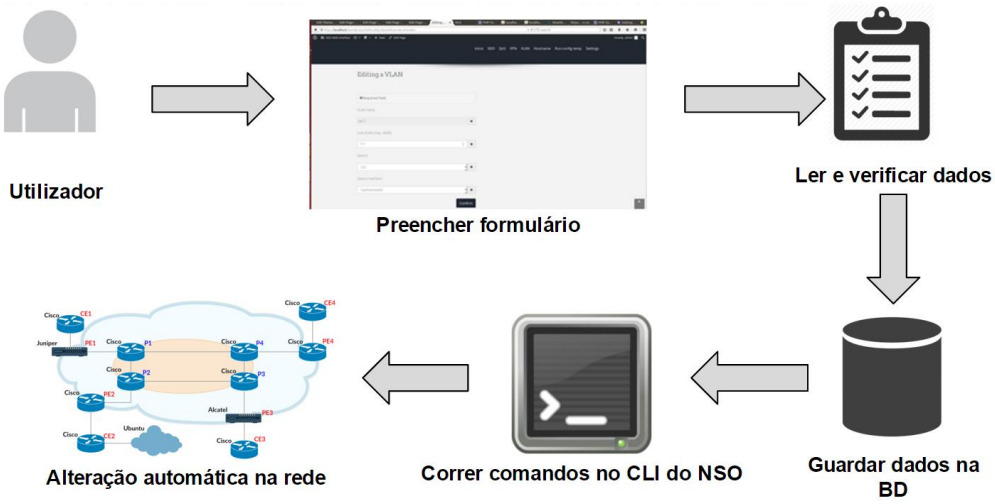


Figura 29 - Sequência do processo executado em *back-end* no *interface* gráfico

Relativamente à base de dados, e relembrando que as tabelas da mesma estão apresentadas no Anexo 3, ela tem um papel importante, pois serve para apresentar os dados ao utilizador. Estes dados estarão sincronizados no NSO e serão lidos na base de dados para posteriormente serem apresentados ao utilizador no formulário.

Como já referido anteriormente, o processo inverso também será realizado pois, caso se faça alterações na rede, estas terão de ser apresentadas na ferramenta gráfica. Esta sincronização também tem de estar bem realizada. Caso contrário, pode dar-se o caso de o utilizador estar a fazer um preenchimento correto dos dados mas a rede não os aceitar. Isto pode ser causado por conflito na sincronização dos dados entre a base de dados, apresentada através do *interface* gráfico ao utilizador, e o NSO.

O comando mais importante, e é através dele que se estabelece a ligação entre o *interface* gráfico e o terminal NSO, é o apresentado na Figura 30.

```
$ /home/tail-f/ncs_new/bin/ncs_cli -C -u admin
```

Figura 30 - Comando de acesso ao terminal NSO

O comando apresentado na Figura 30, e que parece ser bastante simples, é o que vai proceder à ligação ao terminal do NSO, sendo posteriormente aqui que são executados os comandos para a alteração da rede. A execução de um *script* com este comando tem reflexo no terminal NSO e, mais tarde, na rede existente. Aprofundando um pouco mais este ponto, o que acontece é que, depois de o utilizador preencher o formulário e confirmar as alterações que pretender, os dados sofrem uma verificação prévia e é gerado um comando a ser incrementado a um *script bash*. Esse comando gerado automaticamente, e que já está de acordo com o que será executado mais tarde no terminal do NSO, está apresentado na Figura 31.

```
$ nso terminal("config \n host hostname troca device   " . $device . "
changeto " . $hostname ."\n top");
```

Figura 31 - Comando a ser incrementado no script bash

Como apresentado na Figura 31, é gerado e executado automaticamente um comando no *interface* gráfico já com os parâmetros vindos do formulário e que estarão corretamente preenchidos.

Continuando o processo, a maneira como os dados chegam ao NSO é sob forma de *script*. Na Figura 32 é apresentado um *script* exemplificativo, com o intuito de ser executado no NSO.

```
#!/bin/bash

  source "/home/tail-f/ncs_new/ncsrc"

  CLI="${NCS_DIR}/bin/ncs_cli -C -u admin"
  $CLI <<EOF
     config
  host hostname troca device  ce2 changeto CE2Cisco
  top
     commit
  exit
  exit
```

Figura 32 - *Script* para alteração de configurações no terminal NSO

Como se pode concluir, este é um *script bash* compatível com o NSO (Cisco Systems, Inc., 2014). Pode-se notar que o comando apresentado previamente na Figura 31 também está presente na Figura 32. A única diferença que se pode observar é que nesta última o comando já se encontra preenchido com os dados introduzidos pelo utilizador da ferramenta.

Opcionalmente, é através de um *script bash* que se comunica com o CLI do NSO, pois desta maneira faz-se o que se pretende sem grande complexidade e servindo o que se pretende demonstrar. Mas haveria outras opções que se poderia tomar e que podiam passar pela utilização de *REST*, caso já se tenha uma aplicação de gestão neste estilo, ou pela utilização de uma API em JSON RPC, caso se pretenda utilizar *JavaScript* no desenvolvimento de um cliente *web* (Cisco Systems, Inc., 2014). Outras formas haveria para manipular esta configuração no CLI do NSO.

Dá-se, assim, por concluída a implementação realizada. Para finalizar o capítulo, apresenta-se um teste exemplificativo de todo o processo realizado.

4.2 Teste ao funcionamento da ferramenta

De modo a demonstrar o processo explicado até ao momento, apresenta-se um teste simples ao serviço *Hostname*, tal como referido anteriormente. Depois de se ter concluído a implementação do serviço, apresentada no ponto "4.1.2 Desenvolvimento da camada intermédia", irá agora apresentar-se o resultado da execução, na prática, do serviço que tem como objetivo a alteração do *hostname* de um equipamento, independentemente da marca/modelo do fabricante. Para este teste, primeiro tem de se selecionar o equipamento em que se pretende efetuar a alteração e, de seguida, no mesmo formulário definir o novo nome que se pretende dar ao *hostname* do dispositivo. Esta ação pode ser observada na Figura 33.

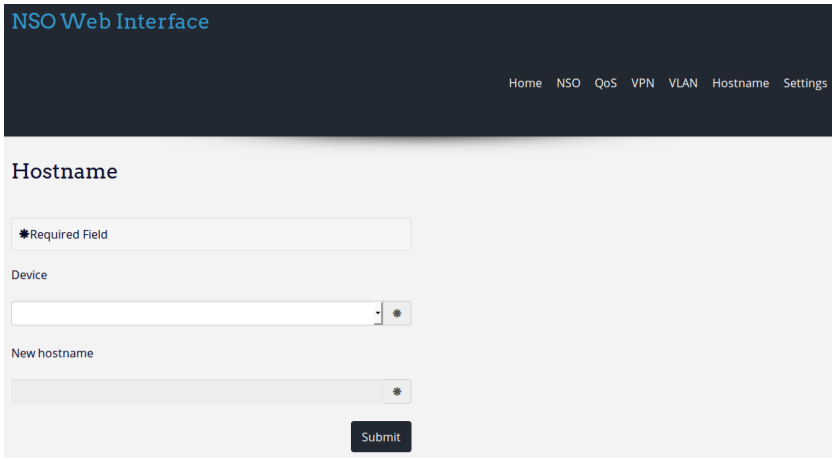


Figura 33 - Formulário relativo ao serviço *Hostname*

Na Figura 33 apresenta-se o preenchimento do formulário relativo ao *router p0*, que é um equipamento Cisco, mas o processo é exatamente o mesmo para outro router, como um Juniper, por exemplo.

Depois do preenchimento dos campos e de se confirmar a alteração a efetuar, ou seja, alteração do *hostname* de um determinado equipamento que esteja presente na rede, o processo está concluído do lado do utilizador. Apesar disso, explica-se agora o que acontece em *back-end* no sistema.

Após a confirmação das alterações e a validação dos parâmetros, os comandos que se executa são os apresentados, previamente, na Figura 26 e na Figura 30.

Na Figura 30 é apresentado o comando que fará a ligação ao terminal NSO, para posterior execução dos comandos.

Na Figura 26 procede-se à parametrização dos dados, isto é, à colocação dos dados no comando a ser executado, dados estes que foram obtidos por parte do formulário, já preenchido pelo utilizador.

Até ao momento que antecede a execução do comando *commit*, os dados estão sincronizados com a base de dados do NSO, ou seja, o CDB. Caso contrário, os dados e as configurações permaneceriam inalteradas. Estando tudo correto, é feita a definição do mapeamento, que já foi apresentada e que passa a estar presente nos *templates* no ficheiro XML.

Depois disso, o NED interpreta os dados recebidos e os comandos são executados no respetivo equipamento. Na Figura 34, está representado o reflexo desta execução no terminal do NSO. Apresenta-se também mais alguns comandos para melhor compreensão da sequência de passos que se dá na execução do serviço que se pretende implementar num *router* Cisco.

```
admin@ncs(config)# host hostname troca device p0 changeto p0Cisco
admin@ncs(config-hostname-troca)# top
admin@ncs(config)# commit dry-run outformat native
device p0
  hostname p0Cisco
admin@ncs(config)# commit
Commit complete.
admin@ncs(config)# show full-configuration devices device p0 config ios:hostname | display xml
<config xmlns="http://tail-f.com/ns/config/1.0">
  <devices xmlns="http://tail-f.com/ns/ncs">
  <device>
    <name>p0</name>
      <config>
      <hostname xmlns="urn:ios" refcounter="1" >p0Cisco</hostname>
      </config>
  </device>
  </devices>
</config>
admin@ncs(config)# show full-configuration devices device p0 config ios:hostname
devices device p0
 config
  ios:hostname p0Cisco
 !
!
```

Figura 34 - Resultado da execução dos comandos a serem executados no terminal do *router* Cisco

Como se pode observar na Figura 34, inicialmente define-se o *hostname* que se pretende dar ao *router p0* e que está realçado na figura, tendo sido previamente preenchido pelo utilizador.

Semelhante processo é efetuado para *routers* de outros fabricantes, como o da Juniper, que é apresentado na Figura 35.

```
admin@ncs(config)# host hostname troca device pe0 changeto pe0Juniper
admin@ncs(config-hostname-troca)# top
admin@ncs(config)# commit dry-run outformat native
device pe0
   ...
      <config>
        <configuration xmlns="http://xml.juniper.net/xnm/1.1/xnm">
          <system>
            <host-name>pe0Juniper</host-name>
          </system>
        </configuration>
      </config>
   ...
admin@ncs(config)# commit
Commit complete.
admin@ncs(config)# show full-configuration devices device pe0 config junos:configuration system hos
t-name | display xml
<config xmlns="http://tail-f.com/ns/config/1.0">
  <devices xmlns="http://tail-f.com/ns/ncs">
  <device>
    <name>pe0</name>
      <config>
      <configuration xmlns="http://xml.juniper.net/xnm/1.1/xnm">
      <system>
      <host-name refcounter="2"  original-value="pe0">pe0Juniper</host-name>
      </system>
      </configuration>
      </config>
  </device>
  </devices>
</config>
admin@ncs(config)# show full-configuration devices device pe0 config junos:configuration system hos
t-name
devices device pe0
 config
  junos:configuration system host-name pe0Juniper
 !
!
```

Figura 35 - Resultado da execução dos comandos a serem executados no terminal do *router* Juniper

Tal como apresentado na Figura 34, o processo repete-se mas, desta vez, para o *router* da Juniper, na Figura 35. De notar que, nos equipamentos deste fabricante, o *output* nativo tem uma aparência diferente do *router* da Cisco, pois este router utiliza o protocolo de configuração de rede, NETCONF - serve para fazer a gestão de configurações em dispositivos de rede -, assemelhando-se bastante à estrutura *eXtensible Markup Language*, ou XML.

Tendo em conta que são notadas alterações no cenário, é previsível que os comandos tenham sido executados nos equipamentos e que o seu serviço também tenha sido executado.

O resultado esperado da consumação dos comandos está apresentado na Figura 36. Este resultado que se veio a confirmar foi a alteração do *hostname* no equipamento que se definiu.

```
CE2#
CE2#
*Mar  3 15:49:08.208: %SYS-5-CONFIG_I: Co
(192.168.200.2)
CE2Cisco#
CE2Cisco#show running-config
Building configuration...

Current configuration : 1288 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname CE2Cisco
!
```

Figura 36 - Resultado da execução do comando de criação do serviço *Hostname* executado no *router* Cisco

Como se observa na Figura 36, a execução dos comandos e respetivo serviço são bem sucedidos, pois verifica-se a alteração do *hostname* do *router* pretendido.

Nos testes realizados com outros serviços também se constatou que as alterações realizadas na ferramenta proposta surtiram efeito nos equipamentos da rede. Mais especificamente, num teste realizado à configuração de QoS - criação de classes e políticas - verificou-se o resultado esperado. No caso testado, pretendia-se fazer a gestão do QoS num *router* Alcatel. No final do preenchimento do formulário presente na ferramenta, fez-se uma análise ao equipamento anteriormente referido e concluiu-se que os parâmetros definidos no preenchimento do formulário tiveram reflexo no equipamento, sendo apresentadas as classes e políticas definidas através da ferramenta desenvolvida, obtendo assim o sucesso desejado neste teste.

Com a apresentação destes testes demonstrou-se que os processos fundamentais são transparentes para o utilizador de alto-nível da ferramenta, pois o "grosso" da solução está em *back-end*. Pensa-se que do ponto de vista de quem a usa seja muito mais fácil e agradável a utilização de uma ferramenta gráfica do que a utilização da linha de comandos. Apesar de, teoricamente, o processo parecer diferente, conclui-se que, na prática, o processo é semelhante ao atual, com as devidas alterações e facilitismos de utilização.

4.3 Síntese

Concluindo este capítulo, nele apresentou-se o desenvolvimento realizado para a correta implementação de uma ferramenta que fizesse a gestão de uma rede e dos seus serviços, previamente à colocação da mesma em produção, caso assim se pretendesse.

Detalharam-se, também, algumas sequências de execução para uma melhor compreensão da execução do processo.

Seguidamente, apresentou-se um teste de execução da ferramenta, de modo a provar o seu funcionamento.

Por último, no capítulo final, apresenta-se a "Conclusão e Trabalho Futuro".

Conclusão e Trabalho Futuro

Inicialmente começou por se fazer uma análise às redes tradicionais como em alguns casos ainda hoje são utilizadas. Seguiu-se uma análise à tendência apresentada pela comunidade científica e as razões para uma melhoria no que toca à gestão de redes de comunicação. Feita essa análise, foi altura de aprofundar o conceito de SDN e analisar algumas tecnologias aí utilizadas. Depois desta análise do SDN, fez-se uma análise ao NFV, chegando-se à conclusão que ambas tendem a convergir, ou seja, a seguirem caminhos semelhantes. De seguida, foi feito um estudo de algumas soluções SDN já existentes ou que ainda estejam em desenvolvimento por parte de grandes empresas da área das redes.

Concluída esta análise do que já existia, propôs-se uma ferramenta *Open Source* que permite realizar a gestão de uma rede e respetivos serviços de um modo mais centralizado, graças à utilização do conceito de SDN. Com esta ferramenta, é possível testar uma rede, seja ela virtual ou real e já existente ou não, e os serviços que lá se pretende implementar antes de a rede ser colocada em produção. Este ponto dos serviços é um ponto fulcral, pois é com eles que as empresas de telecomunicações obtêm algum do seu lucro.

Com uma solução semelhante a esta, dá-se a possibilidade de fazer qualquer tipo de experiências na rede, mesmo que quem o pretenda fazer não possua profundos e bastante específicos conhecimentos de redes informáticas. Para isto ser possível basta, por exemplo, que a ferramenta seja adaptada à rede pretendida, ou seja, que o endereçamento e, caso necessário, o encaminhamento na rede estejam feitos.

Assim, pode-se evitar possíveis erros e falhas de configuração, tanto nos serviços testados como na própria rede. O facto de a solução ser mais automatizada permitirá facilitar alguns processos de configuração. Logo, pode-se canalizar o tempo despendido na criação ou otimização de serviços e/ou da rede para aspetos mais importantes, como a criação de novos serviços. Esta poupança de recursos leva, consequentemente, a uma poupança financeira pois, se um indivíduo realizar as mesmas ações em menos tempo, pode haver mais inovações, sendo assim mais produtivo para a empresa onde trabalha.

Os resultados da solução proposta foram bem sucedidos e apresentados nesta dissertação.

Naturalmente que esta ferramenta não é perfeita mas, sendo aproveitada, pode trazer benefícios a quem a utilize.

Além do melhoramento visual da ferramenta, da implementação de novos serviços e da utilização de equipamento de mais fabricantes, algumas das propostas para trabalho futuro podem ser:

- Execução de testes à ferramenta utilizando um cenário misto (com equipamentos reais e virtuais) e um cenário com equipamentos reais;
- Utilização de outra solução semelhante, que não a "*Cisco Network Service Orchestrator enabled by Tail-f*", de modo a testar as diferenças e verificar eventuais melhorias entre a utilização dessa outra solução e da ferramenta proposta;
- Utilização de outro modo de acesso à comunicação entre a ferramenta proposta e o NSO, ou seja, em vez de utilizar *scripts* em *bash*, utilizar *REST* ou mesmo NETCONF, que também é suportada;
- Homogeneização do processo de conversão das configurações. Tendo em conta que, nesta dissertação, se referiu que os NED convertem as configurações dependendo do dispositivo em que esses comandos fossem executados, e relembrando que, caso o equipamento suportasse NETCONF, essa conversão não seria necessária, esta homogeneização poderia ser interessante, isto é, se esta conversão fosse previamente feita, o processo de configuração poderia ser bastante útil e, quiçá, menos complexo e mais rápido no futuro. O ideal seria que todos os equipamentos falassem a mesma "língua", mas como isso não acontece, uma solução simplificada poderia ser bastante apelativa para ser aplicada na utilização de ferramentas semelhantes a esta;
- Utilização de outra ferramenta de simulação, alternativa ao GNS3, para o desenvolvimento do cenário de testes, pois a ferramenta com outro *software* pode ter outro tipo de respostas que não as previstas nesta dissertação ou pode necessitar de outro tipo de alterações na solução proposta.

Conclui-se assim a dissertação relativa ao SDN, na utilização de gestão de serviços de comunicação.

Bibliografia

6WIND. (13 de março de 2013). *SDN/NFV Resources*. Obtido em 27 de setembro de 2014, de Software Defined Networking: http://www.6wind.com/software-defined-networking/sdn-nfv-resources/

Alexander, S. (15 de dezembro de 2014). *2015: The year SDN and NFV go mainstream*. Obtido em 12 de janeiro de 2015, de Network World, Inc.: http://www.networkworld.com/article/2858736/sdn/2015-the-year-sdn-and-nfv-go-mainstream.html

Altice Labs. (2015). *An NFV/SDN Enabled Service Provider: A New Generation of Digital Services.* Altice Labs, Aveiro, Portugal. Obtido em 30 de janeiro de 2015, de http://www.alticelabs.com/content/WP-An-NFV-SDN-Enabled-Service-Provider.pdf

Amaral. (março de 2010). *Redes de computadores*. Obtido em 26 de janeiro de 2015, de https://jamaralnet.files.wordpress.com/2010/03/2semredes1.pdf

Avramov, L., & Portolani, M. (2015). *The Policy Driven Data Center with ACI: Architecture, Concepts, and Methodology*. Indianápolis, Indiana, Estados Unidos da América: Cisco Press. Obtido em 08 de março de 2016, de https://goo.gl/JdRwrF

Banks, E. (06 de janeiro de 2014). *SDN showdown: Examining the differences between VMware's NSX and Cisco's ACI*. Obtido em 27 de janeiro de 2015, de http://www.networkworld.com/article/2172922/sdn/sdn-showdown--examining-the-differences-between-vmware-s-nsx-and-cisco-s-aci.html

Barry, D. J. (15 de maio de 2014). *SDN and NFV: Faster Than the Future*. Obtido em 13 de outubro de 2014, de NFVZone: http://www.nfvzone.com/topics/nfv/articles/378817-sdn-nfv-faster-than-future.htm

Behera, S., & Desmidt, D. (07 de novembro de 2013). *Under the Hood: Network Virtualization with OpenStack Neutron and VMware NSX*. Obtido em 28 de janeiro de 2015, de https://www.openstack.org/assets/presentation-media/Openstack-Neutron+NSX-Final.pptx

Bode, D. (09 de novembro de 2010). *Ruby DSL*. Obtido em 19 de novembro de 2014, de Puppet & Ruby DSL: http://puppetlabs.com/blog/ruby-dsl

Casa dos Bits. (28 de julho de 2010). *PT escolhe Alcatel-Lucent para melhorar capacidade de rede*. Obtido em 26 de janeiro de 2015, de SapoTEK: http://tek.sapo.pt/noticias/telecomunicacoes/pt_escolhe_alcatel_lucent_para_melhorar_capac_1080754.html

Chiosi, M., Clarke, D., Willis, P., Reid, A., Feger, J., Bugenhagen, M., . . . Sen, P. (2012). An Introduction, Benefits, Enablers, Challenges & Call for Action . *SDN and OpenFlow World Congress*, (pp. 1-14). Obtido em 22 de outubro de 2014, de https://portal.etsi.org/nfv/nfv_white_paper.pdf

Cisco Systems, Inc. (2013). *Cisco and Citrix: Build Application Centric, Application Delivery Controller-Enabled Data Centers.* Obtido em 10 de março de 2016, de https://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/unified-fabric/solution-brief-c22-730002.pdf

Cisco Systems, Inc. (10 de dezembro de 2013). *Cisco Completes Acquisition of Insieme Networks*. Obtido em 20 de outubro de 2014, de Cisco's Technology News Site: http://newsroom.cisco.com/release/1298696

Cisco Systems, Inc. (2014). *Cisco Application Policy Infrastructure Controller (APIC)*. Obtido em 31 de janeiro de 2015, de https://www.cisco.com/c/en/us/products/cloud-systems-management/application-policy-infrastructure-controller-apic/index.html

Cisco Systems, Inc. (09 de julho de 2014). *Cisco Completes Acquisition of Tail-f Systems.* Obtido em 20 de outubro de 2014, de Cisco's Technology News Site.

Cisco Systems, Inc. (30 de maio de 2014). *Device Programmability Strategies.* Obtido em 24 de setembro de 2014, de https://pt.scribd.com/doc/258153080/Device-Programmability-Strategies

Cisco Systems, Inc. (2014). Is Cisco Application Centric Infrastructure an SDN Technology?, (pp. 16). Obtido em 07 de janeiro de 2016, de http://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/white-paper-c11-733456.pdf

Cisco Systems, Inc. (2014). *OpFlex: An Open Source Approach*. Obtido em 19 de novembro de 2014, de http://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/white-paper-c11-731304.pdf

Cisco Systems, Inc. (2014). *Tail-f Network Control System 3.3 Getting Started Guide*. Obtido em 26 de dezembro de 2015, de http://tailf.com/wordpress/wp-content/uploads/2014/12/ncs_getting_started.pdf

Cisco Systems, Inc. (06 de setembro de 2015). *Cisco Network Services Orchestrator*. Obtido em 25 de fevereiro de 2016, de https://www.cisco.com/c/en/us/products/cloud-systems-management/network-services-orchestrator/index.html

Cisco Systems, Inc. (2015). *OpFlex: An Open Policy Protocol*. Obtido em 19 de maio de 2015, de https://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/white-paper-c11-731302.pdf

Cisco Systems, Inc. (2015). *Tráfego global de dados móveis crescerá quase 10 vezes entre 2014 e 2019*. Obtido em 29 de fevereiro de 2016, de https://www.cisco.com/web/PT/press/articles/2015/20150203.html

Cisco Systems, Inc., Tail-f Systems. (2015). *Cisco Network Service Orchestrator (NSO) enabled by Tail-f*. Obtido em 22 de dezembro de 2015, de http://goo.gl/qEGx5M

Citrix. (2015). *Implementing Cisco Application Centric Infrastructure with Citrix NetScaler Application Delivery Controllers*. Obtido em 12 de janeiro de 2016, de https://www.citrix.com/content/dam/citrix/en_us/documents/products-solutions/implementing-cisco-application-centric-infrastructure-with-citrix-netscaler-application-delivery-controllers.pdf

Dignan, L. (02 de abril de 2013). *Alcatel-Lucent's Nuage launches SDN platform, courts IT*. Obtido em 15 de dezembro de 2014, de ZDNet: http://www.zdnet.com/article/alcatel-lucents-nuage-launches-sdn-platform-courts-it/

Dryden, P. (10 de julho de 1995). AT&T aids telecommuters. *Computer World*, 8. Obtido em 30 de setembro de 2014

Dürr, F. (12 de janeiro de 2014). *OpenDaylight: Programming Flows with the REST Interface and cURL*. Obtido em 26 de novembro de 2014, de Networked and Mobile Systems: http://www.frank-durr.de/?p=68

Enns, R., Bjorklund, M., Schoenwaelder, J., & Bierman, A. (junho de 2011). *RFC 6241 – NETCONF Configuration Protocol*. Obtido em 31 de dezembro de 2015, de https://tools.ietf.org/html/rfc6241

Evans, D. (15 de julho de 2011). *The Internet of Things [INFOGRAPHIC]*. Obtido em 10 de novembro de 2014, de Cisco Blogs: http://blogs.cisco.com/diversity/the-internet-of-things-infographic

Ferro, G. (25 de abril de 2012). *ONF Extends their SDN to Embrace NETCONF and YANG*. Obtido em 25 de novembro de 2014, de EtherealMind: http://etherealmind.com/onf-extends-their-sdn-to-embrace-netconf-and-yang/

Ferro, G. (04 de fevereiro de 2013). *New SDN Technology Challenges VMware, Cisco*. Obtido em 15 de dezembro de 2014, de Network Computing: http://www.networkcomputing.com/data-centers/new-sdn-technology-challenges-vmware-cisco/a/d-id/1234143

Ferro, G. (26 de agosto de 2013). *VMware NSX: Game Changer for Data Center Networks*. Obtido em 27 de janeiro de 2015, de http://www.networkcomputing.com/cloud-infrastructure/vmware-nsx-game-changer-for-data-center-networks/a/d-id/1234378

Ferro, G. (15 de março de 2013). *VMware's NSX End Game Is Hybrid Clouds*. Obtido em 28 de janeiro de 2015, de http://www.networkcomputing.com/networking/vmwares-nsx-end-game-hybrid-clouds/900481468

Ferro, G. (06 de novembro de 2014). *Musing: On Using Cisco ACI and VMware NSX in the same network*. Obtido em 27 de janeiro de 2015, de http://etherealmind.com/musing-using-cisco-aci-vmware-nsx-network/

Gouveia, R., Aparício, J., Parreira, B., Sargento, S., & Carapinha, J. (2013). SDN Framework for Connectivity Services. Em A. Pereira, C. Rabadão, P. Domingues, & V. Fernandes (Ed.), *CRC'13 - Atas da 13ª Conferência sobre Redes de Computadores*, (pp. 19-24). Leiria, Portugal. doi:978-972-8793-62-3

Grossner, C. (29 de julho de 2014). *87% of medium and large N. American enterprises surveyed by Infonetics intend to have SDN live in the data center by 2016*. Obtido em 09 de janeiro de 2015, de Infonetics Research: http://www.infonetics.com/pr/2014/sdn-strategies-survey-highlights.asp

Hewlett-Packard Development Company, L.P. (2013). *HP Virtual Application Networks SDN Controller*. Hewlett-Packard Development Company, L.P. Obtido em 31 de dezembro de 2014, de http://h17007.www1.hp.com/docs/networking/solutions/sdn/4AA4-8807ENW.PDF

Higginbotham, S. (02 de abril de 2014). *Google launches Andromeda, a software defined network underlying its cloud*. Obtido em 06 de março de 2015, de Gigaom, Inc.: https://gigaom.com/2014/04/02/google-launches-andromeda-a-software-defined-network-underlying-its-cloud/

Howard, M. (05 de novembro de 2014). *Infonetics forecasts carrier SDN and NFV market to reach $11 billion by 2018*. Obtido em 09 de janeiro de 2015, de Infonetics Research: http://www.infonetics.com/pr/2014/Carrier-SDN-NFV-Market-Highlights.asp

Huawei Technologies Co., Ltd. (2013). *Reshaping the Future of Network Architecture*. Huawei Technologies Co., Ltd. Obtido em 15 de dezembro de 2014, de http://goo.gl/o6aTDK

IBM. (04 de fevereiro de 2014). *New IBM Virtualization Solution to Speed Software Defined Networking Deployments*. Obtido em 14 de fevereiro de 2015, de https://www-03.ibm.com/press/us/en/pressrelease/43082.wss

Juniper Networks, Inc. (2015). *Junos OS NETCONF XML Management Protocol Developer Guide*. Obtido em 21 de fevereiro de 2016, de https://www.juniper.net/documentation/en_US/junos13.2/information-products/pathway-pages/netconf-guide/netconf.pdf

Kiran, S. (2 de abril de 2014). *Introducing OpFlex – A new standards-based protocol for Application Centric Infrastructure*. Obtido de Cisco Blogs: https://blogs.cisco.com/datacenter/introducing-opflex-a-new-standards-based-protocol-for-application-centric-infrastructure

Kreutz, D., Ramos, F., Verissimo, P., Rothenberg, C., Azodolmolky, S., & Uhlig, S. (08 de outubro de 2014). *Software-Defined Networking: A Comprehensive Survey*. Obtido em 24 de fevereiro de 2015, de http://arxiv.org/pdf/1406.0440v3.pdf

Lawson, S. (03 de Fevereiro de 2011). Update: ICANN assigns its last IPv4 addresses. *Computerworld, Inc.* Obtido em 05 de Dezembro de 2015, de http://goo.gl/Zzdw3K

McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., Turner, J. (2008). *OpenFlow: Enabling Innovation in Campus Networks.* Universidade de Stanford; Universidade de Washington; MIT; Universidade de Princeton;

Universidade Washington, em St. Louis; Universidade da Califórnia, em Berkeley. Obtido em 19 de novembro de 2014, de http://archive.openflow.org/documents/openflow-wp-latest.pdf

Mehra, R., & Casemore, B. (2013). *The Impact of SDN on Datacenter and Enterprise Network Architectures.* Massachusetts, Estados Unidos da América. Obtido em 31 de dezembro de 2014, de http://h17007.www1.hp.com/docs/networking/solutions/sdn/4AA4-8807ENW.PDF

Nadeau, T. D., & Gray, K. (2013). *SDN: Software Defined Networks* (1ª ed., Vol. 1). (M. Loukides, & M. Blanchette, Edits.) Estados Unidos da América: O'Reilly Media, Inc. Obtido em 26 de janeiro de 2015

Naguib, H. (13 de março de 2013). *VMware NSX Network Virtualization*. Obtido em 28 de janeiro de 2015, de VMware Blogs: http://blogs.vmware.com/tribalknowledge/2013/03/vmware-nsx-network-virtualization.html

Nuage Networks. (março de 2013). *Unconstrained Datacenter Networks for the Cloud Era*. Nuage Networks. Obtido em 15 de dezembro de 2014, de http://www.nuagenetworks.net/wp-content/uploads/2014/11/2013-03-28-Nuage-White-Paper_final_r2.pdf

Nuage Networks. (16 de julho de 2014). *Nuage Networks: True Potential of Network Virtualization*. Obtido em 08 de março de 2016, de SlideShare: http://pt.slideshare.net/nuage-networks/nuage-networksgigaomstructuresneddonsdn

Nuage Networks. (2014). *Virtualized Services Platform*. Obtido em 15 de dezembro de 2014, de http://www.nuagenetworks.net/wp-content/uploads/2014/11/MKT2014097652EN_NN_VSP_Virtualized_Services_Platform_R3_Datasheet2.pdf

OFELIA. (2012). *OFELIA - The EU FP7 Project and The European OpenFlow Experimental Facility*. Obtido em 06 de março de 2015, de http://www.fp7-ofelia.eu/assets/Publications-and-Presentations/OFELIAFebruary2013.pdf

OFELIA. (2012). *What is OpenFlow? What does OFELIA? An Introduction to OpenFlow and what OFELIA has to do with it*. Obtido em 06 de março de 2015, de http://www.fp7-ofelia.eu/assets/Uploads/About-OpenFlow-OFELIA.pdf

Open Networking Foundation. (13 de abril de 2012). *Software-Defined Networking: The New Norm for Networks*. Obtido em 27 de setembro de 2014, de Open Networking Foundation:

https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf

Open Networking Foundation. (2013). *SDN Architecture Overview.* Obtido em 08 de março de 2016, de https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/SDN-architecture-overview-1.0.pdf

OpenDaylight. (17 de abril de 2013). *OpenDaylight SDN Controller Platform (OSCP):Rest Reference*. Obtido em 26 de novembro de 2014, de OpenDaylight Wiki: https://wiki.opendaylight.org/view/OpenDaylight_SDN_Controller_Platform_%28OSCP%29:Rest_Reference

Palmer, M. (06 de janeiro de 2014). *SDN and NFV Technology Trends to Watch in 2014*. Obtido em 20 de outubro de 2014, de SDxCentral: https://www.sdxcentral.com/articles/news/sdn-nfv-technology-trends-watch-2014/2014/01/

Pate, P. (30 de março de 2013). *NFV and SDN: What's the Difference?* Obtido em 16 de outubro de 2014, de SDxCentral: https://www.sdncentral.com/technology/nfv-and-sdn-whats-the-difference/2013/03/

Pepelnjak, I. (28 de junho de 2012). *We need both OpenFlow and NETCONF*. Obtido em 16 de fevereiro de 2015, de http://blog.ipspace.net/2012/06/we-need-both-openflow-and-netconf.html

Pfaff, B., & Davie, B. (dezembro de 2013). *The Open vSwitch Database Management Protocol*. Obtido em 27 de janeiro de 2015, de https://tools.ietf.org/html/rfc7047

Pinote, A., & Martins, J. L. (2013). Encaminhamento IP Optimizado Através de uma Aproximação de Software Defined Networking. Em A. Pereira, C. Rabadão, P. Domingues, & V. Fernandes (Ed.), *CRC'13 - Atas da 13ª Conferência sobre Redes de Computadores*, (pp. 109-114). Leiria, Portugal. doi:978-972-8793-62-3

Portugal Telecom. (26 de novembro de 2010). *PT faz a primeira ligação a 100Gbps em Portugal*. Obtido em 26 de janeiro de 2015, de PT Media: https://goo.gl/YnBN0u

Prigg, M. (22 de janeiro de 2015). *Google's Eric Schmidt claims the 'internet will disappear' as everything in our life gets connected*. Obtido em 23 de fevereiro de 2015, de Daily Mail: http://www.dailymail.co.uk/sciencetech/article-2922460/Google-s-Eric-Schmidt-claims-internet-disappear-life-gets-connected-says-robots-WON-T-world.html

RESTful control of switches. (17 de agosto de 2013). Obtido em 26 de novembro de 2014, de sFlow - SDN analytics and control using sFlow® standard: http://blog.sflow.com/2013/08/restful-control-of-switches.html

Rouse, M. (setembro de 2013). *OVSDB (Open vSwitch Database Management Protocol)*. Obtido em 27 de janeiro de 2015, de http://searchsdn.techtarget.com/definition/OVSDB-Open-vSwitch-Database-Management-Protocol

Schaefer, S. (21 de janeiro de 2015). *Netflix Doubles Down On Global Growth Bet, Original Content*. Obtido em 23 de janeiro de 2015, de Forbes: http://www.forbes.com/sites/steveschaefer/2015/01/21/netflix-doubles-down-on-global-growth-bet-original-content/

Schroeder, E. (29 de maio de 2014). *Telecom Italia Creates SDN Network Testbed with Five Universities*. Obtido em 09 de dezembro de 2014, de Cisco Blogs: http://blogs.cisco.com/sp/telecom-italia-creates-sdn-network-testbed-with-five-universities

SDxCentral. (agosto de 2013). *What is NFV – Network Functions Virtualization?* Obtido em 16 de outubro de 2014, de SDxCentral: https://www.sdncentral.com/whats-network-functions-virtualization-nfv/

SDxCentral. (janeiro de 2014). *What are SDN Controllers?* Obtido em 11 de março de 2015, de https://www.sdxcentral.com/resources/sdn/sdn-controllers/

SDxCentral. (janeiro de 2014). *What are SDN Southbound APIs?* Obtido em 11 de março de 2015, de https://www.sdxcentral.com/resources/sdn/southbound-interface-api/

SDxCentral. (setembro de 2014). *What is a Floodlight Controller?* Obtido em 11 de março de 2015, de https://www.sdxcentral.com/resources/sdn/sdn-controllers/open-source-sdn-controllers/what-is-floodlight-controller/

SDxCentral. (julho de 2014). *What is Cisco OpFlex?* Obtido em 20 de novembro de 2014, de SDxCentral: https://www.sdxcentral.com/resources/cisco/cisco-opflex/

SDxCentral. (2014). *What is OpenStack Neutron?* Obtido em 14 de novembro de 2014, de SDxCentral: https://www.sdxcentral.com/resources/open-source/what-is-openstack-quantum-neutron/

SDxCentral. (2014). *Which is better - SDN or NFV?* Obtido em 7 de outubro de 2014, de SDxCentral: https://www.sdxcentral.com/resources/nfv/which-is-better-sdn-or-nfv/

Shaikh, K. (10 de setembro de 2013). *HP-VMware Networking Solution Market Reactions*. Obtido em 13 de janeiro de 2015, de HP Blogs: http://community.hpe.com/t5/Networking/HP-VMware-Networking-Solution-Market-Reactions/ba-p/6791335

Silva, J. V., & Santos, J. P. (2014). *Gestão de rede recorrendo a Software Defined Networking e OpenFlow - uma abordagem Opensource.* Escola Superior de Tecnologia e Gestão, Instituto Politécnico de Leiria, Leiria. Obtido em 26 de janeiro de 2015

Simões, P. (11 de abril de 2013). *Tráfego móvel está a crescer em Portugal*. Obtido em 10 de outubro de 2014, de pplware - No Comments: http://pplware.sapo.pt/internet/trafego-movel-esta-a-crescer-em-portugal/

Smith, M., Dvorkin, M., Laribi, Y., Pandey, V., Garg, P., & Weidenbacher, N. (02 de abril de 2014). *OpFlex Control Protocol*. (Internet Engineering Task Force) Obtido em 24 de novembro de 2014, de https://tools.ietf.org/html/draft-smith-opflex-00

Stallings, W. (março de 2013). Software-Defined Networks and OpenFlow. *The Internet Protocol Journal*, 2-14. Obtido em 13 de outubro de 2014, de https://www.box.com/shared/static/13xnhpmvwenlm2ahy1yp.pdf

Swan, C. (23 de junho de 2014). *Can Facebook's different take on SDN scale down from hyper?* Obtido em 13 de março de 2015, de https://thestack.com/iot/2014/06/23/can-facebooks-different-take-on-sdn-scale-down-from-hyper/

Townsend, K. (08 de setembro de 2013). *Is Cisco fighting a losing battle over SDN?* Obtido em 13 de outubro de 2014, de VirtualizedGeek: http://www.virtualizedgeek.com/2013/09/is-cisco-fighting-a-losing-battle-over-sdn/

Townsend, K. (09 de outubro de 2014). *Cisco supports VMware NSX, though customers really want integration*. Obtido em 27 de 01 de 2015, de http://www.techrepublic.com/article/cisco-supports-vmware-nsx-though-customers-really-want-integration/

Townsend, K. (09 de março de 2014). *Understanding the competition between VMware and Cisco on SDN*. Obtido em 28 de janeiro de 2015, de https://www.linkedin.com/pulse/20140309231647-31940198-understanding-the-competition-between-vmware-and-cisco-on-sdn

Vahdat, A. (02 de abril de 2014). *Enter the Andromeda zone - Google Cloud Platform's latest networking stack*. Obtido em 03 de março de 2015, de Google Cloud Platform Blog: http://googlecloudplatform.blogspot.pt/2014/04/enter-andromeda-zone-google-cloud-platforms-latest-networking-stack.html

Vervloesem, K. (2 de agosto de 2010). *Puppet – server management made easy*. Obtido em 19 de novembro de 2014, de Linux User & Developer: http://www.linuxuser.co.uk/tutorials/puppet-server-management

Yang, F. (09 de outubro de 2013). *SDN - The New Problem Solver*. Obtido em 04 de novembro de 2014, de CommScope Inc.: http://www.commscope.com/Blog/SDN--The-Problem-Solver/

Anexos

Anexo 1

Na tabela seguinte estão listadas algumas soluções SDN e respetivas empresas que as desenvolveram.

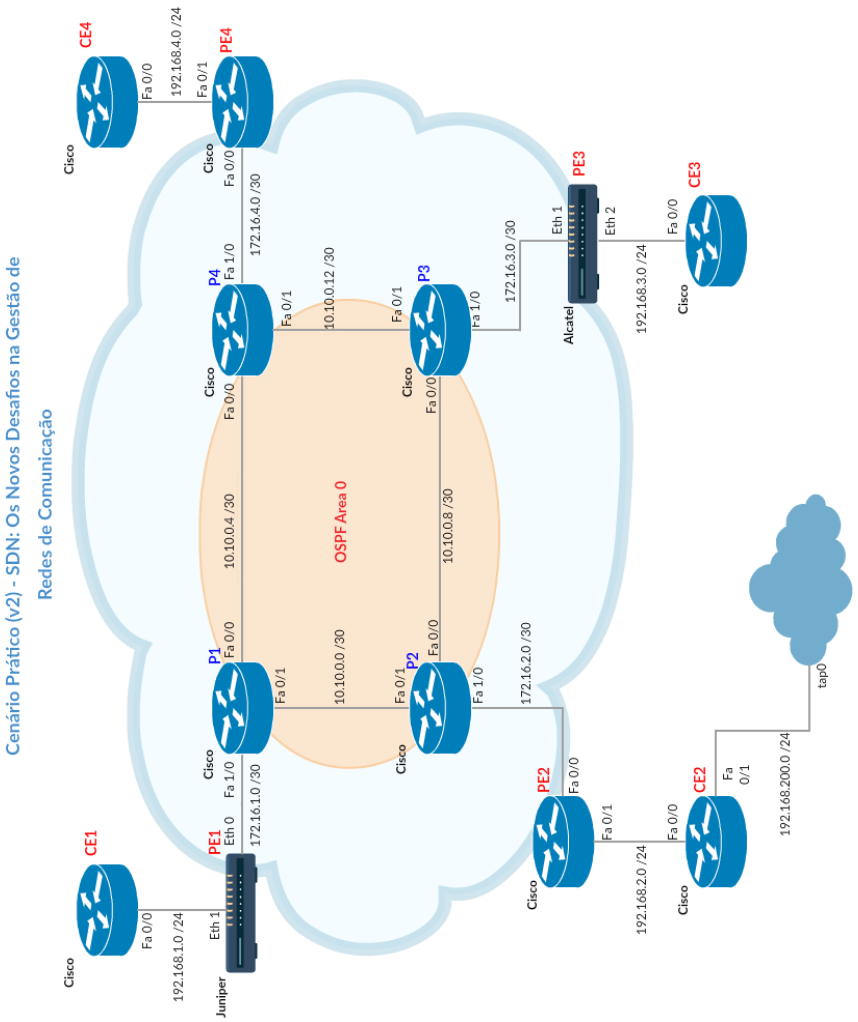
Empresa	**Proposta**	**Notas**	**Referência**
Cisco	*Cisco Network Service Orchestrator (enabled by Tail-f)*	Lançamento previsto no final de 2015	https://goo.gl/okaRJu; http://goo.gl/qEGx5M
VMware	*VMware NSX*	Em parceria com a HP a nível de monitorização, *troubleshooting* e gestão.	http://goo.gl/TjmMo1; http://goo.gl/KAfD0E
Juniper Networks	*Contrail*		http://goo.gl/G80asx
Big Switch Networks	*Big Network Controller*		http://goo.gl/9VHxWy
Alcatel-Lucent (Nuage Networks)	*Nuage Networks Virtualized Services Platform (VSP)*		http://goo.gl/C5el2H; http://goo.gl/IXIsVd
IBM		Várias soluções em conjunto com outras empresas	http://goo.gl/0PmjU9; https://goo.gl/wVA2QA

Huawei	*SoftCOM*		http://goo.gl/o6aTDK
Arista EOS	*Arista Software Driven Cloud Networking* (SDCN)		http://goo.gl/VoPWIb
Palo Alto Networks		Tem parcerias com Nuage Networks e Big Switch Networks, VMware, Citrix, Arista, entre outras, para oferecer segurança à arquitetura da rede SDN destas empresas	https://goo.gl/86rNF0 (VMware); https://goo.gl/RRrjzz (Arista), https://goo.gl/5MSyPA (Citrix), http://goo.gl/KJpLf1 (Nuage Networks)
Citrix	*Citrix NetScaler*		https://goo.gl/umlFCp

Anexo 2

Na imagem abaixo está representado o cenário e endereçamento completo da rede virtual criada em GNS3.

Dispositivo	Interface	IP	Máscara
P1	Fa 0/0	10.10.0.5	255.255.255.252
	Fa 0/1	10.10.0.1	255.255.255.252
	Fa 1/0	172.16.1.1	255.255.255.252
P2	Fa 0/0	10.10.0.9	255.255.255.252
	Fa 0/1	10.10.0.2	255.255.255.252
	Fa 1/0	172.16.2.1	255.255.255.252
P3	Fa 0/0	10.10.0.10	255.255.255.252
	Fa 0/1	10.10.0.13	255.255.255.252
	Fa 1/0	172.16.3.1	255.255.255.252
P4	Fa 0/0	10.10.0.6	255.255.255.252
	Fa 0/1	10.10.0.14	255.255.255.252
	Fa 1/0	172.16.4.1	255.255.255.252
PE1	Eth 0	172.16.1.2	255.255.255.252
	Eth 1	192.168.1.1	255.255.255.0
PE2	Fa 0/0	172.16.2.2	255.255.255.252
	Fa 0/1	192.168.2.1	255.255.255.0
PE3	Eth 0	172.16.3.2	255.255.255.252
	Eth 2	192.168.3.1	255.255.255.0
PE4	Fa 0/0	172.16.4.2	255.255.255.252
	Fa 0/1	192.168.4.1	255.255.255.0
CE1	Fa 0/0	192.168.1.2	255.255.255.0
CE2	Fa 0/0	192.168.2.2	255.255.255.0
	Fa 0/1	192.168.200.1	255.255.255.0
CE3	Fa 0/0	192.168.3.2	255.255.255.0
CE4	Fa 0/0	192.168.4.2	255.255.255.0
Cloud	tap0	192.168.200.2	255.255.255.0



Anexo 3

De seguida estão apresentadas as tabelas criadas de base de dados.

NOME	TIPO
ID	int(11)
VPN_ID	int(11)
ENDPOINT	varchar(50)
EQUIPCE	varchar(50)
INTERCE	varchar(50)
REDE	varchar(50)
MASC	int(11)
LARGBANDA	int(11)

Tabela NCS_EndPoint

NOME	TIPO
ID	int(11)
NOME	varchar(20)

Tabela NCS_EQUIP

NOME	TIPO
ID	int(11)
FABRICANTE	varchar(100)
NOME	varchar(50)
USERNAME	varchar(50)
PASS	varchar(50)
PASS2	varchar(50)

Tabela NCS_Grupo

NOME	TIPO
ID	int(11)
EQUIP_ID	int(11)
TIPO	varchar(10)
NOME	varchar(50)
IP	varchar(20)
IPMASK	varchar(20)

Tabela NCS_INT

NOME	TIPO
ID	int(11)
EQUIP_ID	varchar(50)
TIPO	varchar(50)
NOME	varchar(50)
IP	varchar(20)
IPMASK	int(5)
IPDEST	varchar(20)
MASCDEST	int(5)
PORTOORIG	int(5)
PORTODEST	int(5)
PROTOCOLO	varchar(50)

Tabela NCS_QoS_CLASSE

NOME	TIPO
ID	int(11)
ID_CLASSE	int(11)
ID_POL	int(11)
PERCLB	int(11)
PRIORIDADE	int(11)

Tabela NCS_QoS_CLASSE_POL

NOME	TIPO
ID	int(11)
NOME	varchar(50)

Tabela NCS_QoS_POL

NOME	TIPO
ID	int(11)
NOME	varchar(50)
EQUIP1	varchar(50)
INT1	varchar(50)
IP1	varchar(50)
MASC1	int(11)
EQUIP2	varchar(50)
INT2	varchar(50)
IP2	varchar(50)
MASC2	int(11)
LINKVLAN	int(11)

Tabela NCS_TOPOL

NOME	TIPO
ID	int(11)
NOME	varchar(50)
LINKVLAN_ID	varchar(11)
EQUIP	varchar(50)
INTERFACE	varchar(50)

Tabela NCS_VLAN

NOME	TIPO
ID	int(5)
NOME	varchar(50)
NUMAS	int(5)

Tabela NCS_VPN