II CONVENCIÓN CIENTÍFICA INTERNACIONAL
"II CCI UCLV 2019"

XVIII SIMPOSIO DE INGENIERÍA ELÉCTRICA, SIE-2019

Controladores en Redes Definidas por Software (SDN)

Controllers in Software-Defined Networking (SDN)

Ing. Victor Manuel Peláez Fernández [1], Dr.C Félix Florentino Álvarez Paliza[2]

1- Victor Manuel Peláez Fernández. ETECSA, Cuba. Email: victor.pelaez@cubacel.cu
2- Félix Florentino Álvarez Paliza. UCLV, Cuba. Email: fapaliza@uclv.edu.cu.

Resumen: Actualmente, la tecnología de red está experimentando su tercera gran ola de revolución, el paso del hardware al software. Esta transición se lleva a cabo debido a la virtualización, mediante la cual el equipo de red física es reemplazado por un software que cumple la misma función. La tecnología SDN (Redes Definidas por Software, Software-Defined Networking) se introdujo con la virtualización. Asociado a esta definición, se establece una nueva arquitectura que desacopla el nivel de datos del nivel de control. Hasta ahora, las tablas de rutas se han calculado de forma distribuida por cada enrutador o conmutador. En la nueva arquitectura, los cálculos para un control óptimo los realiza un elemento diferente, llamado controlador. Este trabajo describe el funcionamiento de los controladores como elemento neurálgico de la arquitectura SDN. Se realizó inicialmente un análisis del concepto de SDN. Posteriormente se caracterizaron los controladores y fueron descritas sus funciones, realizando por último una evaluación del desempeño de varios controladores SDN.

- **Problemática:** Cómo seleccionar adecuadamente el controlador en una arquitectura SDN.

- **Objetivo(s):** Describir el funcionamiento de los controladores como elemento neurálgico de la arquitectura SDN.
- **Metodología:** Investigación.
- **Resultados y discusión:** El principal resultado de este trabajo es categorizar los controladores según sus usos en la industria y la academia, a fin de seleccionar el idóneo de acuerdo a los recursos con los que se cuenten y la aplicación que se desarrolle.
- **Conclusiones:** La selección del controlador depende de varios criterios y del desempeño en diferentes entornos. La elección del controlador para la academia puede diferir de la industria pero es vital en el diseño de una red SDN.

Abstract: Currently, network technology is experiencing its third great wave of revolution, the move from hardware to software. This transition takes place due to virtualization, whereby the physical network equipment is replaced by software that fulfills the same function. The SDN (Software-Defined Networking) technology was introduced with virtualization. Associated with this definition, a new architecture is established that decouples the level of data from the control level. So far, the route tables have been calculated in a distributed way by each router or switch. In the new architecture, the calculations for optimal control are made by a different element, called a controller. This work describes the operation of the controllers as a neuralgic element of the SDN architecture. An analysis of the SDN concept was initially carried out. Later the controllers were characterized and their functions were described, finally performing an evaluation of the performance of several SDN controllers.

Palabras Clave: Redes Definidas por Software; Arquitectura; Controlador; Características; Desempeño.

Keywords: Software-Defined Networking; Architecture; Controller; Feature; Performance.

1. Introducción

Actualmente, la tecnología de red está experimentando su tercera gran ola de revolución [1]. El primero fue el paso del modo de conmutación de circuitos (CS) al modo de conmutación de paquetes (PS), la segunda generación del modo cableado al inalámbrico. La tercera revolución, que examinamos en este trabajo, es el paso del hardware al software. Esta transición se lleva a cabo debido a la virtualización, mediante la cual el equipo de red física es reemplazado por un software que cumple la misma función.

Diversos elementos han creado una nueva generación de redes, podemos citar como el más influyente la nube [2]. La nube es un conjunto de recursos que, en lugar de estar en las instalaciones de una empresa o individuo en particular, están alojados en Internet. Los recursos se descentralizan y se reúnen en centros de recursos, conocidos como centros de datos (Datacenter) [3].

La tecnología SDN (Software-Defined Networking) se introdujo con la virtualización, lo que permite que los dispositivos de red se transformen en software. Asociado a esta definición, se ha definido una nueva arquitectura que desacopla el nivel de datos del nivel de control. Hasta ahora, las tablas de rutas se han calculado de forma distribuida por cada enrutador o conmutador. En la nueva arquitectura, los cálculos para un control óptimo los realiza un elemento diferente, llamado controlador. En general, el controlador está centralizado, pero se puede distribuir perfectamente.

El objetivo de SDN (redes definidas por software) es reducir costos mediante la virtualización, la automatización y la simplificación. Para este propósito, SDN facilita la personalización de las redes, un tiempo de configuración muy corto y una implementación de red con la calidad de servicio adecuada en lugar de una calidad de servicio general. En resumen, SDN se refiere a la capacidad de las aplicaciones de software para programar dispositivos de red individuales dinámicamente y, por lo tanto, controlar el comportamiento de la red como un todo.

El plano de control es una parte esencial de la arquitectura SDN, por lo que es muy importante prestar la atención adecuada a cualquier propuesta o diseño de un controlador

SDN. Durante los últimos años, se han desarrollado varios controladores y se han realizado varios estudios para evaluar, comparar y probar el rendimiento de estos controladores.

Como problema científico de este trabajo se ha partido de cómo lograr escoger adecuadamente un Controlador en la Arquitectura SDN.

A fin de responder a esa interrogante el objeto de investigación se enmarca en Redes Definidas por Software. Cuyo campo de acción son las funciones de los controladores en las Redes Definidas por Software.

El objetivo de este trabajo es describir el funcionamiento de los controladores como elemento neurálgico de la arquitectura SDN, estableciendo como objetivos específicos:

- Funciones de los Controladores en SDN
- Tipos de Controladores
- Comparación entre Controladores

2. Redes Definidas por Software (SDN, Software Defined Networking)

La Red Definida por Software (SDN, por sus siglas en inglés) se refiere a un nuevo enfoque de programación de la red, es decir, la capacidad de inicializar, controlar, cambiar y gestionar el comportamiento de la red de forma dinámica a través de interfaces abiertas. SDN enfatiza el papel del software en las redes en ejecución a través de la introducción de una abstracción para el reenvío de paquetes a través del plano de datos y, al hacerlo, lo separa del plano de control. Esta separación permite ciclos de innovación más rápidos en ambos planos [4].

Según la IETF en la RFC 7149 de 2014 [5], SDN es un conjunto de técnicas utilizadas para facilitar el diseño, la entrega y el funcionamiento de los servicios de red de una manera determinista, dinámica y escalable.

En la Figura 1 se aprecia el concepto de SDN según la recomendación de la UIT-T Y.3300 [6]

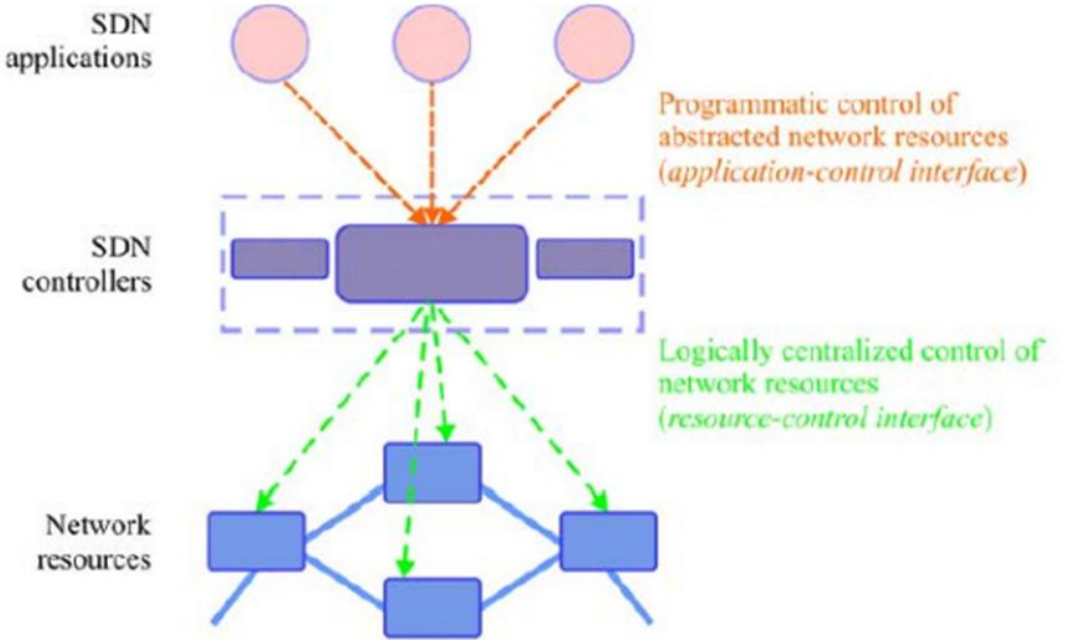

Figura 1. Concepto de SDN (Fuente: recomendación de la UIT-T Y.3300)

2.1 Arquitectura de SDN

SDN comenzó un proceso de estandarización a través de la Fundación de Redes Abiertas (ONF, Open Networking Foundation), que fue creada bajo los auspicios de las grandes empresas en California, a raíz de la propuesta de esta arquitectura por la Universidad de Stanford y Nicira.

La arquitectura propuesta por la ONF se muestra en la Figura 2 y comprende tres capas. La capa inferior es una capa de abstracción que desacopla el hardware del software y es responsable del transporte de datos. Este nivel describe los protocolos y algoritmos que permiten que los paquetes IP avancen a través de la red hasta su destino. Esto se llama el plano de infraestructura.

La segunda capa es el plano de control. Dicha capa contiene los controladores que proporcionan datos de control al plan de datos para que los datos se canalicen de la manera más efectiva posible. La visión de la ONF es centralizar el control para facilitar la recuperación de una gran cantidad de información sobre todos los clientes. El controlador centralizado permite obtener una especie de inteligencia. La infraestructura a ser

administrada se distribuye entre los controladores. Por supuesto, deben tenerse en cuenta los problemas de un entorno centralizado y, por lo tanto, duplicar los elementos de decisión.

La capa superior, el plano de aplicación, es responsable para las aplicaciones que necesitan los clientes y sus requisitos en términos de redes, almacenamiento, computación, seguridad y administración. Esta capa introduce la programabilidad de las aplicaciones y envía al controlador todos los elementos necesarios para abrir las redes de software adaptadas a las necesidades de las aplicaciones. Esta capa también incluye algunas aplicaciones muy especiales como el orquestador, el cual envía el controlador la información necesaria para que abra la interfaz hacia la aplicación correspondiente.

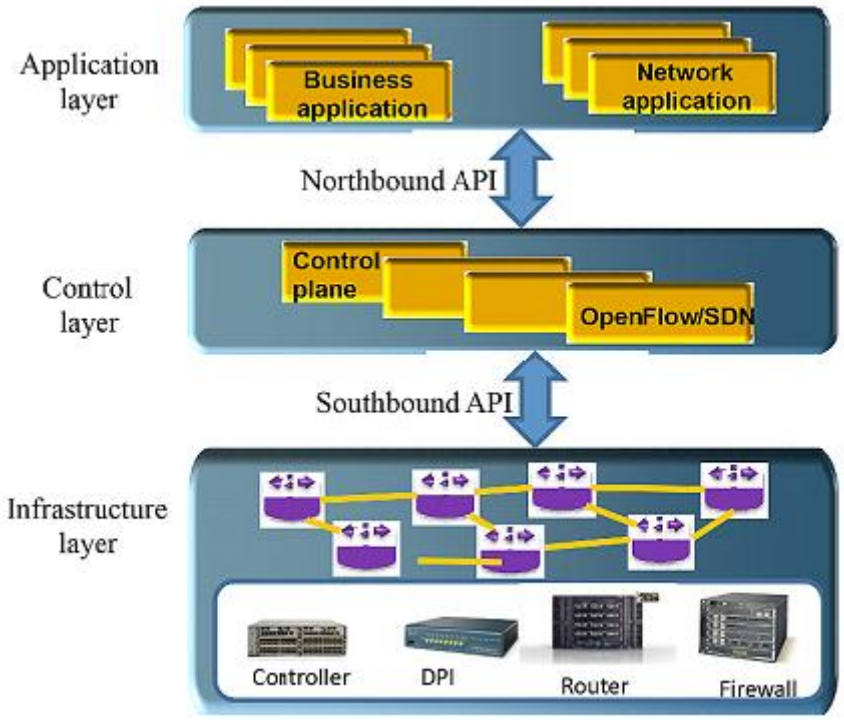

Figura 2. Arquitectura SDN (Fuente: [1])

2.2 Controladores SDN

Los controladores SDN deben brindar un grupo de servicios de red básicos que son considerados funcionalidades esenciales. Haciendo una analogía, estas funciones son como los servicios básicos de los sistemas operativos tales como: ejecución de programas, control de las operaciones de entrada y salida, comunicaciones, protección, etc. Estos

servicios son utilizados por otros servicios a nivel de sistema operativo y aplicaciones de usuario. De manera similar, las funciones tales como la topología, estadísticas, notificaciones, y la gestión de dispositivos, junto con mecanismos de reenvío y selección de rutas más cortas así como la seguridad, son funcionalidades de control esenciales que las aplicaciones de red pueden utilizar en la construcción de su lógica. Por ejemplo, el gestor de notificaciones debe ser capaz de recibir, procesar y reenviar eventos (por ejemplo, notificaciones de alarma, alarmas de seguridad, cambios de estado). Los mecanismos de seguridad son otro ejemplo, debido a que ellos son los componentes críticos a la hora de proporcionar un aislamiento y seguridad entre los servicios y las aplicaciones.

Los controladores están localizados en la capa de control SDN, la cual incluye soporte de aplicaciones, orquestación y funciones de abstracción [7].

Los controladores SDN se basan en protocolos, como OpenFlow el cual provee a su vez otros protocolos para una comunicación efectiva entre el plano de control y el plano de datos dentro de la arquitectura SDN.

OpenFlow es un protocolo que permite que un servidor indique a los conmutadores de red dónde enviar los paquetes. En una red convencional, cada conmutador (switch) tiene un software propietario que le dice qué hacer. Con OpenFlow, las decisiones de movimiento de paquetes están centralizadas, de modo que la red se puede programar independientemente de los conmutadores individuales y el engranaje del centro de datos. En un conmutador convencional, el reenvío de paquetes (la ruta de datos) y el enrutamiento de alto nivel (la ruta de control) se producen en el mismo dispositivo. Un conmutador OpenFlow separa la ruta de datos de la ruta de control. El plano de datos reside en el switch mismo; un controlador separado toma decisiones de enrutamiento de alto nivel. El switch y el controlador se comunican por medio del protocolo OpenFlow. Esta metodología, permite un uso más efectivo de los recursos de red que con las redes tradicionales. OpenFlow tiene aplicaciones tales como la movilidad de máquinas virtuales (VM, Virtual Machine), redes de misión crítica y redes móviles de próxima generación basadas en IP.

OpenFlow lleva a cabo un proceso de señalización entre los controladores y los dispositivos de red tal y como se ilustra en la Figura 3 [1]. Los datos transportados obedecen a la trilogía "relación-acción- estadística" ("match-action-statistics") por lo que es preciso:

- determinar las transmisiones de manera única haciendo coincidir varios elementos tales como direcciones o números de puertos.
- especificar las acciones transmitidas desde el controlador a los dispositivos de red, tales como las tablas de reenvío o tablas de rutas, o más generalmente un dispositivo de reenvío.
- tablas de transferencia que contienen estadísticas sobre el grado de uso de los puertos, líneas de transmisión y nodos, como el número de bytes enviados a través de un puerto dado.

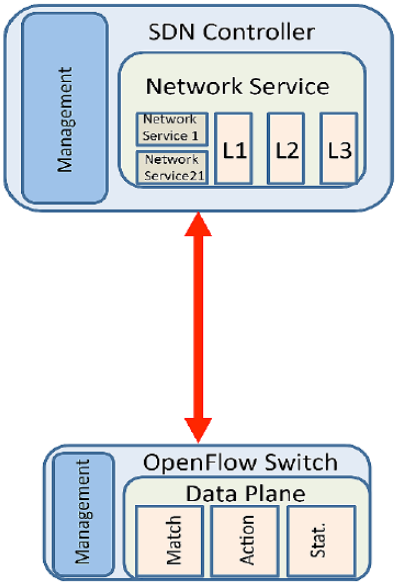

Figura 3. Protocolo de Señalización OpenFlow (Fuente [1])

A continuación se detallan las características que deben estar presentes en los controladores SDN:

- **Lenguaje de Programación:** Ejecutar multiplataforma, permitir multiprocesamiento, ser fácil de aprender, permitir un rápido acceso a la memoria y una buena administración de la memoria son características esenciales de los lenguajes de programación. Al elegir

un controlador determinado, se debe tener en cuenta estos factores porque afectan el rendimiento y la velocidad de desarrollo del controlador. Python, C ++ y Java son los idiomas más utilizados para la programación de controladores SDN.

- Soportar OpenFlow: El protocolo OpenFlow es un habilitador clave para redes definidas por software. Fue la primera interfaz estandarizada hacia el sur (Southbound API). Permite la manipulación directa del plano de reenvío de los conmutadores OpenFlow [8].

- Programabilidad de red: La programabilidad de red es el beneficio más importante de la introducción de SDN para lidiar con las complejidades de administración sin precedentes en la red actual con la explosión en la cantidad de dispositivos conectados y la implementación de nuevos servicios. El soporte del controlador de la programabilidad de la red se basa esencialmente en su grado de integración de un gran número de interfaces hacia el norte, una buena interfaz gráfica de usuario y una interfaz de línea de comandos [9].

- Eficiencia (Rendimiento, Fiabilidad, Escalabilidad y Seguridad)

- Interfaces hacia el Sur (Southbound Interfaces): Las Interfaces de Programación de Aplicaciones (abreviada como API del inglés: Application Programming Interface) en dirección sur permiten un control eficiente sobre la red [10]. Estas API son utilizadas por el controlador para realizar cambios dinámicos a las reglas de reenvío instaladas en los dispositivos del plano de datos que consisten en: conmutadores, enrutadores, etc.

- Interfaces hacia el Norte (Northbound Interfaces): La capa de aplicación utiliza las API hacia el norte para comunicarse con el controlador [11]. Son la parte más crítica en la arquitectura del controlador SDN, ya que la fortaleza de SDN está vinculada a las aplicaciones innovadoras que potencialmente puede admitir y habilitar.

- Afiliación: Al estar bajo buena supervisión de organizaciones asociadas el ONF, un controlador SDN tendrá posibilidades de mantenerse y mejorarse durante mucho tiempo [12]. Cisco, Linux Foundation, Intel, IBM, Juniper, etc. son ejemplos de organizaciones acreditadas que ingresan al mercado de SDN y participan en el desarrollo de controladores.

2.2.1 Funciones de Controladores SDN

En teoría, un controlador SDN proporciona servicios que pueden realizar un plano de control distribuido, así como también los conceptos de administración y centralización de estados efímeros. En realidad, cualquier instancia dada de un controlador proporcionará un segmento o subconjunto de esta funcionalidad, así como la propia interpretación de estos conceptos. [13].

Los controladores llevan a cabo diferentes funciones, como la provisión de infraestructura, la distribución (o no) de cargas en diferentes dispositivos de red para optimizar el rendimiento o reducir el consumo de energía. El controlador también está a cargo de la implementación de firewalls y servidores necesarios para el funcionamiento adecuado de la red y un sistema de gestión [14].

El controlador facilita la administración automatizada de la red y facilita la integración y administración de las aplicaciones comerciales.

2.4 Tipos de Controladores SDN

Las Redes definidas por software utilizan dos tipos de controladores los que están centralizados y los que son distribuidos. La figura 4 describe la clasificación de los diversos controladores en dos categorías.

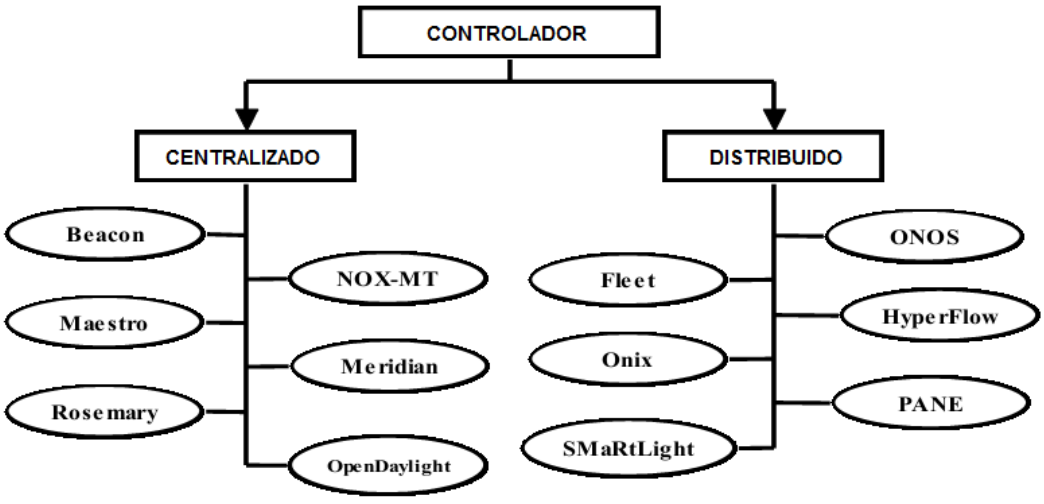

Figura 4. Clasificación de los Controladores (Fuente: elaboración propia)

Los controladores centralizados implementan toda la lógica del plano de control en una sola ubicación. En dicho controlador, el servidor único se ocupa de todas las actividades del plano de control. El principal beneficio de dicho controlador es la simplicidad y la gestión, ya que proporcionan un único punto de control. Sin embargo, adolecen de problemas de escalabilidad porque cada servidor tiene una capacidad limitada para tratar con dispositivos del plano de datos.

2.4.1 Controladores Centralizados

BEACON: [15] es uno de los controladores centralizados de código abierto más populares en SDN. Fue diseñado en la plataforma Java por la Universidad de Stanford y Big Switch Network en 2010. No estamos afirmando que fue el primer y único controlador de código abierto disponible porque NOX [16] se presentó originalmente como código abierto en 2008. Pero pronto surgió (en 2011) una solución multihilo llamada NOX-MT [17].

ROSEMARY: A veces sucede que la aplicación de red interfiere con el programa del controlador y le permite un mal funcionamiento. [18]. Rosmery ofrece la característica de resiliencia del controlador en el que las aplicaciones de terceros pueden interferir.

MAESTRO: Maestro [19] es un controlador multiproceso basado en Java de la Universidad de Rice. Explora la técnica de optimización de rendimiento adicional para lograr máximo rendimiento con aprovechamiento del paralelismo. Maestro presenta el concepto de procesamiento por lotes. Las solicitudes múltiples se agrupan en un solo lote de usuarios. Una vez que un hilo está libre, puede seleccionar cualquier solicitud pendiente disponible del lote y comenzar a ejecutarlo. La disponibilidad de hilos depende de la cantidad de núcleos. Maestro permite la ejecución de solicitudes de flujo múltiple por diferentes subprocesos de trabajo.

NOX-MT: El rendimiento del controlador a lo largo del desarrollo de la arquitectura SDN ha sido la principal preocupación. La comunidad SDN recibió varias sugerencias de los investigadores sobre esto. Kandula [20] presentó un clúster de aproximadamente 1500 servidores que enfrenta una solicitud de flujo de 100000 por segundo. Benson [21] denotó que una red que tiene 100 conmutadores puede experimentar solicitudes de flujo de 10 millones por segundo en el peor de los casos. Originalmente, NOX [13] se introdujo como

una plataforma de control de código abierto de un solo hilo para estudiar las características de rendimiento de la arquitectura SDN. Más tarde, se presentó una versión multiproceso de NOX como NOX-MT que utiliza el procesamiento por lotes de E/S para fines de optimización. El resultado obtenido de NOX-MT muestra una mejora en el rendimiento en un factor de 33 en comparación con el NOX. Aunque, NOX-MT muestra una mejora significativa en el rendimiento, aún no resuelve algunas de las dificultades de NOX, como por ejemplo el uso intensivo de la asignación de memoria dinámica, redundancia en copias múltiples para cada solicitud, etc.

MERIDIAN: El controlador Meridian se diseñó originalmente para la aplicabilidad de la arquitectura SDN en un entorno de nube. La idea era construir una red de nivel de servicio que pueda soportar características como la abstracción de políticas, alta conectividad en la nube. SDN se adapta perfectamente a la arquitectura de la nube ya sea en Infraestructura como Servicio (IaaS) o Plataforma como Servicio (PaaS).

OPENDAYLIGHT: El proyecto Opendaylight comenzó con el concepto de enfoque de ingeniería de software impulsado por modelos (model-driven software engineering, MDSE). Su arquitectura está inspirada en Beacon y hace uso de Open Service Gateway Interface (OSGi). MDSE consiste en un marco que define los modelos y las relaciones entre ellos. Los diferentes modelos se comunican entre sí mediante el lenguaje de modelado de datos. Estos modelos son independientes de la plataforma para soportar las diferentes necesidades de políticas comerciales. NETCONF y RESTCONF se utilizan como un protocolo de gestión de red impulsado por modelos.

La arquitectura de OpenDaylight (ver Figura 5) consiste en un conjunto de complementos hacia el Norte (North Bound) y hacia el Sur (South Bound) que están separados por la Capa de Adaptación de Servicio (Service Adaptation Layer, SAL).

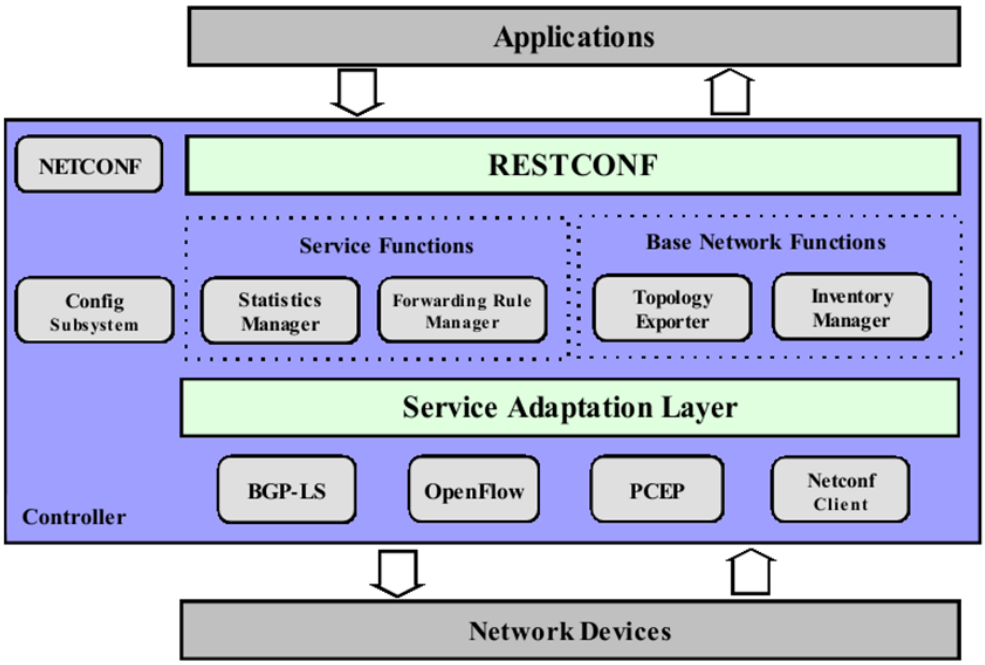

Figura 5. Arquitectura de OpenDaylight

2.4.2 Controladores Distribuidos

En comparación con los controladores centralizados, los controladores distribuidos tienen ventajas desde el punto de vista de escalabilidad y alto rendimiento durante el aumento de la demanda de solicitudes.

HYPERFLOW: HyperFlow [22] es el primer plano de control distribuido diseñado para OpenFlow. El diseño original de HyperFlow está inspirado en el NOX [23]. El diseño se distribuye debido a la disponibilidad física de diferentes controladores, pero forman un entorno lógicamente centralizado.

SMARTLIGHT: SMaRtLight [24] está diseñado para abordar el problema de tolerancia a fallas en la red. Se discuten tres aspectos de la falla que pueden ser: el conmutador o fallas de enlace en el plano de datos, falla de conexión entre el controlador y el conmutador en el plano de control y falla del controlador.

ONOS: El Sistema Operativo de Red Abierta (Open Network Operating System, ONOS) [34] es el controlador distribuido de código abierto diseñado para el entorno SDN. Está

diseñado principalmente para abordar la escalabilidad, la disponibilidad y el problema de rendimiento.

FLEET: Fleet [25] es uno de los primeros controladores que abordó el problema del administrador malicioso. La idea es evitar el mal funcionamiento del controlador debido a una mala configuración por parte del administrador. El administrador puede dañar la capacidad de enrutamiento, el reenvío y el funcionamiento del controlador. Se ha observado que los errores humanos son responsables del 50% al 80% de las interrupciones de la red [26]. El administrador de red puede degradar fácilmente el rendimiento del sistema al configurar incorrectamente el controlador.

ONIX: SDN requiere una plataforma de control común que permita implementar diversas funciones de control como enrutamiento, control de acceso, ingeniería de tráfico, etc. ONIX [27] se presenta como un controlador distribuido que ofrece una plataforma de control común escrita en C ++.

PANE: La idea del controlador PANE [28] sugiere que debería haber una API de configuración entre el usuario y el plano de control. En el escenario de red, ocurre que muchas veces la condición requerida no puede cumplirse al instante, pero podemos reservarla para el futuro. La API de configuración hace lo mismo. Aplica una mayor visibilidad y control sobre la red para realizar una reserva requerida. PANE se ocupa de dos problemas: descomposición del control y visibilidad en la red, resolución de conflictos entre los usuarios y sus solicitudes.

3 Análisis del Desempeño entre Controladores SDN

Esta sección discute el análisis del rendimiento entre varios controladores discutidos en la sección anterior. Se toman dos medidas, concretamente el rendimiento y el tiempo de respuesta para el análisis. El rendimiento define la cantidad de solicitudes de entrada que el controlador puede manejar por segundo. El tiempo de acceso define la latencia, que es un período de tiempo requerido por el controlador para procesar la solicitud. Cbench proporciona una plataforma para evaluar estos parámetros. Las características ofrecidas por Cbench son una medida del tiempo de respuesta máximo y mínimo para el controlador

independientemente de un número de conmutadores conectados, medición de rendimiento en entorno delimitado, es decir, número limitado de los paquetes, cálculo del rendimiento máximo, etc.

El desempeño del rendimiento del controlador se evalúa en entorno multihilo y entornos de un solo hilo o subproceso. Para el análisis comparativo del controlador, se supone que la configuración del sistema es tener procesador Intel Xeon E5-2870, 32-64 RAM con Ubuntu 11.10 o una VM superior. La Figura 6 define las respuestas del controlador en un entorno de un solo hilo en el que Onix lidera con 2,2 millones de solicitudes por segundo. Entre los controladores discutidos, Onix y Beacon son los únicos controladores que muestran un rendimiento superior a 1M / s. Onix tiene respuestas dobles en el período de tiempo de la unidad en comparación con otro controlador como NOX y ONOS. Ryu, Hyperflow y POX están entre los controladores que tienen menos tiempo de respuesta. Para el desarrollo de aplicaciones, que requiere un mayor rendimiento, Onix puede ser una buena opción.

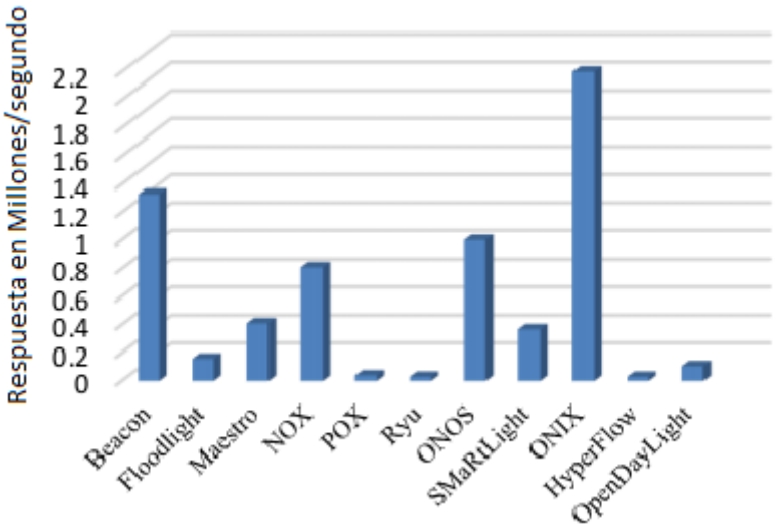

Figura 6. Rendimiento en el entorno de hilo único (single thread) (Fuente: elaboración propia)

Por otro lado, la Figura 7 muestra el rendimiento en un entorno multihilo donde Beacon nuevamente lidera en el rango 4.6 -11.9M respuestas / segundo. Desde el gráfico, podemos ver un crecimiento constante en el rendimiento de Beacon. Los hilos (thread) se toman de un mínimo de 2 a un máximo de 12. En el gráfico, Maestro muestra el

rendimiento hasta 8 hilos solamente. Sin embargo, su rendimiento es mucho mejor que Floodlight, que da 1.39M de respuestas/segundo cuando el número de hilos es 12. NOX tiene un desempeño ligeramente alto en comparación con Maestro, pero comparativamente muy bajo con Beacon.

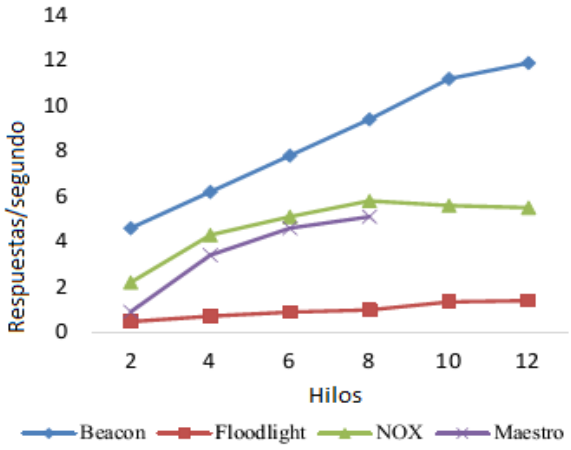

Figura 7. Rendimiento en entorno multihilo (multi thread) (Fuente: elaboración propia)

La Figura 8 muestra el rendimiento de latencia en función del tiempo de respuesta promedio para núcleo simple. Onix tiene menos tiempo de respuesta (latencia) entre los controladores. El controlador de POX muestra el tiempo de respuesta promedio en el intervalo de 100 a 200 que es un poco más alto que otros controladores. POX es el único controlador que tiene un tiempo de respuesta de más de 100µs. OpenDayLight, que muestra una latencia más alta que Onix, se encuentra en la categoría de controlador de baja latencia. PANE, NOX, Maestro y Floodlight tienen latencias similares entre sí.

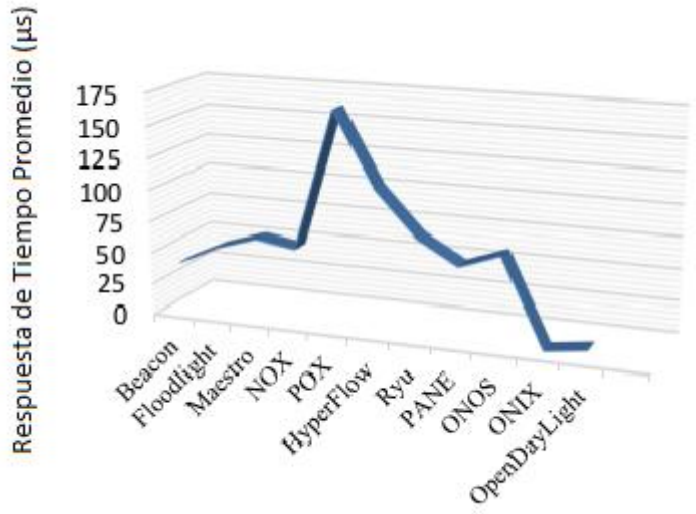

Figura 8. Latencia para hilo simple (Fuente: elaboración propia)

La Figura 9 muestra que Rosemary lidera en el gráfico hasta 10 hilos y funciona mejor que otros controladores como Beacon, etc. Pero cuando el número de hilos supera el 10, el rendimiento se estabiliza y ahora Beacon muestra un mejor rendimiento. Por lo tanto, está claro en la figura que Beacon será la elección correcta para un entorno multiproceso con un alto número de hilos. Por otro lado, Floodlight tiene un rango de rendimiento en 0.27-1.06 M de respuestas / segundo. La Figura 9 representa que inicialmente Floodlight y NOX se comportan de manera similar, pero cuando aumentamos gradualmente los hilos, la diferencia de rendimiento entre ellos se vuelve alta. NOX muestra el rendimiento cerca de unas 4 veces en comparación con Floodlight en 16 hilos.

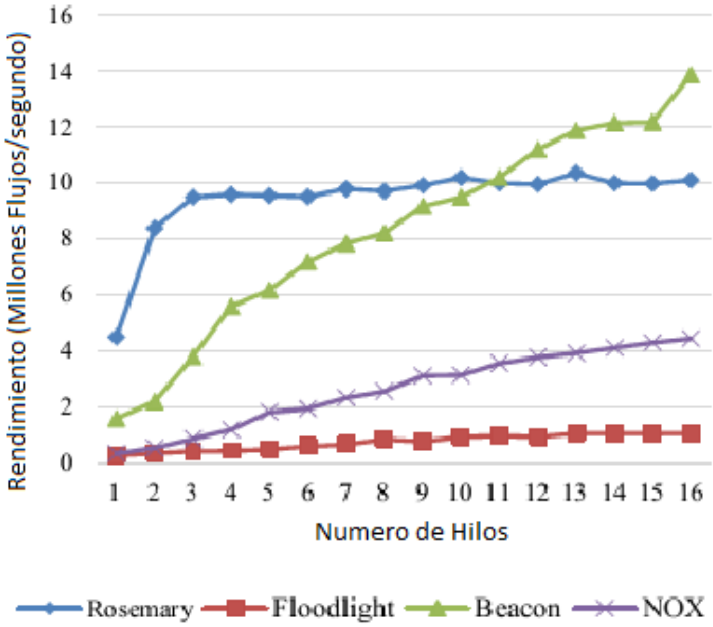

Figura 9 Comparación del rendimiento del controlador en entorno multihilo (Fuente: elaboración propia)

Basándose en el análisis general, Beacon muestra un rendimiento de 5M/s -12.8M/s con hilos en el rango de 2 a 12.

NOX también tiene un rendimiento casi óptimo para Beacon desde 5M/s-8M/s. Maestro carece de este rendimiento al tener un rendimiento de solo 1M/s-5M/s. Por lo tanto, en el escenario de rendimiento, Beacon se desempeña bien sobre NOX y Maestro. Se considera un caso similar para el cálculo de latencia que muestra un resultado de que Beacon tiene 24,7µs (25µs), Maestro tiene 55µs y NOX tiene 50µs de latencia.

4. Conclusiones

Las Redes Definidas por Software reforman el diseño de red existente al introducir el control de forma centralizada. La funcionalidad de control de enrutamiento en la ubicación centralizada disminuye el procesamiento en los dispositivos de la red. Toda la inteligencia de SDN viene desde el controlador que actúa como el cerebro del sistema. La capacidad del controlador se puede definir por la cantidad de solicitudes que puede gestionar desde los conmutadores. Varios módulos dentro del controlador se encargan de descubrimiento de red, descubrimiento de ruta, funcionalidad de flujo de empuje, etc.

El controlador centralizado proporciona una arquitectura simplificada, un manejo eficiente de los mensajes de solicitud pero no aborda el problema de escalabilidad. Por otro lado, los controladores distribuidos funcionan bien en el problema de escalabilidad y ofrecen el máximo rendimiento con alta disponibilidad pero requieren un procedimiento de intercambio de mensajes adecuado en el clúster.

Ambas categorías contienen controladores de código abierto, así como proveedores dedicados. Los controladores de código abierto como ONOS, Beacon, OpenDaylight brindan un amplio apoyo de la comunidad para pasar por el concepto de SDN en breve. Este trabajo ilustró una categorización de los controladores según sus usos en la industria y la academia. La selección del controlador depende de varios criterios, dependiendo del entorno (un solo hilo, multihilo) en el que se desarrollen. La elección del controlador para la academia puede diferir de la industria pero es vital en el diseño de una red SDN.

5. Referencias bibliográficas

[1] Pujolle, G. (2015). Software Networks: Virtualization, SDN, 5G and Security. London, ISTE Ltd.

[2] Recommendation ITU-T Y.3500 (2014), Information technology – Cloud computing – Overview and vocabulary.

[3] (IETF), I. E. T. F. (2014). Framework for Data Center (DC) Network Virtualization, Request for Comments: 7365.

[4] (IRTF), I. R. T. F. (2015). Software-Defined Networking (SDN): Layers and Architecture Terminology, Request for Comments: 7426.

[5] (IETF), I. E. T. F. (2014). Software-Defined Networking: A Perspective from within a Service Provider Environment, Request for Comments: 7149.

[6] Recommendation ITU-T Y.3300 (2014), Framework of software-defined networking

[7] Recommendation ITU-T Y.3301 (2016), Functional requirements of software-defined networking.

[8] Feng Wang, Heyu Wang, Baohua Lei and Wenting Ma, '"A Research on High-Performance SDN Controller," Cloud Computing and Big Data (CCBD), 2014 International Conference on, pp. 168-174.

[9] O.N. Fundation, '"Software-defined networking: The new norm for networks," ONF White Paper

[10] Sridhar Rao, "SDN Series Part Eight: Comparison of Open Source SDN Controllers" [online]. Available at: http://thenewstack.io/sdn-series-part-eight-comparison-of-open-source-sdn-controllers/.

[11] How Do SDN Southbound APIs Work?" [Online]. Available at: https://www.sdxcentral.com/resources/sdn/southbound-interface-api/.

[12] O.N. Fundation, '"Software-defined networking: The new norm for networks," ONF White Paper.

[13] Nadeau, T. D. (2018). SDN: Software Defined Networks by Ken Gray, . S. B. Online.

[14] Paliwal, M., et al. (2018). "Controllers in SDN: A Review Report." IEEE Access: 1-1.

[15] D. Erickson, "The Beacon OpenFlow controller," inProc. 2nd ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw., pp. 13-18, 2013.

[16] N. Gude et al., "NOX: Towards an operating system for networks,"Comput. Commun. Rev., vol. 38, no. 3, pp. 105-110, 2008

[17] A. Tootoonchian, S. Gorbunov, Y. Ganjali, M.Casado and R. Sherwood, "On controller performance in software-defined networks," inProc. 2nd USENIX Conf. Hot Topics Manage. Internet Cloud Enterprise Netw. Services, 2012.

[18] S. Shin et al., "Rosemary: A robust, secure, high-performance network operating system," inProc. 21st ACM Conf. Comput. Commun. Security, Scottsdale, AZ, USA, pp. 78-89, 2014

[19] Z. Cai, A.Cox, "Maestro: A system for scalable OpenFlow control," Rice Univ., Houston, TX, USA, Tech. Rep., 2011

[20] S. Kandula, S. Sengupta, A. Greenberg, P. Patel and R. Chaiken, "The nature of data center traffic: measurements & analysis," inProceedings of IMC, ACM, pp. 202-208, 2009

[21] T. Benson, A. Akella and D. Maltz, "Network traffic characteristics of data centers in the wild," in Proceedings of IMC, ACM, 2010.

[22] A. Tootoonchian and Y.Ganjali, "HyperFlow: A distributed control plane for OpenFlow," in Proc. Internet Netw. Manage. Conf. Res. Enterprise Netw., 2010

[23] N. Gude et al., "NOX: Towards an operating system for networks," Comput. Commun. Rev., vol. 38, no. 3, pp. 105-110, 2008

[24] F. Botelho, A. Bessani, F. Ramos and P. Ferreira, "SMaRtLight: A Practical Fault-Tolerant AND Controller," Cornell University Liberary, Networking and Internet Architecture, 2014

[25] S. Matsumoto, S.Hitz, and A.Perrig, "Fleet: Defending SDNs from malicious administrators," in Proc. 3rd Workshop Hot Topics Softw. Defined Netw., pp. 103-108, 2014.

[26] Juniper Networks, What's behind network downtime? (2008)

[27] T. Koponen et al., "Onix: A distributed control platform for large-scale production networks," in Proc. 9th USENIX Conf. Oper. Syst. Design Implement., pp. 1-6, 2010.

[28] A. Ferguson, A. Guha, C. Liang, R. Fonseca and S. Krishnamurthi, "Participatory networking: An API for application control of SDNs," in Proc. ACM SIGCOMM Conf., pp. 327-338, 2013