Revista de I+D TECNOLÓGICO
Portal de revistas: http://revistas.utp.ac.pa

Diseño de un entorno de pruebas SDN para soportar el IdC: prototipo y evaluación

Designing an SDN testbed environment to support the IoT: prototype and evaluation

Carlos González [1*], Olivier Flauzac [2], Florent Nolot [2]

[1] *Vicerrectoría de Investigación y Postgrado UNACHI, Panamá*
[2] *Université de Reims Champagne-Ardenne, CReSTIC, Francia*

***Autor de correspondencia:** carlos.gonzalez5@unachi.ac.pa

RESUMEN– La Virtualización de las Funciones de Red (NFV) surge como una de las tecnologías más prometedoras para la gestión de la nueva generación de Internet. En los últimos años, los sistemas informáticos y de comunicación han evolucionado enormemente y han influido en el desarrollo de las infraestructuras de red en términos de escalabilidad, programabilidad y gestión dinámica. El número de dispositivos conectados crece exponencialmente con el desarrollo de Internet de las Cosas (IdC) y de múltiples aplicaciones en línea. Con la evolución de nuevas tecnologías emergentes, aparece el Software Defined Networking (SDN) y la NFV, permitiendo una gestión flexible, dinámica y adaptable para optimizar los recursos de la red. En un entorno de pruebas de IdC, se desarrolló una arquitectura de red virtual, la cual proporciona una plataforma que nos permite evaluar la orquestación de un controlador SDN distribuido. Se evaluó en términos de simulación y experimentación aspectos sobre el tratamiento flujo de datos masivos, el análisis de tráfico de red granulado y la utilización de los recursos de cada uno de los dispositivos conectados. Este trabajo de investigación pretende mejorar significativamente la ingeniería de trafica a gran escala, con un enfoque de distribución de la gestión de carga de los nodos controladores, permitiendo una gestión dinámica y flexible. Los resultados experimentales muestran un buen rendimiento de la plataforma de pruebas desarrollada.

Palabras clave– *Administración de redes de computadoras, internet de las cosas, openflow, redes definidas por software, Virtualización*

ABSTRACT– The Network functions virtualization (NFV) emerges as one of the most promising technology for the management of the next Internet generation. In recent years, computer and communication systems have considerably evolved influencing the development of network infrastructures in terms of scalability, programmability and dynamic management. The number of connected devices grows exponentially with the development of the Internet of Things (IoT) and a plethora of online applications. The research community has focused its efforts on optimizing network administration with implementation and configuration techniques improving the network performance. With the evolution of new emerging technologies, appears the Software Defined Networking (SDN) and the Virtualization of Network Functions (NFV), which allow a flexible, dynamic and adaptable management to optimize network resources. In an IoT environment, we developed a virtual network architecture, which provides a test platform that allows us to evaluate the orchestration of a distributed SDN controller, obtaining a scalable and flexible management. This research work aims to enhance the large-scale traffic engineering, with a distributed approach including the load-balanced management of controller nodes, allowing dynamic and flexible management. The experimental results show a good performance of the developed testbed platform.

Keywords– *Computer network management, internet of things, openflow, software-defined networking, virtualization.*

1. Introducción

En los últimos años, con la proliferación de dispositivos conectados a Internet, se ha desarrollado un nuevo paradigma de red, el Internet de las Cosas (IdC). Debido a la constante evolución, los riesgos de seguridad y las dificultades para gestionar las redes en gran escala aumentan considerablemente [1]. Al tratarse de una de las tecnologías emergentes más importantes, el IdC concierne actualmente, a una gran parte de la industria y la comunidad académica. Los sistemas informáticos incluidos en este nuevo paradigma de comunicación digital, nos permiten programar objetos relacionados con muchos aspectos de nuestra vida cotidiana. Además, el IdC puede ser utilizado para supervisar y controlar de forma autónoma, diversos entornos, como las ciudades inteligentes, el medio ambiente, la educación, la salud, el transporte, entre otros. Debido al gran número de dispositivos interconectados, se genera un gran volumen de datos, con lo que se establece un ecosistema de IdC con nuevos requisitos de redes escalables, de seguridad y de privacidad. Según las proyecciones de los sistemas de comunicación, se espera para el año 2020, unos 26 mil

Citación: C. Gonzalez, O. Flauzac y F. Nolot, "Diseño de un entorno de pruebas SDN para soportar el IdC: prototipo y evaluación," *Revista de I+D Tecnológico*, vol. 16, no. 1, pp. (69-77), 2020.
Tipo de artículo: Original. **Recibido:** 25 junio de 2019**. Recibido con correcciones:** 25 junio de 2019. **Aceptado:** 2 diciembre de 2019.
DOI.

millones de dispositivos conectados [2]. En cuanto al grado de complejidad y heterogeneidad de los sistemas, las nuevas arquitecturas de red requieren un desarrollo adaptable a la cuarta generación industrial, con conectividad omnipresente entre máquinas y objetos. Una solución prometedora es proporcionada por otra tecnología emergente, denominada Redes Definidas por Software (SDN) [3].

La función de control es separada de la función de plano de transferencia de datos, lo que proporciona una arquitectura de red escalable, programable y dinámica. En la función de control, la inteligencia se centraliza en un nodo o puede distribuirse en múltiples nodos controladores SDN. Los equipos de red tradicionales son propietarios, lo que dificulta la programación, la innovación y el despliegue de la escalabilidad, que puede dar soporte a redes emergentes como el IdC. Con su entorno de código abierto, es posible programar funciones de red y personalizar las aplicaciones de administración y seguridad de los datos.

Con el desarrollo de un entorno de pruebas, permite evaluar la capacidad de enrutamiento del protocolo *Openflow*, y el balance de carga de los controladores SDN, el cual son uno de los principales inconvenientes mencionados en algunos estudios realizados previamente en [4], [5], [6]. La principal motivación al realizar el prototipo de pruebas, se basa en la literatura científica sobre el presente y el futuro de las redes donde el tráfico será realmente abundante. Una gran parte se centra en conceptos formales y arquitecturas teóricas, que se evalúan mediante modelos analíticos o simulaciones básicas. Aunque los trabajos teóricos son indispensables para el avance del conocimiento y la innovación, los entornos de pruebas y los prototipos son esenciales para demostrar el correcto funcionamiento y el rendimiento de estas tecnologías. El diseño y validación de algunos entornos de pruebas han sido presentados en diversos trabajos de investigación [7], [8], [9], [10] .

Los entornos de pruebas desarrollados están basados en Mininet o utilizando controladores SDN de forma centralizada, lo que impide la evolución en términos de redes emergentes. En este artículo presenta el desarrollo de una plataforma SDN virtualizada, que realizar pruebas de escalabilidad, programabilidad, flexibilidad y la distribución de reglas de flujos de datos para la administración de los dispositivos integrados de tipo IdC.

1.1 Arquitectura de IdC

Con el desarrollo de una amplia variedad de dispositivos integrados con alta demanda de conexión a internet, se genera un gran volumen de datos, que requiere de una arquitectura de red capaz de incluir la recolección, procesamiento y almacenamiento de información [11]. Actualmente, existen varios modelos de arquitectura, como caso de uso de IdC. Tanto los sectores académicos como industriales proponen diversas soluciones en función de las necesidades y de los productos desarrollados. Los modelos propuestos varían según el número de capas de comunicación o los protocolos utilizados. Sin embargo, actualmente no hay consenso para un único tipo de arquitectura de IdC. La estandarización de un modelo universal evoluciona con las aportaciones de diferentes desarrolladores de esta tecnología.

La tecnología emergente en mención ofrece una amplia gama de oportunidades para crear valor agregado en los sectores académicos e industriales. En una de las proyecciones publicadas recientemente, cisco prevé unos 6.5 dispositivos conectados por persona, así como unos 156 exabytes de tráfico de datos al mes y unos 1.5 zettabytes al año 2018 [12].

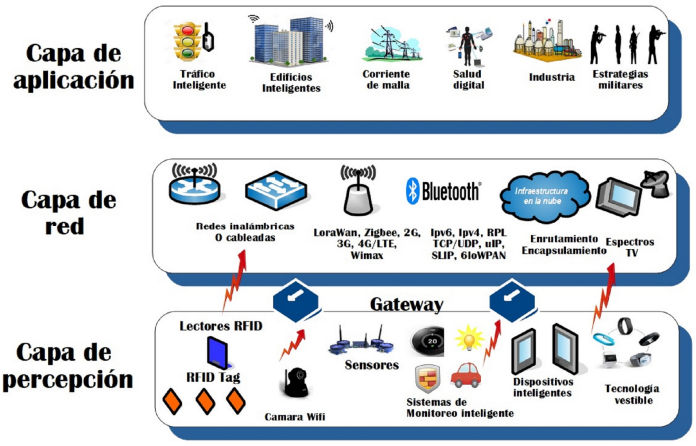

Figura 1. Arquitectura de la Internet de las Cosas [13].

Debido al desarrollo de la tecnología de IdC, es importante mencionar algunas de las arquitecturas propuestas. Sin embargo, tras realizar un estudio profundizado de las principales características de estas arquitecturas, se han simplificado las funcionalidades, para entender el tratamiento de los datos. Desde esta perspectiva, se puede definir una arquitectura de tres capas: la capa de percepción, la capa de red y la capa de aplicación (figura. 1).

- Capa de aplicación: permite implementar el desarrollo de aplicaciones para interactuar con usuarios, sectores industriales, como la domótica, ciudades inteligentes, logística, medio ambiente, seguridad pública, sanidad. Además, se pueden integrar las funciones de decisión de control y seguridad.
- Capa de red: su función principal es establecer un vínculo entre la capa de percepción y la capa de aplicación. La transferencia de datos se realiza a través de diferentes tecnologías y protocolos que garantizan el flujo de la información.
- Capa de percepción del artículo: incluye los objetos físicos conocidos como sensores y actuadores. El objetivo principal de estos dispositivos es recopilar datos e identificar otros objetos utilizando radiofrecuencia RFID y redes inalámbricas.

La heterogeneidad de las múltiples redes crea una administración compleja para abordar temas de escalabilidad y la seguridad en entornos de IdC. La comunidad científica está centrando sus esfuerzos en una tecnología emergente basada en el concepto SDN, siendo una de las tecnologías más prometedoras para este tipo de redes.

1.2 Arquitectura de SDN

La complejidad en la administración de las redes tradicionales, se debe al acoplamiento del plano de control y al plano de datos. Cada dispositivo de interconexión de red se gestiona de forma individual, lo que impide la evolución e innovación al implementar políticas y reglas de flujo de datos dinámicos. La mayoría de los sistemas son cerrados y propietarios, lo que impide el acceso al despliegue de nuevos protocolos de comunicación o políticas de seguridad, con un proceso extenso y complejo a desarrollar.

Con una gestión dinámica, la arquitectura SDN simplifica la gestión de la red al separar las funciones de control del plano de datos. En lugar de configurar individualmente cada dispositivo de interconexión de red, un nodo denominado controlador SDN gestiona el plano de control de forma centralizada. La transferencia de paquetes está organizada por tablas de enrutamiento en nivel del plano de datos. Todas las decisiones de enrutamiento son realizadas por el controlador, dada la visión global de la red.

La figura 2 muestra la arquitectura SDN, compuesta por tres capas: capa de aplicación, capa de control y capa de infraestructura.

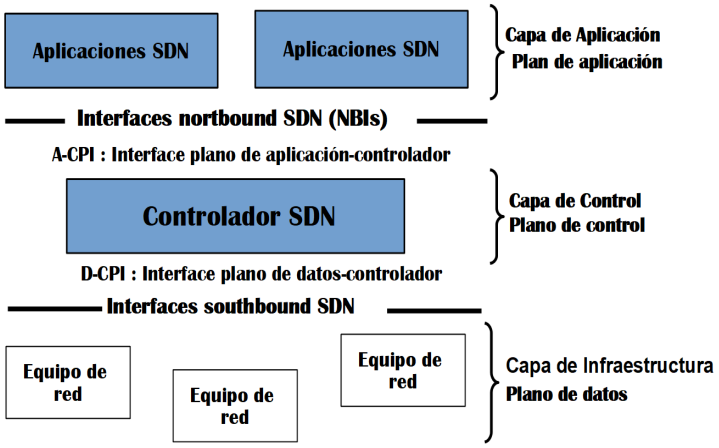

Figura 2. Arquitectura SDN [14].

- La capa de aplicación permite a los administradores de red configurar, gestionar, automatizar y optimizar los recursos de red, a través de una interfaz de programación de aplicaciones.
- La capa de control utiliza una política de gestión de datos, basada en algoritmos y protocolos específicos. Las directivas de red se pueden distribuir entre varios controladores SDN.
- En la capa de infraestructura, se incluyen todos los dispositivos de interconexión de red, donde el tráfico es reenviado o procesado con las decisiones de transferencia recibidas del controlador.

El protocolo de comunicación más utilizado para acceder al plano de datos es *Openflow* (OF) y está estandarizado por la *Open Networking Foundation* [14]. Actualmente, una gran parte de los fabricantes de equipos de red han desarrollado sistemas con capacidades para soportar el SDN. Los mensajes intercambiados entre el controlador y los conmutadores OF pueden ser cifrados con TLS. Cada conmutador OF contiene tablas de flujos de datos con un conjunto de entradas, que representan las reglas de enrutamientos. De esta manera, el controlador SDN almacena información de los dispositivos de red y establece las configuraciones necesarias.

2. Trabajos Relacionados

La escalabilidad del recurso NFV ha sido objeto de estudio, y existen algunas técnicas de escalabilidad para integrar nuevas tecnologías emergentes. El estudio en [15], destaca el Objeto Virtual (VO) como un componente clave para proporcionar una gestión energéticamente eficiente, la heterogeneidad y la escalabilidad para el IdC. Los autores proponen un dispositivo inteligente como servicio (SDaaS), que incluye el VO de cada uno de los dispositivos físicos. La arquitectura virtual propuesta incluye un modelo de plataforma como servicio (PaaS), virtualización de funciones de Red (NFV) y técnicas de *Fog Computing*. La solución en nube PaaS está situada cerca del borde de la red, lo que permite la comunicación directa con los dispositivos virtuales finales. De esta manera, la plataforma de pruebas proporciona un resultado de rendimiento de red y una visión general de las funcionalidades de VO. Sin embargo, la solución propuesta no dispone de información sobre la integración de los servicios *cloud* con NFV para la toma de decisiones de transferencia de datos.

La investigación en [16], presenta una arquitectura escalable basada en NFV. Desde esta perspectiva, este trabajo aborda los beneficios y los retos que plantea el apoyo a la creciente tecnología de IdC prevista. Obviamente, la creciente movilidad y la computación ubicua emergen como uno de los retos más importantes para una integración con el NFV. La conectividad de red en la interfaz en dirección sur, *Southbound* es proporcionada por pasarelas de IdC, y el protocolo IP comunica la información recopilada a la interfaz *Northbound*. Todas las funciones de red y la aplicación de IdC están instaladas en un centro de datos. Como parte del trabajo futuro, los autores planean la integración de una plataforma de implementación, para evaluar el enfoque propuesto.

En [17], la arquitectura SDIoT extiende la alta escalabilidad, gestión y seguridad de IdC. Esencialmente, esta arquitectura consiste en tres capas: capa física, capa de *middleware*/control y capa de servicio de datos. La capa física incluye los dispositivos de punto final. La capa de middleware/control consiste en un bloque de aplicaciones definidas por *software*: la seguridad definida por *software* (SDSec), el almacenamiento definido por *software* (SDStore), el controlador de Internet de objetos (IoT-C) y el controlador definido por software (SDN-C). Los datos recogidos se procesan desde una pasarela de IdC. El SDSec procesa la autenticación de los dispositivos conectados. En caso de una autenticación exitosa, los datos se etiquetan, incluyendo un indicador positivo (P); de lo contrario, un indicador se etiqueta como (N). A continuación, el IoT-C calcula la ruta de datos a los dispositivos de destino. El SDN-C procesa las reglas de reenvío que se añaden a los conmutadores de red. En cuanto a la escalabilidad, la arquitectura SDIoT cuenta con muchas aplicaciones definidas por *software*, que evitan la evolución del entorno de IdC.

Erran *et al.* en [18], proponen una plataforma denominada CellSDN. Su objetivo es simplificar la gestión de las redes celulares. Basados en aplicaciones SDN, especifican las políticas atribuidas y las reglas de reenvío de cada dispositivo de usuario final, lo que permite un control detallado sobre la red LTE. El agente de control local agregado a los conmutadores, es clave para manejar la inspección profundizada de paquetes. El proceso de verificación puede monitorear aplicaciones de flujo de tráfico, como video, peer-to-peer, web y VoIP. Con una alta eficiencia, el rendimiento de estos agentes locales debe aumentar la escalabilidad y reducir la carga excesiva en múltiples controladores, para proporcionar un tiempo de reacción rápido, que permita superar los eventos críticos. Una extensión de este trabajo fue presentada en [19]. La arquitectura *SoftCell* propuesta se basa en cuatro componentes principales: el controlador, los interruptores de acceso, los interruptores de núcleo y las cajas intermedias. Las casillas centrales definen las reglas de seguridad o aplicaciones de transcodificación para la transferencia multimedia recibida por el controlador y procesada en nivel de conmutador. Los flujos de tráfico recibidos de los usuarios finales, se verifican con precisión en el nivel del interruptor de acceso, para poder ubicarlos en la estación base. Cada conmutador de acceso incluye un agente local, que especifica una clasificación de los paquetes recibidos de los usuarios finales. Los flujos se procesan localmente, buscando minimizar la sobrecarga entre controladores. Los conmutadores centrales actúan como una pasarela de conexión a Internet. El tráfico de red es reenviado desde la pasarela, a través de los conmutadores centrales. En esta arquitectura, las redes LTE no necesitan elementos de red especializados, incluidas las pasarelas de servicio

(S-GW) o las pasarelas de red de datos por paquetes (P-GW).

Una investigación similar se realizó en [20], donde los autores presentan un marco de monitorización con virtualización de funciones NFV/SDN, que integra el proyecto SONATA, para soportar aplicaciones y servicios 5G. La escalabilidad se basa en un sistema distribuido de monitorización en cascada. Una pasarela *push* recoge información del contenedor LXC, para enviarla al servidor de monitorización. Cada contenedor define un punto de presencia (PoP), que se comunica con su respectivo servidor *websocket.* El marco es una iniciativa para monitorear los servicios 5G.

La integración de dispositivos de IdC con SDN ha sido desarrollado en diversos entornos de pruebas y plataformas de simulación. Bellavista *et al*. [4] describen una arquitectura en entorno de pruebas de dominios de IdC haciendo uso de MATLAB. Otra herramienta utilizada para la emulación de redes definidas por *software* es Mininet. Seeger *et al*. implementaron Mininet para medir las limitaciones de SDN al configurar QoS en los dispositivos de IdC [5]. La arquitectura SDN para el IdC propuesta por Sinh *et al*. [6] utilizan Mininet en el desarrollo del prototipo de simulación. En entornos de pruebas minimizados la distribución de controladores SDN haciendo uso de Mininet presenta un sistema flexible y plataformas escalables [7] . Otra solución interesante que permite la integración de dispositivos de IdC y SDN es el uso de *rapsberry pi*. Kim *et al*. [8] desarrollaron una plataforma con controladores SDN, conmutadores y dispositivos de IdC haciendo uso de múltiples *raspberry pi* en cada uno de los equipos utilizados. Schaerer *et al*. [9] implementan una *raspberry pi* como *gateway* entre los dispositivos IdC y el controlador SDN. Existen otras soluciones como entornos de pruebas con la implementación de la computación en nube y el uso de contenedores [10], [21].

Las soluciones propuestas se han desarrollado con el propósito obtener la integración y la escalabilidad del IdC haciendo uso de SDN. Sin embargo, estas propuestas carecen de una plataforma de evaluación experimental para el control masivo del flujo de datos, el uso flexible y dinámico de los controladores SDN.

3. La Solución Escalable de SDN Propuesta

Debido al gran número de dispositivos de IdC conectados, es necesario disponer de infraestructuras de redes capaces de soportar la escalabilidad. Desde un punto centralizado, el SDN hace posible la creación de un sistema de automatización con protocolos que permitan una gestión macro de los flujos de datos [22].

También con el SDN es viable predefinir las políticas de comunicación y seguridad de los dispositivos conectados e, incluso, definir estas políticas antes de la solicitud de conectividad, lo que, básicamente, posibilita una administración dinámica, sin tener en cuenta los nuevos dispositivos conectados. SDN permite incluir una escalabilidad inherente, debido a su concepción centralizada para la administración de aplicaciones y protocolos, así como generar una respuesta rápida para expandir redes escalables.

Con la abstracción de equipos de red, como los conmutadores, enrutadores y dispositivos intermedios, tiene lugar una reducción significativa de costes [3]. Estos equipos en redes tradicionales utilizan sistemas propios programados, con reglas específicas y protocolos complejos, para establecer las comunicaciones. Por lo tanto, la configuración de políticas adecuadas para satisfacer los requisitos específicos de las aplicaciones de IdC, constituye un reto que puede resolverse, aplicando tecnologías de redes definidas por *software*.

La arquitectura propuesta para mejorar la escalabilidad y flexibilidad en la administración de número elevado de dispositivos, introducimos el concepto del Internet de las Cosas definidas por *software* (ICDS), asumiendo que cada controlador contiene cientos o miles de dispositivos por administrar. Normalmente, una red en gran escala no puede lograr, eficientemente, una estructura organizativa. Por esta razón, proponemos el *cluster* SDN, considerando que cada controlador gestiona su propio dominio. Cada ICDS se encarga de gestionar los dispositivos de los puntos finales de la operación (figura 3). Con esta arquitectura de *clustering*, los datos del entorno de red se procesan en nivel de controlador y se envían al controlador más cercano. Además, al tener una visión global de todo el dominio, el nodo controlador puede monitorizar, suministrar y definir reglas de flujo reactivas. En la clusterización a través de Atomix del proyecto Onos, el nodo controlador descubre a sus pares vecinos, mediante mecanismos de descubrimiento dinámicos. El *cluster* Atomix ONOS puede soportar la tolerancia de fallo de alguno de los nodos agrupados. Cada controlador tiene control total de acceso a los

switches y a las reglas del flujo de datos. Basándose en esta arquitectura de *clusterización*, es posible configurar la gestión de la red, procesar los datos recogidos y agregar la información en el dominio o distribuir la información a otro controlador.

Atomix se configura individualmente en cada controlador ONOS. Cada agente Atomix está encargado de almacenar la información del clúster. Una vez iniciado, un agente Atomix comienza el proceso de descubrir los nodos que forman parte de un grupo configurado en el archivo de configuración. Para asignar los agentes Atomix a un controlador Onos, es necesario configurar el archivo *cluster.json* con la dirección IP de cada agente y los puertos de comunicación.

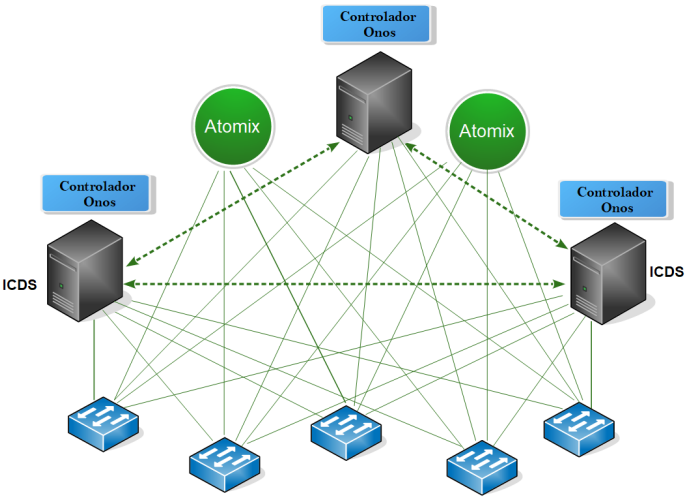

Figura 3. Arquitectura clúster escalable con Onos.

La interfaz web de ONOS GUI facilita la identificación de las instancias, los conmutadores y los dispositivos finales conectados a los controladores. Una vez se tiene acceso, se puede añadir o eliminar flujos de datos, y con una vista relativamente completa del entorno virtualizado, es fácil ver el tráfico, desplazar los dispositivos y obtener detalles de cada uno de los enlaces.

El despliegue de un modelo de alta disponibilidad basado en *cluster* se basa en el proyecto Onos. El hecho de mejorar el rendimiento de la red, la tolerancia a fallos y gestión de red escalable introduce nuevos retos para desplegar controladores SDN. Debido a que cada controlador SDN agrupado tiene una vista parcial de su dominio e intercambia información completa, se produce una sobrecarga del sistema. Al utilizar múltiples controladores, si uno de ellos falla, otro puede tomar el control sobre toda la red. El modelo de clúster propuesto proporciona un alto nivel de escalabilidad, una gestión flujos macros y una fácil gestión de implementación.

4. Implementaciones

En esta sección, se describe la plataforma desarrollada para lograr una escalabilidad de IdC y las aplicaciones utilizadas, así como el diseñado específico para lograr la integración con las tecnologías del SDN. Nuestro entorno de IdC proporciona virtualización de dispositivos integrados, *Openvswitch* y controladores SDN.

Después de analizar las diferentes plataformas y herramientas que existen actualmente, resulta evidente que no hay una que se adapte a un escenario realista de experimentación SDN-IdC. Por esta razón, construimos una plataforma de pruebas, diseñada en un entorno virtualizado, el cual permite insertar flujo de datos entre dispositivos finales, monitorear las peticiones de conexión, verificar las reglas y políticas para cada dispositivo conectado al clúster. La plataforma desarrollada consiste en una red externa para la gestión remota y una red interna para establecer la comunicación *Openflow*. En una primera aproximación, la plataforma de pruebas incluye ocho servidores Linux, dos servidores para la instalación del *cluster* atomix, un controlador ONOS y cinco servidores para la virtualización de los dispositivos integrados de IdC.

La gestión de la infraestructura se realiza en un entorno de *cloud computing*, basado en *VMware vSphere*. Con este *software* es posible gestionar, instalar y configurar múltiples máquinas virtuales, simultáneamente, en un único nodo físico. La emulación de la arquitectura de un dispositivo con capacidad de IdC, se realiza con la herramienta de código abierto Qemu. El emulador Qemu permite adaptarlo a las necesidades del sistema, como el tipo de procesador, interfaces de red, disco duro, memoria RAM, entre otros. Existen varios sistemas operativos para simular dispositivos con recursos limitados de tipo IdC, como Contiki, RIoT, LiteOS. Sin embargo, escogimos TinyOS, ya que contiene varias funciones diseñadas para permitir la escalabilidad y la integridad en una arquitectura, pruebas para dispositivos con capacidades de IdC.

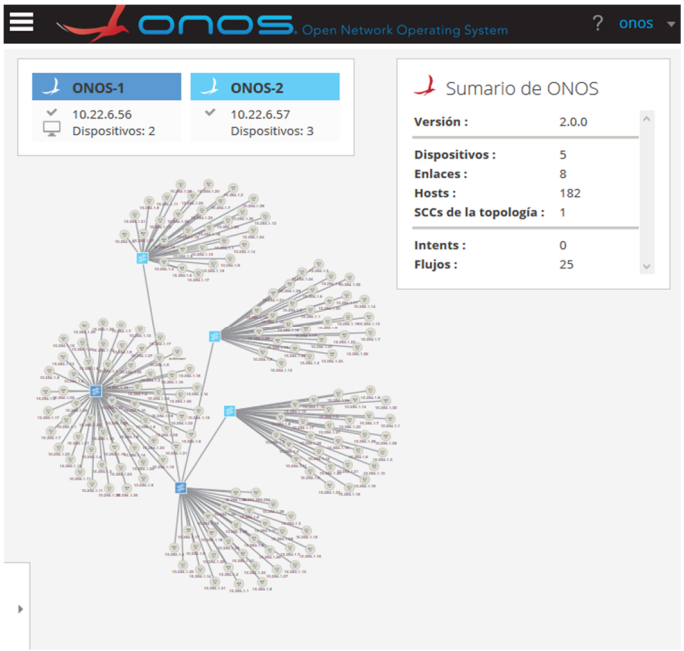

Figura 4. Controladores Onos entorno de pruebas.

La plataforma de prueba SDN consiste en herramientas de código abierto, como Onos y *OpenvSwitch* (OVS) (figura 4). El proyecto Onos es compatible con la distribución Windows y Linux, así como con algunos desarrolladores de dispositivos de red importantes. El protocolo *Openflow* está configurado para crear un canal de comunicación entre el controlador y los conmutadores OVS. La aplicación de gestión de control y las decisiones de transferencia de datos son gestionadas por *Openflow*, y los datos se organizan en tablas de flujo. El controlador Onos tiene un plug-in integrado, que permite el proceso de automatización de conmutación en IPv4 e IPv6 (figura 5). Sin embargo, el proceso de enrutamiento para una escalabilidad óptima, requiere muchas operaciones sucesivas, con el fin de añadir flujos a los conmutadores *Openflow*. Para permitir el proceso de automatización de redes escalables de SDN con *Openflow*, es necesario un amplio conocimiento de la segmentación de tablas de flujo, incluyendo las instrucciones de enrutamiento, así como scripting en Shell o Python.

El escenario experimental permite verificar el buen rendimiento de *Openflow* en redes escalables, utilizando un número importante de dispositivos conectados simultáneamente, organizados en clúster. En el caso de uso experimental, se utilizó el protocolo *OpenFlow* versión 1.3. En cuanto a las operaciones de escalabilidad, *Openflow* permite la implementación de políticas de seguridad, de forma dinámica y flexible, a través de algoritmos programados en Shell script y Python. Sin embargo, el protocolo OF no proporciona las herramientas adecuadas para adaptar reglas de flujos al nivel de la capa de aplicación. *Openflow* solo permite la gestión de las decisiones de transferencia de datos y de las tablas de flujo de comunicación. Por lo tanto, una vez que se instalan las reglas de flujo, la transferencia de conmutación y de enrutamiento únicamente la realizan los conmutadores OVS.

La programación de una red a gran escala de capa 3, las decisiones de reenvío de IPv4 *Openflow* requieren cientos de miles de flujo de datos en un solo *switch*, para permitir la implementación de redes escalables. Esto crea un nuevo reto en el intento de interconectar múltiples dispositivos en nuestra plataforma de pruebas desarrollada. Para configurar un sistema escalable *Openflow*, hemos desarrollado un algoritmo, con el propósito de simplificar las entradas flujos de datos añadidas a cada uno de los OVS. Los flujos de datos instalados permiten la gestión de toda la red, mediante el controlador Onos dado a la visión global de todo el entorno de la red. La gestión de reglas de enrutamiento de forma dinámica a través de OF, necesita una puerta de enlace para comunicar diferentes subredes. El método utilizado se realiza con resoluciones ARP entre los OVS más cercanos, permitiendo el flujo de tráfico en gran escala. Debido a la gran cantidad de flujo de datos que deben procesar los OVS, es preciso organizar las reglas de encaminamiento en tablas de flujos. Cada tabla de flujo de datos tiene tareas específicas, desde obtener las direcciones MAC de cada dispositivo conectado, hasta el procesamiento de datos entre cada una de las subredes. El enrutamiento es de tipo dinámico, y puede definirse de acuerdo la configuración reactiva o pasiva en cada uno de los conmutadores OF

El clúster Onos utiliza una elección distribuida del controlador principal, al emplear el protocolo Raft. Una vez que los controladores líderes electos están conectados en el *clúster*, los miembros del grupo comparten una visión global de la red. El tiempo de elección depende de cada proceso de descubrimiento, el cual empieza inmediatamente después de iniciar cada miembro del *clúster*. El primer miembro ya iniciado tiene una alta probabilidad de ser seleccionado como controlador principal. Cada nodo cliente agrupado recibe

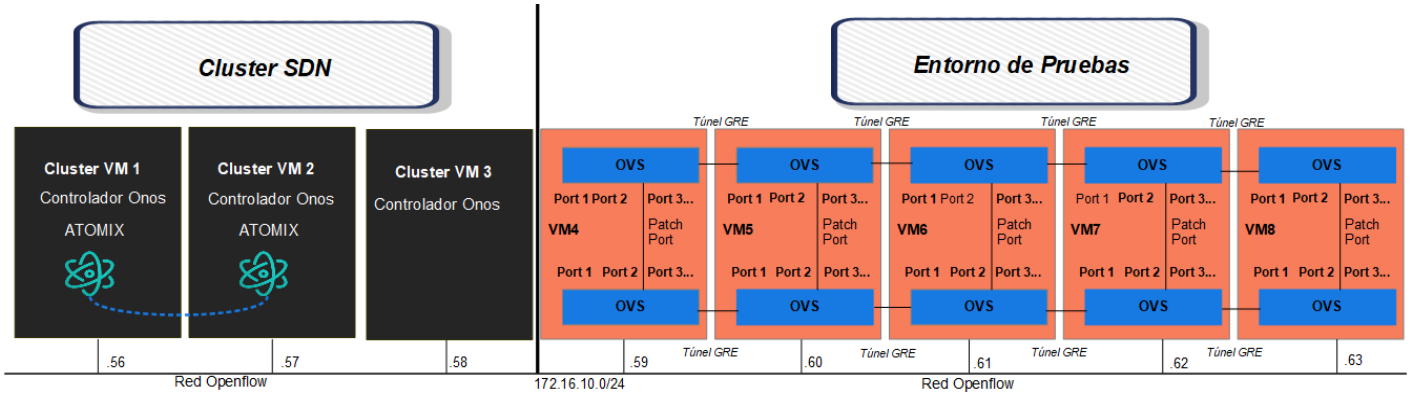

Figura 5. Plataforma de pruebas clúster SDN.

una entrada del líder para unirse al clúster. El líder recibe las solicitudes de respuesta de los otros clientes del clúster y comparte el inventario y topología de la red.

El proceso de agrupamiento indica que el procedimiento de elección del líder, y el intercambio de información dependen, en gran medida, de la hora de inicio de cada uno de los miembros del grupo y del tiempo de espera de las elecciones.

Un experimento trascendental con nuestras arquitecturas SDN es la organización del clúster. Los controladores SDN agrupados permiten limitar el número de dispositivos conectados en un dominio, la cantidad de datos y el dispositivo que se debe gestionar. La información de *clúster* intercambiada entre los controladores, se realiza a través de las herramientas Atomix, proporcionadas por los controladores de interconexión de Onos. Dentro de la plataforma de pruebas desarrollada, se pueden realizar simulaciones con números variables de 1 a 1,000 dispositivos conectados al mismo tiempo, en un entorno de IdC y, a su vez, pueden ser organizados de acuerdo a las necesidades de simular diversos tipos de pruebas haciendo uso de SDN. La adaptabilidad y la flexibilidad hacia diferentes entornos de desarrollo permiten implementar nuevas herramientas y tecnologías emergentes en la plataforma diseñada.

5. Conclusiones

En este artículo, presentamos las principales contribuciones de la virtualización de redes para el IdC. Con las pruebas realizadas, observamos un alto grado de escalabilidad, confiabilidad y alta tolerancia del SDN. La arquitectura ICDS desarrollada permite el control, la configuración y la gestión de redes complejas de forma dinámica y eficiente. Además, el resultado de la evaluación muestra un rendimiento superior a otras plataformas de pruebas desarrolladas incluyendo el desempeño de las funcionalidades del protocolo *Openflow*

Con base en los resultados obtenidos con la implementación y evaluación de nuestra arquitectura distribuida basada en SDN, se consideran nuevos temas de investigación por estudiar. Dentro de la plataforma se requiere un protocolo de gestión intra-cluster, a través de los controladores SDN. Con este enfoque, se puede proporcionar un mejor encaminamiento del flujo de datos para los objetos/dispositivos conectados en el clúster. Otro trabajo futuro para un despliegue escalable es incluir una combinación de protocolos IPv4 e IPv6 que proporcionen un rendimiento del concepto emergente de virtualización de redes.

6. Agradecimiento

Agradecimiento a la Secretaría Nacional de Ciencia y Tecnología e Innovación (SENACYT) por financiar la investigación, a través del programa del Sistema Nacional de Investigación SNI.

7. Referencias

[1] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, and M. Guizani, “A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security,” *CoRR*, vol. abs/1807.1, 2018.

[2] (i3 FORUM), “Internet of Things,” *Internet of Things*, 2018. .

[3] N. Bizanis and F. A. Kuipers, “SDN and Virtualization Solutions for the Internet of Things: A Survey,” *IEEE Access*, vol. 4, pp. 5591–5606, 2016.

[4] P. Bellavista, C. Giannelli, T. Lagkas, and P. Sarigiannidis, “Quality Management of Surveillance Multimedia Streams Via Federated SDN Controllers in Fiwi-Iot Integrated Deployment

Environments," *IEEE Access*, vol. 6, pp. 21324–21341, 2018.

[5] J. Seeger, A. Bröring, M. Pahl, and E. Sakic, "Rule-Based Translation of Application-Level QoS Constraints into SDN Configurations for the IoT," in *2019 European Conference on Networks and Communications (EuCNC)*, 2019, pp. 432–437.

[6] D. Sinh, L. Le, B. P. Lin, and L. Tung, "SDN/NFV — A new approach of deploying network infrastructure for IoT," in *2018 27th Wireless and Optical Communication Conference (WOCC)*, 2018, pp. 1–5.

[7] B. Lantz and B. O'Connor, "A Mininet-based Virtual Testbed for Distributed SDN Development," *SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, pp. 365–366, 2015.

[8] H. Kim, J. Kim, and Y. Ko, "Developing a cost-effective OpenFlow testbed for small-scale Software Defined Networking," in *16th International Conference on Advanced Communication Technology*, 2014, pp. 758–761.

[9] J. Schaerer, Z. Zhao, J. Carrera, S. Zumbrunn, and T. Braun, "SDNWisebed: A Software-Defined WSN Testbed," in *Ad-Hoc, Mobile, and Wireless Networks*, 2019, pp. 317–329.

[10] F. Yang, S. Zhang, S. Song, R. Li, Z. Zhao, and H. Zhang, "A Testbed for Intelligent Software Defined Security Framework," in *Proceedings of the ACM Turing Celebration Conference - China*, 2019, pp. 48:1--48:2.

[11] I. Alam *et al.*, "IoT Virtualization: {A} Survey of Software Definition & Function Virtualization Techniques for Internet of Things," *CoRR*, vol. abs/1902.1, 2019.

[12] and T. K. T. Barnett, S. Jain, U. Andra, "Cisco visual networking index (vni) complete forecast update," p. 38, 2019.

[13] C. J. Gonzalez Santamaria, "Management of a heterogeneous distributed architecture with the SDN," Universite de Reims Champagne Ardenne, 2017.

[14] P. Alto, "Software-Defined Networking: The New Norm for Networks [white paper]," *ONF White Paper*, 2012. .

[15] L. Atzori *et al.*, "SDN&NFV contribution to IoT objects virtualization," *Comput. Networks*, vol. 149, pp. 200–212, 2019.

[16] I. Miladinovic and S. Schefer-Wenzl, "A Highly Scalable IoT Architecture through Network Function Virtualization," *OJIOT*, vol. 3, pp. 127–135, 2017.

[17] Y. Jararweh, M. Al-Ayyoub, A. Darabseh, E. Benkhelifa, M. A. Vouk, and A. Rindos, "SDIoT: a software defined based internet of things framework," *J. Ambient Intell. Humaniz. Comput.*, vol. 6, pp. 453–461, 2015.

[18] L. E. Li, Z. M. Mao, and J. Rexford, "Toward Software-Defined Cellular Networks," in *2012 European Workshop on Software Defined Networking*, 2012, pp. 7–12.

[19] X. Jin, L. E. Li, L. Vanbever, and J. Rexford, "SoftCell: Scalable and Flexible Cellular Core Network Architecture," in *Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies*, 2013, pp. 163–174.

[20] P. Trakadas *et al.*, "Scalable monitoring for multiple virtualized infrastructures for 5G services," 2018, pp. 1–4.

[21] S. Mahamat Charfadine, O. Flauzac, F. Nolot, C. Rabat, and C. Gonzalez, "Secure Exchanges Activity in Function of Event Detection with the SDN," in *e-Infrastructure and e-Services for Developing Countries*, 2019, pp. 315–324.

[22] O. Salman, I. Elhajj, A. Chehab, and A. Kayssi, "IoT survey: An SDN and fog computing perspective," *Comput. Networks*, vol. 143, pp. 221–246, 2018.