

# Controladores de Redes Definidas por Software: Caracterización comparativa

## Controller software-defined networking: Characterization comparative

F.R. Cuesta Q., D. Rico B., *Member*, INGAP - UFPSO

**Abstract**— Conventional networks which at the time maintained an added value today after the need for change have managed to evolve with other protocols and architectures in such networks. Software Defined Networks or SDN (Software Defined Network for its acronym in English) refers to a network architecture that allows drivers to get through automatable networks. This article develops the architecture of SDN and some types of drivers both commercial and open source.

**Keywords**— SDN, OpenFlow, Virtualization Controller

### INTRODUCCIÓN

Las redes convencionales manejan diversos protocolos y componentes, que fueron agregados, de acuerdo a las necesidades que surgieron a lo largo del tiempo. Esa arquitectura de redes, a pesar de atender a las necesidades actuales, se ha vuelto muy compleja, y de cierta manera, poco flexible para continuar evolucionando a la velocidad del surgimiento de nuevas aplicaciones. Los sistemas de próxima generación requerirán un cambio de paradigma en la forma en que se construyen y gestionan, las plataformas de control y de gestión convencionales se enfrentan a retos considerables con respecto a la flexibilidad, la fiabilidad y la seguridad de que los sistemas de próxima generación tienen que manejar. Una manera de reducir esta complejidad es a través de las redes definidas por Software ya que son un enfoque emergente que busca dar solución a problemas de seguridad, flexibilidad y optimización de las redes tradicionales [1]. La separación del plano de datos y el plano de control, la extracción de este último a un sistema centralizado programable denominado controller y la abstracción de los recursos facilita la

adopción de aplicaciones que se comportan como los protocolos establecidos en las redes tradicionales, con la diferencia de que pueden ser personalizados por programadores, operadores de red o por sistemas automatizados [2].

Debido a este tipo de eventos en el 2005 en la Universidad de Stanford se inició el desarrollo de un nuevo enfoque basado en la centralización del control de toda la red, en su automatización a nivel de software y en disminuir las tareas de los dispositivos intermedios (switches y routers) a solo reenvío de paquetes [3][4]. Esto otorga a la red características flexibles y programables, además, abre las puertas a desarrollos de nuevos servicios [2], este novedoso enfoque es llamado SDN (Redes definidas por Software) [5].

### ARQUITECTURA SDN

En la figura número 1 se evidencia las tres capas que constituyen la arquitectura de las redes definidas por software, estas capas interactúan a través de unas APIs que permiten una mayor flexibilidad, fiabilidad y seguridad elementos trascendentales en las nuevas redes emergentes.

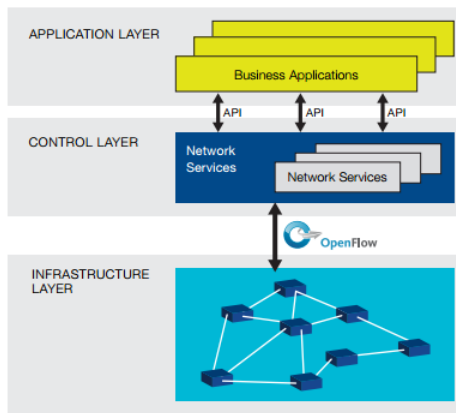


Fig. 1. Gráfica de la arquitectura de la SDN [6]

- **Capa de infraestructura.** Es la capa más baja de SDN donde se encuentran los dispositivos físicos o virtuales como (switch, routers) los cuales están conectados a través de una interfaz abierta que permite el switcheo y envío de paquetes en una conexión de red. Estas conexiones se hacen a través de medios de transmisión diferentes, incluyendo, cobre, redes inalámbricas, y fibra óptica. Si bien las preocupaciones básicas están asociadas a esta capa ya que dependen de estos dispositivos para el correcto funcionamiento de las SDN.
- **Capa de control.** Es la capa central de la arquitectura SDN, esta capa permite controlar un conjunto abstracto de recursos de plano de control, es decir, es la entidad que controla y configura los nodos de red para dirigir correctamente los flujos de tráfico. El controlador SDN elimina la inteligencia de conmutación y encaminamiento de datos de los nodos que realizan dicha función, pasando al controlador SDN, que toma esas decisiones y selecciona el mejor camino para el tráfico.
- **Capa de aplicación.** Consiste en las aplicaciones de negocio de los usuarios finales, que utilizan servicios de comunicación de SDN a través de las API hacia arriba (*northbound*) de la capa de control, tales como REST, JSON, XML, etc., permite a los servicios y aplicaciones simplificar y automatizar las tareas de configuración, provisión y gestionar nuevos servicios en la red, ofreciendo a

los operadores nuevas vías de ingresos, diferenciación e innovación.

## CONTROLADOR

En la redes SDN, la inteligencia del tráfico (en forma lógica) se encuentra centralizado en controladores; los controladores son “un mecanismo que permite a un servidor gestionar e interactuar con las tablas de flujo en los *switches*” que permiten concentrar la operatividad de los conmutadores; en su sistema operativo pueden configurarse nuevas funcionalidades a través de Interfaces de Programación de Aplicaciones (Application Programming interface: API: por sus siglas en inglés).

Los controladores permiten el control a través de las APIs conocidas: southbound, northbound, and east-west [7]. Las interfaces *northbound* permiten la comunicación con componentes de comunicación de alto nivel; las interfaces *southbound* permiten la comunicación con componentes de bajo nivel. [8]

La figura número 2 muestra la interacción entre los planos de control y datos a través del protocolo openflow

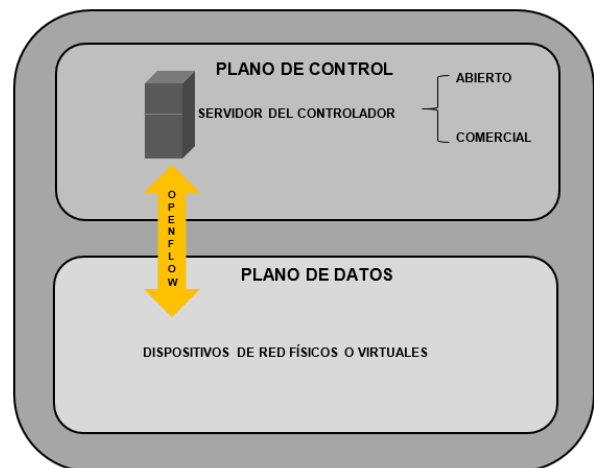


Figura 2. Arquitectura SDN. Fuente: Elaborada por los autores, mediante Microsoft Office Word.

## Algunos Controladores existentes

### Controladores comerciales [9]

- **Infraestructura de Políticas de Aplicaciones (APIC) de Cisco.** Es un cluster de controladores, centraliza las funciones en una API, con un repositorio

de datos globales y repositorio de datos de políticas.

- **El controlador SDN Virtual Application Networks (VAN) de HP.** Se ejecuta en switches habilitados para OpenFlow con el centro de datos; permitiendo un control centralizado y automatización, para la integración de la red y el sistema de negocios. Puede establecerse clusters del controlador para respaldar las funciones si en algún momento alguno falla.
- **El controlador ProgrammableFlow PF6800 de NEC.** Proporciona un punto de control para las redes físicas y de gestión para las redes virtuales y físicas.
- **El Controlador de Servicios Virtualizados (VSC) de Nuage Networks.** Establece un directorio de servicios con políticas basado en roles basados en las reglas de la red. Pueden añadirse inquilinos siendo detectados por el controlador cuando se conectan o se desconectan.
- **El controlador NSX de VMware.** Es un sistema de gestión distribuido que usa APIs northbound mientras mantiene información de las máquinas virtuales, hosts, switches lógicos y VXLANs se considera un sistema de gestión de estado distribuido que controla las redes virtuales y superpone los túneles de transporte.

#### Controladores de código abierto.

- **Beacon [10]** Es un controlador programado en Java, presenta un interfaz de usuario y visualización de la red. Por sus características Open Source puede manipularse el código.
- **Pox.** Es un controlador multiplataforma desarrollado en lenguaje Python.
- **Floodlight.** Controlador desarrollado en Java con la capacidad de manejar paquetes y tablas de flujo.
- **Nox.** Es un controlador desarrollado en C++ para administrar el flujo de datos,

con la capacidad de manejar grandes cantidades de flujos de datos.

- **Trema.** Es un controlador que tiene integrado un emulador de red *Openflow* que hace no necesario equipos de usuario final para aprobar las aplicaciones

La tabla número 1 muestra algunos parámetros que permiten realizar una comparativa de controladores de código abierto de acuerdo a sus características más relevantes.

Controlador	Parámetros comparativos				
	Desarrollo	Plataforma	Openflow	Virtualización	OpenStack
Nox	C++	Linux	Si	Mininet y Open ySwitch	No
Pox	Python	Linux, Mac OS, Windows	Si	Mininet y Open ySwitch	No
Beacon	Java	Linux, Mac OS, Windows y para Android móviles	Si	Mininet y Open ySwitch	No
Floodlight	Java	Linux, Mac OS, Windows	Si	Mininet y Open ySwitch	Si
Trema	Ruby/C	Linux	Si	Construcción de una herramienta virtual de simulación	Si

Tabla 1. Parámetros comparativos

#### OPENFLOW

Los switches Ethernet contienen tablas de flujo (flow-tables), que aunque son diferentes para cada fabricante, tiene un conjunto de funciones genéricos, que pueden ser explotados por un protocolo conocido como OpenFlow, usado en las capas de control de las SDN. Contando con la ventaja de ser un protocolo abierto que permite trabajar sobre varios *switches* y *routers*.

Con OpenFlow los flujos de la red pueden ser controlados para definir las rutas en las cuales los paquetes serán tratados, lo que permite flexibilidad establecido reduciendo la arbitrariedad de la transmisión de los paquetes. Dentro de los componentes de las tablas de flujo se encuentran: Match fields que permite detectar los paquetes conformada por sus puertos de entrada y cabeceras; priority para indicar la preferencia de flujo; counters para actualizar los paquetes coincidentes; instructions para modificar las acciones de procesamiento; y el timeout para señalar el tiempo de expiración del flujo [11].

#### MININET

Es una plataforma de emulación de red, que crea redes definidas por software totalmente escalables que están contenidas en una PC que utiliza procesamiento Linux. Permite crear, interactuar y personalizar un prototipo de red definido mediante software.

## Conclusiones

Establecer una red SDN es importante para el desarrollo de redes que puedan crecer de forma constante sin implicar un aumento en los costos y arriesgar el flujo de paquetes de la red organizacional. El desarrollo de ésta tecnología ha permitido mejorar la gestión por la gran adaptabilidad del software sin importar el creador de la arquitectura del dispositivo físico; para lo cual es importante conocer las capas que conforman una SDN como la infraestructura, el control y la capa de aplicación.

Entendiendo que las redes definidas por software ofrecen las características de escalabilidad, en el momento de elegir un controlador, podría elegirse uno comercial, por su mayor soporte técnico al contar con un departamento técnico de evaluación de product; sin embargo uno de código abierto ofrece la ventaja de manipular el código de la configuración que permitiría una mejor capacidad para ensayar con la implementación

## Referencias bibliográficas

- [1] OPEN NETWORKING FOUNDATION, «Software-Defined Networking: The New Norm for Networks,» ONF White Paper, Palo Alto, 2012.
- [2] S. Seker, S. Scott-Hayward, C. P. Kaur, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller y N. Rao, «Are we ready for SDN? Implementation challenges for softwaredefined networks,» Communications Magazine, IEEE, vol. 51, n° 7, pp. 36-43, 2013.
- [3] M. Casado, T. Garfinkel, A. Akella, M. J. Freedman, D. Boneh, N. McKeown y S. Shenker, «SANE: a protection architecture for enterprise networks,» de USENIXSS'06: Proceedings of the 15th conference on USENIX Security Symposium - Vol 15, Berkeley, 2006.
- [4] Luo, Jianying et al. "Prototyping fast, simple, secure switches for etha." 15th Annual IEEE Symposium on High-Performance Interconnects (HOTI 2007) 22 Aug. 2007: 73-82.
- [5] [5] N. Feamster, J. Rexford and E. Zegura, "The Road to SDN", *Queue*, vol. 11, no. 12, pp. 20-40, 2013.
- [6] OPEN NETWORKING FOUNDATION, *OpenFlow-enabled SDN and Network Functions Virtualization*, 1st ed. ONF Solution Brief, 2014.
- [7] Y. Jararweh, M. Al-Ayyoub, A. Darabseh, E. Benkhelifa, M. Vouk and A. Rindos, "Software defined cloud: Survey, system and evaluation", *Future Generation Computer Systems*, vol. 58, pp. 56-74, 2016.
- [8] "What is northbound interface / southbound interface? - Definition from WhatIs.com", *WhatIs.com*, 2016. [Online]. Available: <http://whatIs.techtarget.com/definition/northbound-interface-southbound-interface>. [Accessed: 01-Jul- 2016].
- [9] "Cinco controladores SDN comerciales que hay que conocer", *SearchDataCenter en Español*, 2016. [Online]. Available: <http://searchdatacenter.techtarget.com/es/cronica/Cinco-controladores-SDN-comerciales-que-hay-que-conocer>. [Accessed: 01-Jul- 2016].
- [10] D. Erickson, "Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking", in *The Beacon Openflow Controller*, Hong Kong, China, 2013, p. 13.
- [11] "Investigación de redes SDN (parte III). La Arquitectura SDN", *principia tecnologica*, 2014. [Online]. Available: <http://principletechnologica.com/2014/03/25/investigacion-de-redes-sdn-parte-iii-la-arquitectura-sdn/>. [Accessed: 01-Jul- 2016].
- [12] A. García Centeno, C. Rodríguez Vergel, C. Anías Calderón and F. Casmartíño Bondarenko, "Controladores SDN, elementos para su selección y evaluación.", *Revista Digital de las Tecnologías de la información y las comunicaciones Telematica*, vol. 13, no. 3, 2014.
- [13] M. Team, "Mininet: An Instant Virtual Network on your Laptop (or other PC) - Mininet", Yuba.stanford.edu, 2016. [Online]. Available: <http://yuba.stanford.edu/foswiki/bin/view/OpenFlow/Mininet>. [Accessed: 25-Jul- 2016].
- Wilkins, S. (2014). *Software Defined Networking (SDN) Controllers - A Guide To Software Defined Networking (SDN) Solutions*. Tom's IT Pro. Retrieved 31 May 2016, from

<http://www.tomsitpro.com/articles/software-defined-networking-solutions,2-835-2.html>  
SDN: Cómo el nuevo universo trazado por las redes definidas por software impactará en los negocios. (2014) (1st ed., pp. 1-16). Disponible en  
[http://www.la.logicalis.com/globalassets/latin-america/advisors/es/advisor\\_sdn.pdf](http://www.la.logicalis.com/globalassets/latin-america/advisors/es/advisor_sdn.pdf)  
<http://www.sciencedirect.com/science/article/pii/S1389128616300688>  
<http://www.sciencedirect.com/science/article/pii/S1084804516300297>



**Fabian Cuesta Quintero.**

Ingeniero de Sistemas. Especialista en práctica docencia universitaria. Grupo de Investigación en Ingenierías Aplicadas para la Innovación, la Gestión y el Desarrollo - INGAP. Universidad Francisco de Paula Santander Ocaña.



**Dewar Rico Bautista.**

Ingeniero de Sistemas, Especialista en Telecomunicaciones, Maestría en Ciencias Computacionales. Grupo de Investigación en Ingenierías Aplicadas para la Innovación, la Gestión y el Desarrollo - INGAP. Docente Universidad Francisco de Paula

Santander Ocaña. Colombia.