

## Controladores SDN, elementos para su selección y evaluación.

**Alejandro García Centeno<sup>1</sup>, Carlos Manuel Rodríguez Vergel<sup>2</sup>, Caridad Anías Calderón<sup>3</sup>, Frank Camilo Casmartíño Bondarenko<sup>4</sup>**

<sup>1</sup>UCI, [alejandrogc@uci.cu](mailto:alejandrogc@uci.cu)

<sup>2</sup>CNIC, [carlos.vergel@cnic.edu.cu](mailto:carlos.vergel@cnic.edu.cu)

<sup>3</sup>ISPJAE, Dra.CT, [cache@tesla.cujae.edu.cu](mailto:cache@tesla.cujae.edu.cu)

<sup>4</sup>ISPJAE, [fcasmartino@electrica.cujae.edu.cu](mailto:fcasmartino@electrica.cujae.edu.cu)

### RESUMEN

Las Redes Definidas por Software o SDN (Software Defined Network por sus siglas en inglés) constituyen una tecnología en la que se desacopla el plano de datos del plano de control y en la que se emplean controladores, que son los responsables de gestionar la información de reenvío de los conmutadores o *switch* en redes cableadas. Hoy en día se han desarrollado muchos controladores SDN, tanto de código abierto como comerciales, por lo que uno de los principales aspectos a tener en cuenta en el entorno actual de las Redes Definidas por Software, es cuál controlador elegir para una solución de este tipo. En este artículo se revisan las características más destacadas de los controladores SDN y se precisan algunos elementos a tener en cuenta para su selección y evaluación.

Palabras claves: Controlador, SDN, red de telecomunicación

### ABSTRACT

*The Software Defined Radio or SDN is a technology in which the data plane and the control plane are decoupled and in which are used controllers, which ones are responsible to manage the switch forwarding information in wired networks. Today have been developed a lot of SDN controllers, as much of open code as commercials, which is why one of the main aspects to have in account in the present-day surroundings the Software Defined Network, is which controller select for this kind of solution. In this article the most prominent features of the SDN controllers and some elements to have in account for their selection and evaluation are described in detail.*

*Key words: Controller, SDN, telecommunication network*

## INTRODUCCIÓN

La explosión de los dispositivos móviles, la virtualización de servidores, la llegada de los servicios de la nube y, al mismo tiempo, la modificación del patrón de tráfico por parte de los usuarios hacia los centros de datos, son algunas de las innovaciones que han impulsado a la industria de las TIC (Tecnologías de la Información y las Comunicaciones) a reexaminar las tecnologías de red tradicionales.

Por su parte, las arquitecturas tradicionales de redes no son óptimas para satisfacer todos los requerimientos de las empresas y los usuarios finales. Los diseños y equipos utilizados tradicionalmente en las redes de telecomunicaciones, aunque funcionan correctamente, no están alineados con los objetivos de negocio ni con la lógica de las aplicaciones, sino que forman una estructura cerrada y estática que no se puede adaptar en tiempo real a la demanda de las aplicaciones <sup>1</sup>. Entre las limitaciones de las arquitecturas actuales se encuentran: complejidad para agregar o eliminar dispositivos y para configurar calidad de servicio, lo que hace a las redes actuales relativamente estáticas; dificultad para aplicar políticas; incapacidad para escalar en gran magnitud y dependencia de los proveedores para desplegar servicios y capacidades rápidamente en respuesta a las necesidades cambiantes del negocio, entre otros <sup>2</sup>.

En ese contexto, las SDN se presentan como un nuevo paradigma para cubrir las nuevas necesidades y promete transformar las arquitecturas y la gestión de las redes que se conocen en la actualidad. Las SDN han surgido como un enfoque para fomentar la innovación en la red a través de una mayor flexibilidad, capacidad de programación, gestión y rentabilidad <sup>2</sup>.

En las SDN la inteligencia de la red se encuentra lógicamente centralizada en controladores, basados en software, que mantienen una visión global de la red. Como resultado, la red aparece frente a las aplicaciones y a las decisiones de política como un conmutador lógico y único. De esta forma, las empresas que utilizan la tecnología SDN logran la independencia de los proveedores y adquieren el control sobre toda la red desde un único punto lógico, lo que simplifica en gran medida el diseño y operación de sus redes. Las SDN también permiten simplificar los dispositivos de red, pues ya estos no tienen que entender y procesar varios protocolos estándares, sino simplemente aceptar instrucciones de los controladores SDN <sup>2,3</sup>.

Quizás lo más importante de las SDN es que los operadores de red pueden configurar, mediante programación y de forma centralizada, la red y sus servicios, modificando su comportamiento en tiempo real, pudiendo desplegar nuevas aplicaciones y servicios en cuestión de horas o días, en lugar de las semanas o meses necesarios en la tecnología de redes anteriores. Al estar la red centralizada en la capa de control, las SDN ofrecen a los operadores de red flexibilidad para configurar, administrar, proteger y optimizar los recursos de la red a través de programas dinámicos y automatizados. De esta forma, los administradores no tienen que esperar a que los fabricantes lancen sus programas, lo cual es muestra de la relativa independencia de esta tecnología, con el desarrollo del software por parte de las empresas proveedoras <sup>2,3</sup>.

De lo dicho anteriormente, se hace evidente que el control programable es el aspecto más importante de las SDN. En este trabajo se examinan las características de los controladores, se precisan los aspectos a tener en cuenta para la selección de los mismos y se revisan herramientas para su evaluación.

## LOS CONTROLADORES EN LA ARQUITECTURA SDN

La figura 1 muestra una visión general de la arquitectura básica de las SDN, sus componentes arquitectónicos e interacciones. Como se observa, la arquitectura está compuesta por tres capas: Infraestructura, Control y Aplicaciones. La inteligencia de la red, en SDN, se encuentra (lógicamente) centralizada en controladores basados en software, que mantienen una visión global de la misma.

Los conmutadores o *switch* delegan su inteligencia al controlador y pasan a ser simples unidades de conmutación de tráfico. Es en el sistema operativo del controlador donde se configura el enrutamiento y estará abierto a la implementación de las nuevas funcionalidades a través de las Interfaces de Programación de Aplicaciones (Application Programming interface: API: por sus siglas en inglés) hacia las aplicaciones, también denominadas APIs “hacia el norte”.

La capa Infraestructura está compuesta por los elementos o dispositivos de red, los que exponen sus capacidades a través de una interfaz de control del plano de datos, también denominada interfaz “hacia el sur”, que permite la comunicación entre el controlador SDN y los dispositivos de red. La interfaz permite al menos:

- (i) el control de programación de todas las operaciones de reenvío por parte del controlador
- (ii) el anuncio y aviso de capacidades, informes de estadísticas y la notificación de eventos de los dispositivos de red al controlador

La interfaz de control del plano de datos se formaliza a través del protocolo *OpenFlow*, que se ha convertido en el protocolo oficial a utilizar para la conexión remota entre el controlador SDN y los *switches*. Como resultado, la creación de una red SDN, debe basarse en la selección de dispositivos y de software de control que soporten el estándar *OpenFlow*.

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 <sup>4</sup>.

La capa Aplicaciones permite comunicar al controlador SDN, mediante las APIs hacia el norte, sus necesidades y el comportamiento que desean de la red. La interfaz de los controladores SDN hacia las aplicaciones es un conjunto de interfaces ya que la definición de aplicaciones SDN es muy amplia, cubriendo desde servicios de red, como QoS, a aplicaciones de negocio. Las diversas aplicaciones SDN tienen que hablar con la red de maneras diferentes y requerir interfaces diferentes.

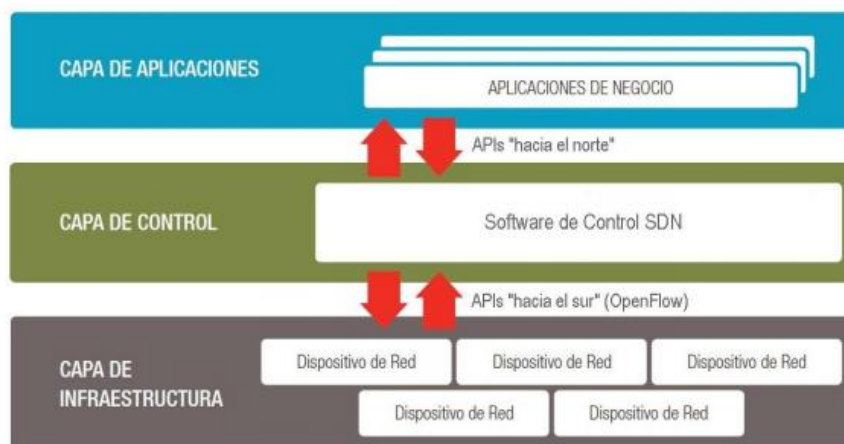


Figura 1: Arquitectura SDN <sup>4</sup>

### CONTROLADORES OPENFLOW

Un controlador *OpenFlow* ofrece una interfaz de programación para los conmutadores *OpenFlow* de tal forma que, las aplicaciones de gestión, a través de la misma, pueden realizar tareas de gestión y ofrecer nuevas funcionalidades. Un controlador SDN puede ser descrito de forma general como un sistema de software, o colección de sistemas, que ofrecen <sup>3</sup>:

- Gestión del estado de la red, que implica una base de datos. Estas bases de datos sirven como un repositorio para la información de los elementos de red gestionados, incluyendo el estado de la red, alguna información de configuración temporal e información sobre la topología de la red.
- Un modelo de datos de alto nivel que captura las relaciones entre los recursos gestionados, las políticas y otros servicios prestados por el controlador. En muchos casos estos modelos de datos se construyen utilizando el lenguaje de modelado Yang [5].
- Un mecanismo de descubrimiento de dispositivos, topología y servicio; un sistema de cálculo de ruta y, potencialmente, otros servicios de información centrados en la red o en los recursos.
- Una sesión de control segura sobre el Protocolo de Control de Trasmisión (Transmission Control Protocol: TCP por sus siglas en inglés) entre el controlador y los agentes asociados en los elementos de la red, por ejemplo con el uso del protocolo TLS (Seguridad en la Capa de Transporte o Transport Layer Security, por sus siglas en inglés).
- Un protocolo basado en estándares (OpenFlow) para obtener el estado de la red impulsado por las aplicaciones de los elementos de red.
- Un conjunto de APIs, a menudo RESTful (Transferencia de estado representacional o Representational State Transfer, por sus siglas en inglés) que exponen los servicios del controlador a las aplicaciones de gestión. Esto facilita la mayor parte de la interacción del controlador con estas aplicaciones. Esta interfaz se representa a partir del modelo de datos que describe los servicios y funciones del controlador. En algunos casos, el controlador y su API son parte de un entorno de desarrollo que genera el código de la API a partir del modelo de datos [6].

- Algunos controladores ofrecen entornos de desarrollo robustos que permiten la expansión de las capacidades básicas del núcleo y la posterior publicación de las APIs para los nuevos módulos, incluyendo los que soportan la expansión dinámica de las capacidades del controlador.

Por tanto, en las SDN, es el controlador central el que dicta el comportamiento general de la red a partir de los requerimientos de las aplicaciones [7]. Ejemplos de algunos controladores de código abierto existentes son: Beacon [8], Floodlight [9], NOX [10], POX [11], Ryu [12], Trema [13] y OpenDayLight [14] (ODL). En la tabla 1 se muestran algunas características de estos controladores.

	Beacon	Floodlight	NOX	POX	Trema	Ryu	ODL
Soporte OpenFlow	OF v1.0	OF v1.0	OF v1.0	OF v1.0	OF v1.3	OF v1.0, v1.2, v1.3 y extensiones Nicira	OF v1.0
Virtualización	Mininet y Open vSwitch	Mininet y Open vSwitch	Mininet y Open vSwitch	Mininet y Open vSwitch	Construcción de una herramienta virtual de simulación	Mininet y Open vSwitch	Mininet y Open vSwitch
Lenguaje de desarrollo	Java	Java	C++	Python	Rudy/C	Python	Java
Provee REST API	No	Si	No	No	Si (Básica)	Si (Básica)	Si
Interfaz Gráfica	Web	Web	Python+, QT4	Python+, QT4, Web	No	Web	Web
Soporte de plataformas	Linux, Mac OS, Windows y Android para móviles	Linux, Mac OS, Windows	Linux	Linux, Mac OS, Windows	Linux	Linux	Linux, Mac OS, Windows
Soporte de OpenStack	No	Si	No	No	Si	Si	Si
Multiprocesos	Si	Si	Si	No	Si	No	Si
Código Abierto	Si	Si	Si	Si	Si	Si	Si
Tiempo en el mercado	4 años	2 años	6 años	1 años	2 años	1 años	5 meses
Documentación	Buena	Buena	Media	Pobre	Media	Media	Media

**Tabla 1: Algunas características en los controladores**

## ELEMENTOS PARA LA SELECCIÓN DE CONTROLADORES SDN

La selección del controlador es vital en el diseño de una red SDN. A continuación se presentan las características que los autores de este trabajo, después de revisar la bibliografía [15], consideran que se deben tener en cuenta al evaluar un controlador SDN. Estas son:

1. **Soporte *OpenFlow*:** Al elegir un controlador los administradores de red necesitan conocer las características de las versiones de OpenFlow que el controlador soporta, así como las posibilidades que ofrece el proveedor para migrar a las nuevas versiones del protocolo, tales como la v1.3 y la v1.4. Una razón por la que esto es necesario, es que algunas funciones importantes como, por ejemplo, el soporte de IPv6 no es parte de OpenFlow v1.0 pues se incluyen a partir del estándar OpenFlow v1.2.
2. **Virtualización de red:** Debido a los beneficios que ofrece la virtualización de red, un controlador SDN debe soportarla. Esta característica permite a los administradores crear dinámicamente las redes virtuales basadas en políticas, disociadas de las redes físicas, para satisfacer una amplia gama de requisitos como, por ejemplo, la ampliación horizontal de la capacidad, sin afectar los flujos existentes. Otra de las muchas ventajas de la virtualización de red es que permite un completo aislamiento entre cada segmento de red lo que es muy útil por razones de seguridad como, por ejemplo, mantener aislados los datos generados por un grupo de usuarios de otros usuarios y permitir a los desarrolladores de aplicaciones ejecutar las mismas en un entorno de trabajo sin afectar el tráfico. Para cumplir estos requisitos de manera eficiente, en los controladores SDN se deben configurar las redes virtuales de forma centralizada, con total aislamiento unas de otras, y dichas configuraciones deben estar automatizadas.
3. **Funcionalidad de la red:** Para lograr mayor flexibilidad en términos de cómo los flujos son enrutados, es importante que el controlador SDN pueda tomar decisiones de enrutamiento basado en múltiples campos de la cabecera de OpenFlow. También es importante que el controlador pueda definir los parámetros de QoS flujo por flujo. Otra importante funcionalidad en un controlador SDN es su capacidad para descubrir múltiples caminos desde el origen del flujo a su destino y para dividir el tráfico de un flujo dado a través de múltiples enlaces. Esta capacidad elimina la necesidad de STP (Protocolo de Árbol Extendido o Spanning Tree Protocol por sus siglas en inglés) y aumenta el rendimiento y la escalabilidad de la red permitiendo, también, eliminar la necesidad de añadir a la complejidad de la red nuevos protocolos como TRILL (Transparent Interconnection of Lots of Links) o SPB (Shortest Path Bridging).
4. **Escalabilidad:** Una consideración fundamental con respecto a la escalabilidad de una red SDN es el número de conmutadores o ***switches*** que un controlador SDN puede soportar. En la actualidad se debe esperar que los controladores soporten un mínimo de 100 ***switches***, pero en última instancia esto depende de las aplicaciones que soportan. Otro factor que limita la escalabilidad de una red SDN es la proliferación de entradas en la tabla de flujo, ya que sin algún tipo de optimización, se requiere de una entrada salto por alto para cada flujo. Al evaluar los controladores SDN, es necesario asegurarse que el controlador puede disminuir el impacto de sobrecarga de difusión de red, la cual limita la escalabilidad de la arquitectura de red implementada y reducir al mínimo la proliferación de las entradas de la tabla de flujo. Otro aspecto de la escalabilidad es la capacidad del controlador de SDN para crear una SDN que pueda abarcar múltiples sitios. Esta capacidad permite el movimiento de máquinas virtuales y el almacenamiento virtual entre sitios. Para maximizar el beneficio de esta capacidad, el

controlador SDN debe permitir que las políticas de red para el enrutamiento y reenvío se apliquen automáticamente para la migración de servidores y / o almacenamiento.

5. **Rendimiento:** Una de las principales funciones de un controlador SDN es establecer flujos. Por ello, dos de los indicadores claves de rendimiento asociados con un controlador SDN son el tiempo de conformación de flujo y el número de flujos por segundo que puede establecer el controlador. Estas métricas de desempeño influyen en gran medida cuando se requiere añadir controladores como, por ejemplo, cuando los **switches** inician más flujos de los que pueden ser soportados por el controlador o los controladores SDN existentes.

Los flujos se pueden configurar de dos maneras: de manera proactiva o reactiva. La configuración proactiva de los flujos se realiza antes de que el paquete llegue al switch OpenFlow. Así, cuando al switch llega el primer paquete, este conoce qué hacer con el mismo, dando lugar a insignificantes retrasos, no existiendo límite real en el número de flujos por segundo que el controlador puede soportar. Idealmente, el controlador SDN pre-llena las tablas de flujo lo más posible.

La configuración reactiva de flujo se produce cuando el switch OpenFlow recibe un paquete que no coincide con las entradas de su tabla de flujo y, por tanto, tiene que enviarlo al controlador quien decide qué hacer con el mismo y almacena la nueva información en el switch en cuestión. El tiempo asociado con la configuración reactiva de flujo es la suma del tiempo que se necesita para enviar el paquete desde el switch OpenFlow al controlador SDN, el tiempo de procesamiento en el controlador y el tiempo que este tarda en llenar la tabla de flujo en el switch. Los factores claves que afectan al tiempo de establecimiento del flujo incluyen la capacidad de procesamiento de los switches que están conectados al controlador y el procesamiento y rendimiento de entrada / salida del controlador.

6. **Programación de red:** Una de las características fundamentales de las SDN es la existencia de interfaces para la programación del controlador, lo que posibilita que este ofrezca varias funcionalidades. Ejemplos de programación que se deben buscar en un controlador SDN son la capacidad de redirigir el tráfico (por razones de seguridad se puede disponer que el tráfico entrante a un servidor pase a través de un corta fuegos o firewall, pero para no consumir los recursos del servidor de seguridad con tráfico limpio, se puede decidir que no pase por el firewall el tráfico saliente del mismo servidor) y la posibilidad de aplicar filtros sofisticados a los paquetes (que pueden ser pensados como ACLs (Listas de Control de Acceso o Access Control List por sus siglas en inglés) dinámicas e inteligentes como combinaciones complejas de múltiples campos de cabecera de paquetes).

Un controlador SDN también puede soportar la programación, proporcionando plantillas que permitan la creación de secuencias de comandos CLI (Interfaz de Línea de Comando o Command Line Interface, por sus siglas en inglés) con las que es posible la programación dinámica de la red.

7. **Confiabilidad:** Una de las técnicas que un controlador SDN puede utilizar para aumentar la fiabilidad de la red, es la capacidad de descubrir múltiples caminos desde el origen hasta el destino lo cual puede realizar si continuamente controla la topología de la red. Si el controlador SDN establece varias rutas entre el origen y el destino, la disponibilidad de la solución no se ve afectada por la interrupción de un solo enlace. Alternativamente, si el controlador SDN sólo

establece una única ruta del origen al destino, cuando ocurra un fallo en un enlace, el controlador debe ser capaz de redirigir el tráfico rápidamente a un enlace activo.

Relativo a la disponibilidad de las conexiones externas, es importante que el controlador soporte tecnologías alternativas de diseño, como el VRRP (Virtual Redundancy Protocol, por sus siglas en inglés) y MC LAG (Multi-Chassis Link Aggregation Group, por sus siglas en inglés), que tienen como objetivo aumentar la fiabilidad de la red.

En cuanto a la disponibilidad del controlador en sí mismo, es fundamental que el mismo se construya utilizando redundancia tanto para las características de hardware como para las de software. También es imprescindible que el controlador SDN permita agrupaciones (clusters).

8. **Seguridad de la red:** Con el fin de proporcionar seguridad a la red, un controlador SDN debe ser capaz de soportar la autenticación y autorización. Debido a que un controlador SDN es candidato para un ataque malicioso, necesita poseer la capacidad de limitar las comunicaciones de control, y ser capaz de cuando la red está experimentando una sospecha de ataque.

9. **Monitorización centralizada y visualización:** Un controlador SDN tiene que ser capaz de utilizar los datos ofrecidos por el protocolo OpenFlow para identificar los problemas en la red y, automáticamente, cambiar la ruta que toma un flujo determinado. El controlador también debe poder definir qué tipo de tráfico controlar. Por ejemplo, se puede optar por no controlar el tráfico de replicación. Ser capaz de visualizar una red tradicional siempre ha sido a la vez importante y difícil lo que se incrementa en un entorno en el que múltiples redes virtuales se ejecutan en la parte superior de una red física donde es necesario que el controlador SDN sea capaz de descubrir y presentar la visualización de los enlaces físicos de la red y de las múltiples redes virtuales que se ejecutan en la red física. El controlador también debe permitir ver los flujos, tanto de la perspectiva de la red física como de la virtual, y obtener información detallada sobre los mismos.

También, debería ser posible monitorizar el controlador SDN utilizando protocolos y técnicas estándares de gestión tales como SNMP (Protocolo Simple de Gestión de Redes o Simple Network Management, por sus siglas en inglés). Adicionalmente el controlador SDN debe ofrecer soporte para una amplia gama de MIBs (Bases de Información de Gestión o Management Information Bases, por sus siglas en inglés) estándares y MIB privadas para poder controlar los elementos de la red virtual. Lo ideal sería que el controlador SDN ofrezca acceso a la información de red, por ejemplo a los dispositivos conectados, al estado del puerto, y al estado de enlace a través de una API REST.

10. **Fabricantes de controladores SDN:** Dado el creciente interés en las SDN, numerosos fabricantes han entrado en el mercado y muchos más han anunciado la intención de hacerlo. Debido a la volatilidad del mercado SDN en general, y del mercado del controlador SDN, en particular, las organizaciones que deben evaluar controladores SDN para sus redes deben centrarse no sólo en los atributos técnicos del controlador, sino también en las características del vendedor. Entre estas se encuentra la competencia técnica y financiera del proveedor pues las organizaciones no pueden permitirse el lujo de tener sus desarrollos de red SDN perjudicadas por adquirir controladores de un proveedor que no puede mantenerse al día en el cambiante entorno SDN.

11. **Soporte de plataformas:** Los controladores SDN corren sobre sistemas operativos siendo importante que el que se esté evaluando sea multiplataforma pues ello permite mayor



flexibilidad e independencia al implantarlo. En muchas entidades es de gran interés que el controlador corra sobre plataformas de código abierto.

12. **Procesamiento:** Al evaluar un controlador se debe conocer si el mismo soporta procesos múltiples o no, pues esto puede repercutir en la escalabilidad de los núcleos de la CPU. No tendría sentido que un controlador monoproceso se ejecute sobre un hardware que posee múltiples CPUs pues no se estaría usando adecuadamente el mismo o que un controlador que soporte procesos múltiples se esté ejecutando sobre un hardware de un solo CPU. Por otra parte, los controladores de procesos múltiples deben ser los utilizados en redes empresariales mientras que los monoprocesos se pueden emplear en red pequeñas.

## HERRAMIENTAS PARA LA EVALUACIÓN DE LOS CONTROLADORES SDN

Los autores de este trabajo proponen dos herramientas para la evaluación de los controladores, Cbench y Hcprobe, las cuales son muy efectivas para realizar pruebas de rendimiento, escalabilidad, disponibilidad y seguridad.

### Cbench

Cbench (Controller Benchmark, por sus siglas en inglés) es una herramienta para monitorear controladores OpenFlow a través de la generación de eventos. Cbench emula un número configurable de switches OpenFlow los cuales se comunican con un controlador OpenFlow para medir diferentes aspectos del rendimiento y latencia del mismo. Su funcionamiento básico consiste en que cada switch emulado envía un número configurable de mensajes de nuevos flujos (mensajes OpenFlow packet-in) al controlador OpenFlow, espera por las respuestas apropiadas de configuración de flujos (mensajes OpenFlow packet-out o mensajes OpenFlow de modificación de flujos flow-mod) y registra estadísticas de la diferencia de tiempo entre las solicitudes y las respuestas, así como otras métricas de desempeño [16].

### Hcprobe

Escrita en Haskell, permite crear con facilidad escenarios para pruebas de control SDN. Es capaz de simular un gran número de *switch* y *host* conectados a un controlador. Empleando Hcprobe<sup>17</sup> se pueden analizar varios índices de operación del controlador de forma flexible. Con esta herramienta se pueden especificar patrones para generar mensajes OpenFlow (incluidos los malformados) y establecer perfiles de tráfico, entre otros. Sus características principales son:

- Generación de paquetes OpenFlow y de tablas de red;
- Implementación en un lenguaje de alto nivel, lo que hace que sea más fácil de extender;
- Existencia de una API para el diseño de pruebas personalizadas;
- Un lenguaje específico de dominio embebido (EDSL) para la creación de pruebas.

Además, Hcprobe proporciona un framework para la creación de varios casos de uso para estudiar el comportamiento de los controladores OpenFlow a través del procesamiento de diferentes tipos de mensajes. Con esta herramienta se puede generar todo tipo de mensajes OpenFlow switch-controlador y reproducir diferentes escenarios de comunicación entre ellos.

## CONCLUSIONES

Las SDN como tecnología novedosa introducen una nueva etapa en el desarrollo de las redes, permitiendo mejorar significativamente tanto la capacidad de gestión, como la escalabilidad y agilidad dentro de la red al emplear controladores.

Es importante conocer las características de los diferentes tipos de controladores y las funciones que los mismos ofrecen para seleccionar adecuadamente el que se va emplear en una implementación de red en particular. Se plantea que los elementos a tener en cuenta para la selección de controladores SDN son: soporte OpenFlow, virtualización de red, funcionalidad de la red, escalabilidad, rendimiento, programación de red, confiabilidad, seguridad de la red, monitorización centralizada y visualización, fabricantes de controladores SDN, soporte de plataformas y procesamiento.

Además, se debe poder evaluar el comportamiento de los controladores proponiéndose en este trabajo las herramientas **Cbench** y **Hcprobe**.

## REFERENCIAS

1. **GAREA AV:** “Cómo OpenFlow transformará las redes de datos tradicionales” Smarter Networking for Smarter Data Centers; September: IBM; 2012.
2. **OPEN NETWORKING FOUNDATION** “Software-Defined Networking: The New Norm for Networks”. 2013. Disponible en: <https://www.opennetworking.org>
3. **NADEAU TD, GRAY K.** “SDN: Software Defined Networks”. O’Reilly; 2013. Disponible en: <http://shop.oreilly.com/product/0636920027577.do>.
4. **SPERA C.** “Software Defined Network: el futuro de las arquitecturas de red”. DATA CENTER. 2013.
5. **M. Bjorklund.:** “YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)”. RFC 6020 (Proposed Standard), 2010. Disponible en: <http://www.ietf.org/rfc/rfc6020.txt>.
6. **GUIS I.** “The SDN Gold Rush To The Northbound API”. 2012 [updated 2014/05/14]. Disponible en: <http://www.sdncentral.com/technology/the-sdn-gold-rush-to-the-northbound-api/2012/11/>.(Referencia en Internet)
7. **DE TEJADA MUNTANER, GUILLERMO ROMERO:** “Evaluación de Controladores OpenFlow” (en inglés), Tesis de Maestría, pp. 23-24, Octubre 15, 2012.
8. **ERICKSON D.** “The Beacon OpenFlow Controller”. ACM 2013.
9. **“Floodlight OpenFlow Controller”.** [updated 2014/03]; Disponible en: <http://www.projectfloodlight.org/floodlight/>
10. **GUDE N, KOPONEN T, PETTIT J, PFAFF B, CASADO M, MCKEOWN N, ET AL.** “NOX: towards an operating system for networks”: ACM SIGCOMM Computer Communication Review, Volume 38, Number 3, July 2008
11. **POX.** <http://www.noxrepo.org/pox/about-pox/> (visitado en 9/2014)
12. **Ryu.** <http://osrg.github.com/ryu/> (visitado en 9/2014)
13. **Trema.** <http://trema.github.com/trema/> (visitado en 9/2014)
14. **OpenDayLight.** [www.opendaylight.org/](http://www.opendaylight.org/) (visitado en 9/2014)
15. **METZLER A.** “Ten Things to Look for in an SDN Controller”.2013.
16. **TOOTOONCHIAN A, GORBUNOV S, GANJALI Y, CASADO M, SHERWOOD R, EDITORS.** “On controller performance in software-defined networks”. USENIX Workshop on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services (Hot-ICE); 2012.
17. **GITHUB:** “ARCCN/hcprobe” (en inglés), 2013, disponible en: <https://github.com/ARCCN/hcprobe>. (visitado en 9/2014)