

# Explorando las redes definidas por software (SDN)

Diego R. Rodríguez Herlein, Carlos A. Talay, Claudia N. González

Campus Universitario – oficina 18/ Dpto. Ciencias Exactas e Informática UARG / UNPA  
(dherlein, ctalay, cgonzalez)@uarg.unpa.edu.ar

Luis A. Marrone

L.I.N.T.I. – Universidad Nacional de La Plata  
Calle 50 y 115 – 1er. Piso – Edificio Bosque Oeste  
lmarrone@linti.unlp.edu.ar

## Resumen

*Este trabajo prende dar una visión de las redes definidas por software (SDN), los fundamentos que llevaron a su desarrollo y evaluar los controladores que se utilizan para el despliegue de las mismas.*

**Palabras clave:** SDN, controladores, Interfaces

## Contexto

El presente trabajo está enmarcado en el PI 29/A451-1 “Análisis de performance del protocolo TCP utilizado en redes móviles” radicado de la UNPA-UARG y actualmente se encuentra en su etapa inicial de desarrollo. El proyecto está compuesto por cuatro investigadores de la UNPA, uno de la UNLP y un alumno becado con beca de iniciación en la investigación para estudiantes de la UNPA-UARG, dirigido por el Sr. Carlos A. Talay y codirigido por el Sr. Luis A. Marrone perteneciente a la UNLP. Este proyecto se financia íntegramente con fondos destinados a proyecto de investigación de la UNPA-UARG.

## 1. Introducción

En la actualidad la complejidad de las redes es cada vez mayor. Además de su creciente tamaño, requieren el despliegue dinámico de servicios. Sin embargo, las redes tradicionales

no poseen la flexibilidad para probar nuevas ideas sin interrumpir los servicios que se estén ejecutando.

Paralelamente, la configuración y mantenimiento de las redes siguen siendo complejas: un administrador de red accede mediante la interfaz de línea de comandos a los dispositivos [1], lo que presenta dificultades en el caso de trabajar con dispositivos de distintos fabricantes.

Surgen también nuevas exigencias por parte de los usuarios, como una gran movilidad, seguridad, o aplicaciones ‘Big Data’.

Es así que la capacidad de las redes de comunicación de datos está siendo rebasada, llegando a ser insuficientes para poder atender los grandes volúmenes de información que demandan estos nuevos servicios. No solo se trata del volumen, sino también de la gran variedad de información disponible en dispositivos móviles, GPS, contenidos multimedia de alta definición, sensores eléctricos, medidores, etc., que además deben comunicarse a una gran velocidad.

Las redes convencionales consisten de dispositivos cerrados, donde cada dispositivo tiene que ser configurado de manera individual, utilizando comandos específicos que tiene cada proveedor de equipos de red. Esto se ha convertido en un obstáculo, al desplegar nuevas versiones de protocolos existentes, actualizaciones o protocolos nuevos en las redes ya instaladas y operando.

El concepto SDN surgió como la arquitectura de red, donde el plano de control se desacopla del plano de datos. En lugar de hacer cumplir políticas y protocolos que se ejecutan en dispositivos dispersos, la red se limita a realizar el simple reenvío de información y el controlador de la red toma las decisiones [2], agilizando los flujos de información en la red. La idea fundamental de las Redes Definidas por Software (SDN) es clara: trasladar el plano de control de los dispositivos físicos (enrutadores y conmutadores) unificándolo en un solo elemento externo a la red física, el controlador [3].

La arquitectura de SDN se puede observar en la siguiente figura (Figura 1) [4], se puede ver como se intercambia información entre las diferentes capas de la arquitectura: aplicación, control e infraestructura.

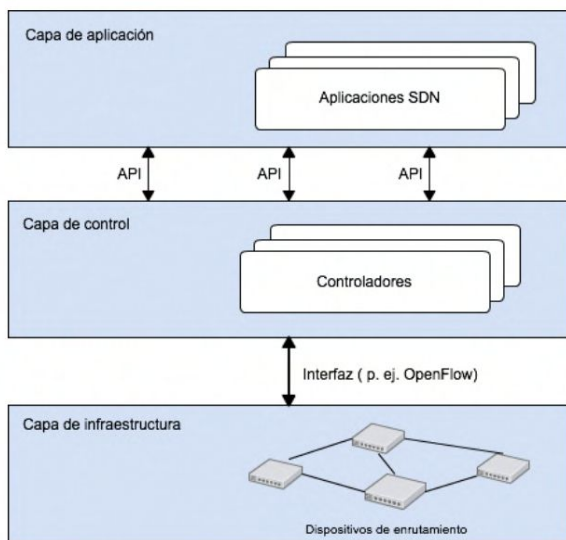


Fig.1: Arquitectura SDN

En las redes tradicionales, el plano de reenvío, control, administración y servicios está integrado en cada dispositivo de red. Por el contrario, el modelo SDN implementa la mayoría de las decisiones de administración, servicios y selección de ruta como funciones avanzadas dentro del controlador o en aplicaciones [5].

De esta manera, en una SDN se observa [6]:

- El plano de control y de datos están desacoplados. La funcionalidad de control se elimina de los dispositivos físicos de red que se convertirán en elementos de reenvío de paquetes.
- Las decisiones del reenvío de paquetes se toman en función a flujos, en lugar de basarse en las direcciones destino de los paquetes.
- La lógica de control se mueve a una entidad externa, llamado el controlador de la SDN o sistema operativo de red.
- La red es programable a través de aplicaciones en ejecución en la parte superior del controlador que interactúa con los dispositivos en el plano de datos subyacente.
- La arquitectura de una SDN contempla tres capas funcionales, el plano de datos, el plano de control y el plano de aplicación.

### Interfaces

Toda la lógica de control de la red se implementa como una aplicación en la parte superior del sistema operativo de red que interactúa con las API de red que ofrece el controlador. [7].

El controlador cuenta con dos interfaces principales, una que interactúa con el plano de gestión llamada Northbound (NBI), y otra que interactúa con el plano de datos, llamada Southbound (SBI) (Figura 1). Así mismo, y con la finalidad de establecer comunicación con otros controladores en la red, se cuenta con las interfaces Eastbound (EBI) y Westbound (WBI) [8].

El objetivo de estas interfaces es proporcionar una apropiada abstracción de estados y eventos relacionados con la red subyacente y un canal de comunicaciones con características apropiadas, para apoyar la necesaria funcionalidad requerida entre los clientes en cualquier extremo del protocolo.

*Interfaz Southbound:* Formaliza la manera en que los elementos de control y plano de datos interactúan [6]. Esto permite la comunicación entre el controlador SDN y los dispositivos de Red en la capa de Infraestructura. Básicamente, su función es permitir al Controlador comunicarse con los elementos de conmutación de la red y programar la lógica de las comunicaciones en el hardware. Hay varias implementaciones de estas APIs, entre ellas OpenFlow que es el elemento principal en esta interfaz para implementar las soluciones SDN y VxLAN (más utilizadas en virtualización de redes, para definir capas intermedias u overlays).

OpenFlow es el protocolo empleado para comunicación entre el controlador (plano de control) y el switch (plano de datos). Se define como un protocolo emergente y abierto de comunicaciones y es la primera interfaz de comunicaciones estándar definida entre los planos de control y datos de una arquitectura de SDN.

*Interfaz Northbound:* Permite el intercambio de datos entre el controlador y la aplicación. El tipo de información que se intercambia, su forma y su frecuencia depende de cada aplicación. No hay estandarización para esta interfaz [9].

*Interfaces Eastbound y Westbound:* Permiten la interconexión de redes convencionales con las SDN, además sirven como conducto de información entre varios planos de control de diferentes dominios de SDN. [9].

### **Plano de Aplicación**

Las aplicaciones SDN comunican sus requisitos a la red a través de una API que conecta con la capa de control. Están diseñadas para satisfacer las necesidades de los usuarios. De esta manera, a la capa de aplicación la componen las aplicaciones de usuarios. Presentan una interfaz común para el desarrollo de aplicaciones y permiten la

interacción entre las aplicaciones externas y el controlador SDN. Actualmente los controladores de red ofrecen una variedad de APIs Northbound, como APIs RESTful, sistemas de archivos y lenguajes de programación. Incluye aplicaciones tales como enrutamiento, cortafuegos, balanceadores de carga, seguimientos y entre otros. En esencia, una aplicación de gestión define las políticas, que se traducen en última instancia en instrucciones específicas para la SouthBand Interface que programan el comportamiento de los dispositivos de reenvío [6].

### **Plano de Control**

La capa intermedia la forma el Controlador SDN, quien tiene una visión global de la red. Al igual que en un sistema operativo tradicional, la plataforma de control abstrae los detalles de bajo nivel de la conexión y la interacción con dispositivos de reenvío, es decir materializa las políticas de red [6]

El controlador es el componente más importante de la arquitectura SDN ya que gestiona la capa de aplicación e infraestructura mediante sus interfaces, una con cada plano adjunto. El controlador es un elemento crítico en una arquitectura SDN, ya brinda el apoyo a la lógica de control para generar la configuración de la red con base en las políticas definidas.

Mediante la comunicación con el plano de infraestructura, se recoge el estado de la red y, según las exigencias de las aplicaciones, actualiza, en los dispositivos, las reglas de reenvío.

Por otra parte, se comunica con las aplicaciones SDN con un lenguaje de alto nivel, teniendo opciones como lenguajes existentes (Python, Java, C++), librerías en un kit software de desarrollador (SDK) como OnePK de Cisco, o lenguajes nuevos como Flowbased Management Language (FML), Frenetic o Nettle. Además, del estado de la

red que se recoge de la capa de infraestructura.

Aunque el modelo del plano de control es centralizado, la implementación puede ser distribuida para la escalabilidad y redundancia. Se necesita que un controlador sea capaz de comunicarse con otros debido a que un solo controlador puede ocasionar problemas de congestión ya que se trata de un punto crítico. Se puede tener controladores de 'backup'. La solución más citada en este sentido es HyperFlow, que proporciona una vista sincronizada y consistente entre múltiples controladores.

### **Plano de Datos**

La capa inferior es la capa de infraestructura y hace referencia a los dispositivos de red como los hosts, switches/routers físicos y/o virtuales, y los medios de transmisión.

El plano de datos proporciona el servicio fundamental de reenvío de datos.

Se utiliza el término conmutador SDN (SDN switch) para referirse al elemento de reenvío en el plano de datos. El término conmutador SDN no debe considerarse como una limitación de las funciones del elemento de red [8], si el dispositivo es un enrutador (Router), conmutador (Switch), o cortafuegos (Firewall). Estos dispositivos recogen el estado de la red (topología o estadísticas de tráfico) y se lo comunica al controlador, el cual a su vez le indica las reglas de reenvío de paquetes.

Es aquí donde cambia totalmente la filosofía respecto a las redes tradicionales.

Las aplicaciones definen el uso que se va a dar a la red, lo comunican a través de las Northbound APIs al controlador SDN, el cuál toma las decisiones oportunas y las comunica a la infraestructura de red mediante las SouthBound APIs. Estas APIs pueden ser abiertas o propietarias.

## **2. Líneas de Investigación, Desarrollo e Innovación**

En el centro de la arquitectura de las SDN se encuentra la capa Control con el controlador SDN, que es quien gestiona los flujos. El controlador equivale al sistema operativo de la red que controla todas las comunicaciones entre las aplicaciones y los dispositivos. El controlador SDN se encarga de traducir las necesidades o requisitos de la capa Aplicación a los elementos de red, y de proporcionar información relevante a las aplicaciones SDN, pudiendo incluir estadísticas y eventos. Es por ello que se intentará realizar un estudio y análisis de la capa de control.

## **3. Resultados y Objetivos**

El objetivo de esta primera parte es el estudio y análisis de los diferentes controladores de red disponibles en el mercado y poder realizar comparaciones cualitativas y cuantitativas de sus funciones y alcances.

## **4. Formación de Recursos Humanos**

La formación de recursos humanos es una preocupación permanente. En este proyecto se tienen dos investigadores en plena formación y un estudiante avanzado que se encuentra usufructuando una beca de investigación.

## **5. Referencias**

- [1] Juniper Networks, "What's behind network downtime?" (2008). <https://www-935.ibm.com/services/au/gts/pdf/200249.pdf> (Consulta: 6 de agosto de 2015).
- [2] Egilmez, H. E., & Dane, S. T. "OpenQoS: An OpenFlow controller design for

- multimedia delivery with end-to-end Quality of Service over Software-Defined Networks”. Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), IEEE, 1–8. (2012). Retrieved from [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=6411795](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6411795)
- [3] Hakiri, A., Gokhale, A., Berthou, P., Schmidt, D. C., & Gayraud, T. Software-Defined Networking: Challenges and research opportunities for Future Internet. *Computer Networks*, 75, 453–471. (2014). <http://doi.org/10.1016/j.comnet.2014.10.015>
- [4] Xia, W.; Weng, Y.; Foh, C. H.; Niyato, D.; Xie, H. “A Survey on Software-Defined Networking”. (2015). <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6834762>
- [5] Centeno, Alejandro & Rodriguez Vergel, Carlos Manuel & Anías Calderón, Caridad & Camilo, Frank & Bondarenko, Casmartíño & Uci,. (2014). “Controladores SDN, elementos para su selección y evaluación”. *Telemática*. 13. 10-20.
- [6] Kreutz, D., Ramos, F. M. V, Verissimo, P., Rothenberg, C. E., Azodolmolky, S., Uhlig, S., Ramos, F. Software-Defined Networking: A Comprehensive Survey. *IEEE*, 103(1), 14–76. (2014). <https://ieeexplore.ieee.org/document/6994333>
- [7] Tijare, Poonam & Vasudevan, Deepika. (2016). The Northbound APIs of Software Defined Networks. [10.5281/zenodo.160891](https://zenodo.org/record/160891).
- [8] DeCusatis, C. (Ed.). Handbook of fiber optic data communication: a practical guide to optical networking (pp. 427–445). Academic Press. (2013).
- [9] ONF Open Networking Foundation. ONLINE: <https://www.opennetworking.org/> [Dic 2019].