

# SDN Implementation of virtual computing networks

Jhordy Esteban Salinas Santiago, Ing. Sistemas<sup>1</sup>, Carlos Andrés Sánchez Venegas, Ing. Sistemas, Juan Camilo Herrera Velásquez, Ing. Sistemas y Claudia P Santiago C, Msc. Gestión de Información <sup>1</sup>

<sup>1</sup> Escuela Colombiana de Ingeniería Julio Garavito, Colombia, jhordy.salina@mail.escuelaing.edu.co, claudia.santiago@escuelaing.edu.co

<sup>2</sup> Escuela Colombiana de Ingeniería Julio Garavito, Colombia, carlos.sanchez-v@mail.escuelaing.edu.co, juan.herrera@escuelaing.edu.

*Abstract -- At this moment is difficult to think in a world without internet, that is why it is important to guarantee the correct and efficient network operation and the easy administration. On the other hand, it is important to use new technologies used by companies such as Google and Facebook that improve the operation of the networks. One of this technologies is Software Defined network – SDN, which allow centralizing the administration of the network and virtualizing its operation. This paper presents the SDN technology, its advantages, implementation mechanisms and a group of application than can be used to create and configure it.*

*Keywords– SDN, Virtual networks, software defined networks.*

Digital Object Identifier (DOI):  
<http://dx.doi.org/10.18687/LACCEI2019.1.1.478>  
ISBN: 978-0-9993443-6-1 ISSN: 2414-6390

# SDN Implementación de redes de cómputo virtuales

Jhordy Esteban Salinas Santiago, Ing. Sistemas<sup>1</sup>, Carlos Andrés Sánchez Venegas, Ing. Sistemas, Juan Camilo Herrera Velásquez, Ing. Sistemas y Claudia P Santiago C, Msc. Gestión de Información <sup>1</sup>

<sup>1</sup> Escuela Colombiana de Ingeniería Julio Garavito, Colombia, jhordy.salina@mail.escuelaing.edu.co, claudia.santiago@escuelaing.edu.co

<sup>2</sup> Escuela Colombiana de Ingeniería Julio Garavito, Colombia, carlos.sanchez-v@mail.escuelaing.edu.co, juan.herrera@escuelaing.edu.co

*Abstract— At this moment is difficult to think in a world without internet, that is why it is important to guarantee the correct and efficient network operation and the easy administration. On the other hand, it is important to use new technologies used by companies such as Google and Facebook that improve the operation of the networks. One of this technologies is Software Defined network – SDN, which allow centralizing the administration of the network and virtualizing its operation. This paper presents the SDN technology, its advantages, implementation mechanisms and a group of application than can be used to create and configure it.*

*Keywords—SDN, Virtual networks, software defined networks.*

*Resumen- En este momento es muy difícil pensar en una empresa sin conexión a internet, eso hace indispensable que las redes y la infraestructura sean una parte importante del día a día de una organización, por esta razón es indispensable dichas redes sean fácilmente configurables, rápidas, seguras. Es necesario que a medida que las nuevas tecnologías van siendo aceptadas por el mercado, las redes de igual manera se “actualicen” y mejoren la operación para suplir todas las necesidades requeridas por los usuarios finales. Esto lleva a las redes definidas por software – SDN, una nueva tecnología que está tomando fuerza y está siendo implantada por empresas como Google y Facebook. Las SDN tienen una nueva arquitectura basada en la centralización y virtualización de las redes para facilitar su administración. En el presente artículo se presenta de manera general las SDN, su arquitectura, funcionamiento y requerimientos en hardware y software. Se plantean ejemplos útiles que facilitan ser aplicados en redes empresariales. Adicionalmente, se estudian y configuran varias aplicaciones de software que pueden ser usadas en su implementación.*

*Palabras clave- SDN, Redes virtuales. Redes definidas por software.*

## I. INTRODUCCIÓN

Las redes actuales hacen parte de la infraestructura básica de las organizaciones ya que todos los equipos (computadores, servidores, etc.) requieren estar conectados y es a través de las redes que logran dicha interconexión. Estas redes están en muchos casos compuestas por una gran cantidad de dispositivos de interconexión (switches y routers) físicos que implican para el administrador, y las empresas, un trabajo de gestión, actualización (que puede ser muy costoso) y hasta tediosa dado que cada equipo de red se encuentra separado y requiere una administración individual, de modo que para

Digital Object Identifier (DOI):

<http://dx.doi.org/10.18687/LACCEI2019.1.1.478>

ISBN: 978-0-9993443-6-1 ISSN: 2414-6390

17<sup>th</sup> LACCEI International Multi-Conference for Engineering, Education, and Technology: “Industry, Innovation, And Infrastructure for Sustainable Cities and Communities”, 24-26 July 2019, Jamaica.

realizar una modificación es necesario ir equipo por equipo. Todo esto implica incurrir en costos que pueden llegar a ser elevados y tiempos de configuración y mantenimiento para el personal de TI que pueden convertirse en una porción alta de su trabajo.

Software Defined Network – SDN es una nueva tecnología, la cual le da una nueva visión a la administración de las redes de cómputo, de la tradicional forma distribuida de hacerlo hacia una nueva manera centralizada. Esta implementación permite tener redes más flexibles, adaptables, expandibles y también llegar a reducir significativamente los costos de implementación y de operación.

En el presente artículo, se parte de una investigación sobre qué es SDN, cómo están configuradas las redes definidas por software, cuál es su arquitectura y cuáles son sus beneficios como lo puede ser la escalabilidad, centralización, agilidad, funcionamiento y bajo costo. Para luego presentar la implementación de una red SDN en un ambiente virtual de una infraestructura empresarial y poder ver cómo es su operación.

## II. JUSTIFICACIÓN

Las redes han venido creciendo exponencialmente y los usuarios cada vez requieren mayor velocidad, rendimiento y confiabilidad, para hacer posible el uso de diferentes servicios proporcionados por Internet. En un principio no se esperaba la masificación de dispositivos conectados entre sí, es decir, que se tenían en cuenta las principales primitivas para su escalabilidad, pero nunca se imaginaron la magnitud del requerimiento de flujo de datos sobre la red (“Desde 1984 a 2012: 1.2 zettabytes generados, Pronóstico al 2017 fue de: 1.4 zettabytes” [2]). Proporcionando una base, pero también dejando un gran campo de acción para su mejora y evolución [1].

A finales del siglo XX e inicios del siglo XXI las organizaciones y las personas sintieron la necesidad de interconectarse con el fin de compartir información en la búsqueda de la generación de un conocimiento globalizado, a tal punto de requerir nuevas tecnologías y recursos para abastecer sus requerimientos [2]. A lo largo de la historia se ha venido trabajando sobre las bases que se pautaron inicialmente (protocolos, maquinaria, arquitecturas, etc.) mejorando poco a poco dichas bases, pero nunca generando grandes cambios

como lo podría ser el cambio de la arquitectura que actualmente se maneja.

Hoy en día la administración de las redes es complicada y tediosa por diferentes motivos, bien sea por falta de documentación, desconocimiento de su topología o porque simplemente se están usando prácticas discontinuadas [3], a la hora de construir o manipular una red se encuentran escenarios en donde todos los elementos están distribuidos, esto hace que su configuración se tenga que hacer de manera independiente en cada uno de los elementos que la componen pero se debe lograr una unión integral para su correcto funcionamiento [4].

En la universidad de Stanford en el año 2008 surgió una nueva tecnología de red emergente llamada SDN (Software Defined Network) en donde se separa el plano de control del plano de infraestructura. Esta tecnología ofrece la oportunidad de tener control sobre la red de manera centralizada, reduciendo la complejidad y costo de las operaciones que ésta requiere para su mantenimiento y funcionamiento [5]. La implementación de SDN en una red también ofrece más beneficios, permite el manejo dinámico de la misma permitiendo una mejor experiencia para el usuario final, además, brinda acceso a sus servicios de manera rápida y segura [6].

SDN no se encuentra ligado a ningún protocolo o tecnología, por ejemplo, se puede realizar la implementación de una red típica por medio del protocolo de internet IPv4 y también implementarlo con el protocolo IPv6 [7] generando una red moderna, sin cambios o configuraciones adicionales [8]. Esto le permite a SDN ser una arquitectura flexible y de fácil implementación. Por ejemplo, al crear una infraestructura SDN con IPv6, permitirá tener una red rápida, dinámica, escalable, económica, centralizada y segura abasteciendo las necesidades de los usuarios que tendrán la oportunidad de conectarse a esta red [9].

Por medio de la implementación de nuevas tecnologías como lo es SDN se pueden crear “redes modernas” en donde se pueden obtener varios beneficios, entre ellos, una red escalable, segura, expansible, dinámica, económica e innovadora

### III. SDN

Redes Definidas por Software (Software Defined Network - SDN) es una tecnología que surge en la universidad de Stanford a principios del año 2008 y que tiempo después (año 2011) fue apoyada por empresas como Deutsche Telekom, Facebook, Google, Microsoft, Verizon y Yahoo! con la creación de la ONF (Open Networking Foundation) para promover y adaptar redes SDN proponiendo el uso del protocolo OpenFlow[10], esta tecnología separa el control de la operación de red del proceso de reenvío de paquetes a

través de la misma, es decir, SDN hace la separación del plano de control del de datos (ver figura 1), con el propósito de centralizar la red por medio de un controlador que ahora se encargará de la toma de decisión sobre el reenvío de paquetes y que se encuentra en la capa de control (p.ej. Con el controlador OpenDaylight, del cual se hablará más adelante en el artículo) haciendo que la red funcione de manera dinámica y escalable, así mismo SDN proporciona velocidad y agilidad al momento de implementar aplicaciones y servicios empresariales.

#### A. Arquitectura

Esta tecnología tiene una arquitectura que se divide en 3 capas las cuales se presentan a continuación:

##### 1) Aplicación

En esta capa se encuentran las aplicaciones de negocio, estas aplicaciones de negocio son programas que conocen los requerimientos que existen en la red, que son usadas por usuarios o software [11] y trabajan de la mano con el controlador para cubrir completamente estos requerimientos. La función principal de la capa de aplicación en una arquitectura SDN es comunicar los requerimientos a la capa de control, así mismo, estas aplicaciones reciben información por parte de la capa de control para saber cómo está funcionando la red (hay comunicación bidireccional) y además le permite la construcción de una vista abstracta de la misma red para la correcta toma de decisiones [12]. Básicamente, desde la capa de aplicación se puede administrar toda la infraestructura y decirle a ella qué es lo que tiene que hacer, todo esto por medio de las aplicaciones de negocio.

##### 2) Control

Esta capa cuenta con un elemento fundamental que es el controlador, es el corazón de la arquitectura SDN ya que por medio de éste se gestionan las otras dos capas (aplicación e infraestructura) por medio de dos interfaces, hacia la capa de aplicación por medio de APIs y hacia la capa de infraestructura por medio de protocolos como lo es OpenFlow. El controlador primeramente recibe toda la información de las aplicaciones de negocio y dada las aplicaciones realiza cambios sobre la infraestructura, posteriormente la infraestructura le da información para que el controlador le comunique dicha información a la capa de aplicación y se realice una muy buena gestión y administración de la red.

##### 3) Infraestructura

La capa de infraestructura es la parte física de una arquitectura SDN, aquí se encuentra el hardware (el cual debe

soportar el protocolo OpenFlow) que realiza las tareas dadas por la capa de aplicación y traducidas por la capa de control.

Este componente se encarga del reenvío de paquetes a través de la red teniendo en cuenta toda la información recogida en las capas anteriores.

### B. Forma de operación

SDN tiene muchas funcionalidades, por ejemplo, suponga que se encuentra en una empresa de desarrollo de software en donde todos los viernes se realiza un backup de una aplicación que se está desarrollando. El backup se envía desde el servidor en donde se encuentra la aplicación hasta otro servidor en donde se tiene un Document Management System, encargado de controlar todo el versionamiento del backup de dicha aplicación. Los desarrolladores consideran que el backup tarda mucho tiempo y les está haciendo perder tiempo. SDN es la solución a esta problemática, Se puede crear una aplicación de negocio encargada de gestionar todo el proceso de backup de la red, esta aplicación es definida de modo que realice una priorización de envío de paquetes, por ejemplo, creando un “canal” exclusivo en la red en donde únicamente permita por ese canal el tráfico del servidor que está realizando el backup para que esto se haga de manera mucho más rápida. Esta aplicación (que se encuentra en la capa de aplicación) es conectada con la capa de control por medio de una interfaz que es llamada application-controller plane interface (A-CPI)[13]. Una vez el controlador sepa que la red se tiene que comportar de cierta manera los viernes, le informa a su infraestructura por medio del protocolo OpenFlow que se comporte de la manera previamente descrita, de modo que el servidor que va a realizar backup tendría un canal priorizado, realizando la tarea mucho más rápido, mejorando el rendimiento de la red, haciendo que los desarrolladores puedan trabajar en su proyecto sin distracción o interrupciones.

### C. Software para la capa control

SDN es una nueva tecnología, a pesar de esto, ya hay muchas empresas y proyectos enfocados en el desarrollo de controladores SDN, a continuación, se mostrarán 3 controladores SDN que son los más usados actualmente.

#### 1) OpenDayLight

Es un proyecto de código abierto, patrocinado por la fundación Linux, con el objetivo de promover la creación de redes definidas por software (SDN) y la virtualización de las funciones de dicha red, es un software escrito en el lenguaje de programación Java y usado como el Controlador SDN que es el cerebro de la arquitectura SDN. El controlador OpenDayLight expone las APIs hacia la capa superior

(Northbound API), es decir, hacia la capa de aplicación, que son utilizadas por las aplicaciones SDN. Estas aplicaciones utilizan el controlador para recopilar información sobre la red, ejecutar algoritmos que realizan analítica y así tener el poder para, a continuación, usar este controlador con el objetivo de crear nuevas reglas en toda la red.

#### 2) Open Network Operating System

Open Network Operating System, o más conocido por sus siglas como ONOS, es un software lanzado el 5 de septiembre de 2014 escrito en java y que proporciona una plataforma distribuida de plataformas SDN sobre un contenedor Apache. Se maneja como una vista de nodos los cuales pueden soportar fallas sin causar problemas en la operación de la red. Como todo controlador SDN, éste proporciona a las aplicaciones SDN una cantidad significativa de abstracciones a alto nivel.

ONOS es una mezcla de modelos de gobierno de código abierto, el consejo de ONOS provee liderazgo estratégico y de pensamiento. El primer lanzamiento de ONOS se llamó Avocet, y el siguiente a éste fue Blackbird, que fue lanzado recientemente centrándose en optimizaciones rendimiento y la definición de métricas “carrier-grade quotient” [14].

ONOS provee un Shell en donde se puede administrar el controlador por medio de comandos y en dado caso que sea muy complicado de administrar (o poco amigable para el usuario), ONOS también provee una interfaz web desde donde se puede administrar todas las aplicaciones de negocio (agregar, eliminar, activar, pausar, detener) y hasta ver toda la topología de la red implementada por medio de nodos y enlaces.

#### 3) Floodlight

Es otro SDN controller que es open source, tiene características como: trabajar con dispositivos físicos y virtuales que hablen el protocolo OpenFlow, corre sobre servidor apache, lo cual le permite ser usado para casi cualquier propósito, es un controlador muy fácil de usar, por lo simple de construir y ejecutar, los miembros de la comunidad son desarrolladores que lo prueban y mejoran constantemente [15].

El controlador Floodlight fue originalmente ofrecida para grandes switch como parte del proyecto OpenDaylight [16].

Como se ha mostrado, hay varios controladores SDN que ofrecen servicios similares. Ahora, para el desarrollo de esta investigación, se decidió continuar con el controlador ONOS, dado que el controlador cuenta con más prestaciones y tiene una interfaz gráfica más fácil de usar, más intuitiva y que permitía acceder a más opciones de configuración y en consecuencia una mejor experiencia de usuario.

#### D. Software para la capa infraestructura

En la capa de infraestructura es donde se encuentran todos los dispositivos físicos en una infraestructura (routers, switches, computadores, etc.). Para poder implementar una red SDN es necesario que el hardware de la infraestructura tenga la tecnología adecuada, es decir, deben soportar el protocolo OpenFlow en los equipos no finales de la red (comúnmente, switches y routers) para que el controlador pueda tener comunicación con dichos equipos e indicarle el funcionamiento según las aplicaciones de negocio.

Para poder probar, implementar y conocer una red SDN no es necesario contar con el hardware requerido, hay un software llamado Mininet el cual permite la virtualización de una infraestructura por medio de un script en Python, la cual soporta una red definida por software. Mininet es muy útil para poder realizar pruebas, conocer más de las redes definidas por software dado que la simulación de la red se puede realizar con un controlador real (montado en otro servidor). Además, de permitir realizar funciones sobre la red como, por ejemplo, realizar ping entre equipos.

A continuación, se presenta una descripción más a fondo el protocolo OpenFlow y el software para la implementación virtual llamado Mininet.

##### 1) OpenFlow

El protocolo OpenFlow, es considerado el primer estándar SDN que da inicio en la universidad de Standford en el año 2008 (junto con el concepto de las redes definidas por software) pero no es sino hasta el mes de diciembre del siguiente año que es lanzada la versión 1.0 de este protocolo [17] y que en la actualidad es manejada por la ONF (Open Networking Foundation) la cual es una organización fundada por grandes empresas como Google, Facebook y Microsoft, enfocada en la promoción de redes por medio de las redes definidas por software.

##### 2) Mininet

Mininet es un emulador de red de código abierto que es de fácil uso y manejo gracias a que admite su instalación en una máquina virtual con sistema operativo Linux y que permite la creación de una red por medio de scripts programados en Python. Dichos scripts contienen hosts virtuales, conmutadores, enlaces y controladores de muy alta flexibilidad y sencillez dado que de base soporta protocolos en los equipos como OpenFlow, al igual que arquitecturas de red como lo es SDN. Además, cuenta con un CLI (interfaz de línea de comandos) que es amigable [18]. Para crear una red virtual con Mininet es muy fácil, simplemente hay que realizar un script en Python en donde se importen las librerías de Mininet

y se defina la red por medio de instrucciones definidas mediante una programación orientada a objetos muy básica.

## IV. ANÁLISIS Y DESARROLLOS

A continuación, se mostrará la instalación y configuración de una red empresarial en un ambiente virtual con una arquitectura SDN en donde se tendrá un servidor totalmente dedicado a hacer la función de controlador y otro servidor simulando la red empresarial.

Para la virtualización se crearon objetos de diferentes tipos, como computadores, celulares y switches, lo cual permite tener una topología de red empresarial de manera virtual y del tamaño que se requiera.

### A. Selección de herramientas

Para el desarrollo de esta etapa de la investigación, se seleccionaron herramientas para cada una de las capas definidas en la arquitectura SDN. Para la capa de control, se seleccionó ONOS y para la capa de infraestructura se seleccionó la implementación de una red virtual por medio de Mininet.

- ONOS: Se seleccionó esta aplicación para ser el controlador de la red, de modo que este se encuentra en la capa de control. ONOS (figura 1) fue seleccionado por considerar que es uno de los controladores más usados debido a que provee una experiencia de usuario muy buena, es de fácil instalación y configuración. ONOS al ser un proyecto de software libre, se tiene fácil acceso a sus fuentes de manera gratuita y su implementación está perfectamente documentada, adicionalmente sus API's son abiertas para poder usar la conexión con las aplicaciones SDN que son las que definirán las reglas de negocio, además de esto brinda una interfaz gráfica bastante amigable en donde se puede ver todas las aplicaciones SDN y activarlas o desactivarlas con un clic.



- La virtualización de la red se realizó haciendo uso de Mininet.
- Estos dos software se encuentran en máquinas virtuales creadas por medio de VirtualBox con sistema operativo Ubuntu server.

### B. Montaje

## 1) Instalación de herramientas

Lo primero es construir el ambiente de trabajo, así que lo que se debe hacer es descargar VirtualBox (el cual permite tener máquinas virtuales en un equipo) e instalarlo. Para la infraestructura se necesita tener un servidor con el Mininet instalado, de modo que es necesario dirigirse a la página de Mininet para poder descargar su software ([www.mininet.org/download](http://www.mininet.org/download)).

Al momento de la instalación se presentan 4 formas para instalar Mininet en la máquina virtual, estas son:

- Por medio de una máquina virtual ya preinstalada y configurada, lo que significa que ya se encuentra instalado Mininet en un servidor, sólo se necesita descargar el servidor, crear la red y asignarla al controlador.
- Una instalación nativa descargado el repositorio de Github, lo que implica una instalación y configuración desde código fuente.
- Una instalación por medio de paquetes, lo que implica descargar el software ya compilado para su posterior instalación.
- Una actualización de un Mininet ya existente en la máquina virtual, lo que significa que se debe de tener instalado ya Mininet en el computador o servidor.

En este caso, se hizo una instalación nativa desde el recurso porque facilitaba controlar lo que se instalaría y daba mayor claridad sobre lo que se requería versus lo que se instalaba, Para esto, se configuró una máquina virtual, con un servidor Ubuntu 14.04 como se ve en la figura 2.

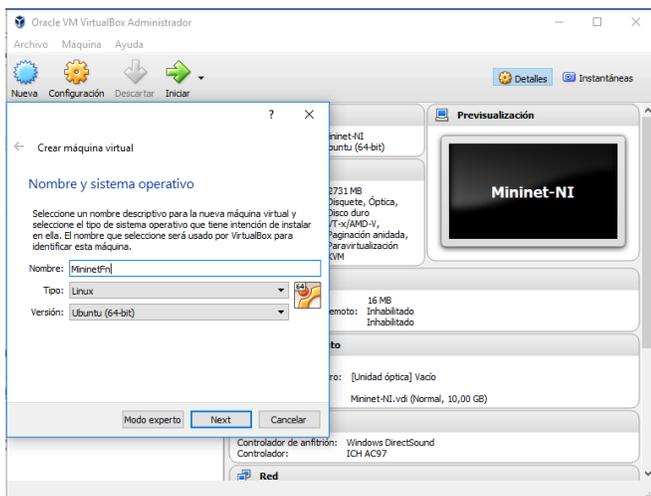


Figura 2. Máquina virtual de Mininet

Se configuró la máquina virtual, indicando que la interfaz de red estuviera configurada, en el administrador de virtualización, como “puente”/Bridge para que dicha máquina

virtual se pudiera conectar (hacer ping) con las otras máquinas virtuales que se tuvieran activas en la simulación, es decir, para que la máquina virtual que tiene la red simulada tuviera conexión con la máquina virtual que contiene el controlador.

## 2) Configuración

Posteriormente se procede a configurar el software instalado en la máquina virtual que contiene el controlador, es decir, que contiene a ONOS. Para esto se ejecuta ONOS con el comando que se ve en la figura 3, el cual es el encargado de prender el servicio de ONOS, luego de esto se procede a conectar la red al controlador por medio del protocolo OpenFlow.

```
onos@onos:/$ /opt/onos/bin/onos start
```

Figura3. Comando para iniciar ONOS, de este modo podremos acceder de manera web o CLI.

Finalizado estos pasos, la infraestructura está completamente configurada y lista para ser usada en el diseño, gestión y operación de una SDN. El controlador ya se encuentra en servicio y solo faltaría asignarle la red al controlador.

## C. Implemetación

Con el software ya listo, ahora se procedió a diseñar, construir y probar las SDN. Para esto se va a verificar que se pueda asignar correctamente la infraestructura a ONOS.

Después de iniciar el servidor es muy útil ver si servidor está escuchando por los puertos requeridos para conectarse con la infraestructura por medio del protocolo OpenFlow (en este caso, los puertos 6653 y 6633) y el puerto web para mostrar su interfaz gráfica (8181) con el comando “netstat -putona” el cual muestra los puertos en donde estpa escuchando o en donde hay una conexión establecida, esto se puede apreciar en la figura 4.

```

ONOS2.0 [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

(No se pudo leer la información para "p": geteuid()=1000 pero usted debe ser superusuario.)
Conexiones activas de Internet (servidores y establecidos)
Proto Recib Envíat Dirección local Dirección remota Estado PID/Programa name Ten
porizador
tcp 0 0 0.0.0.0:22 0.0.0.0:* ESCUCHAR - apa
jdo (0.0.0.0) 0 :::9076 :::* ESCUCHAR - apa
tcp 0 0 :::36212 :::* ESCUCHAR - apa
jdo (0.0.0.0) 0 :::8181 :::* ESCUCHAR - apa
tcp 0 0 :::11990 :::* ESCUCHAR - apa
jdo (0.0.0.0) 0 :::22 :::* ESCUCHAR - apa
tcp 0 0 :::6653 :::* ESCUCHAR - apa
jdo (0.0.0.0) 0 127.0.0.1:41346 :::* ESCUCHAR - apa
jdo (0.0.0.0) 0 :::8101 :::* ESCUCHAR - apa
tcp 0 0 :::6633 :::* ESCUCHAR - apa
jdo (0.0.0.0) 0 0.0.0.0:44544 0.0.0.0:* - apa
udp 0 0 0.0.0.0:65036 0.0.0.0:* - apa
jdo (0.0.0.0) 0 0.0.0.0:68 0.0.0.0:* - apa
jdo (0.0.0.0) 0 0.0.0.0:68 0.0.0.0:* - apa
udp 0 0 :::3196 :::* - apa
jdo (0.0.0.0) 0 :::65036 :::* - apa
jdo (0.0.0.0) 0 0.0.0.0:68 0.0.0.0:* - apa
muestros:~$ netstat -gntona_

```

Figura 4. Comando netstat para ver los puertos por el cual el servidor está escuchando o tiene conexión.

Luego se procedió a realizar un script en Python (ver figura 5) por medio de Mininet en donde se simulará una red empresarial mínima, contará con un computador como host, un servidor y dos switches conectados entre ellos, de allí se pueden formar redes más complejas, por ejemplo, al agregar más equipos finales en cada uno de los switches, mas switches en la red y más conexiones entre ellos, etc.. El script es una clase la cual implementa la clase Topo, en el creador se definen todos los hosts (con el método addHost), todos los switches (con el método addSwitch) y todas las relaciones (con el método addLink).

```

from mininet.topo import Topo

class PgrTopo( Topo ):
    "PGR1-SDN Topology"
    def __init__( self ):
        "Create custom topo"
        # Initialize topology
        Topo.__init__( self )

        # Add hosts
        PC0 = self.addHost('PC0')

        # Add servers
        Server0 = self.addHost('Server0')

        # Add switches
        Switch0 = self.addSwitch('Switch0')
        Switch1 = self.addSwitch('Switch1')

        # Add links
        ##Between Switchs
        self.addLink(Switch1,Swicth0)
        ##Between Switch-Server
        self.addLink(Server0,Swicth1)
        ##Between Switch-Host
        ###Switch1
        self.addLink(PC0,Swicth1)

topos = { 'PgrTopo': ( lambda: PgrTopo() ) }

```

Figura 5. Script en Python con la topología de una red mínima empresarial.

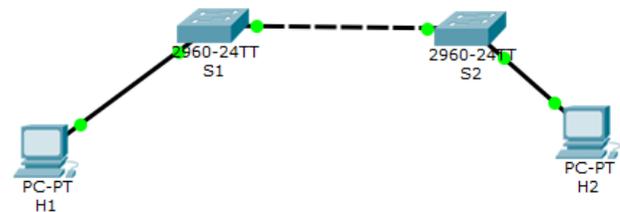


Figura 6. Red gráfica de ejemplo que simula la red creada con el script de la figura 5. Creada por medio de PacketTracer

Después de crear la topología de red, se procede a indicarle a MININET que ONOS será su controlador SDN para que éste lo reconozca y ejecute las acciones indicadas sobre la red, acciones que fueron creadas en la capa de aplicación y también se usaron las que provee el controlador, como se puede apreciar en la figura 7. Cabe resaltar que el comando mostrado en la figura 7 genera una red dado el script y asigna direcciones IPv4 de manera aleatoria en una red 10.0.0.0/8.

```

mininet@mininet:~$ sudo mn --custom topology/Complex.py --topo mytopo --mac --switch ovs,protocols=OpenFlow13 --controller=remote,ip=192.168.56.101,port=6633

```

Figura 7. Creación de la red y asignación del controlador ONOS a la misma

Por medio de la interfaz web se puede ver la capa de aplicación (las aplicaciones de negocio). ONOS trae preinstaladas muchas aplicaciones, como por ejemplo la que permite el reenvío de paquetes (forwarding) o la aplicación que permite la conexión con el controlador. Estas aplicaciones pueden ser creadas, eliminadas o pausadas. La aplicación relacionada con el reenvío de paquetes no viene activada por defecto, de modo que la red no podrá tener conexión, es decir, ni siquiera se podrá realizar ping entre los equipos de la red como se puede observar en la Figura 8 hasta que se active la aplicación de forwarding.

```
mininet> pingall
*** Ping: testing ping reachability
PC0 -> X X X X X X X X X X X X X X X X X X X X X X
PC1 -> X X X X X X X X X X X X X X X X X X X X X X
PC2 -> X X X X X X X X X X X X ^C
Interrupt
```

Figura 8. Ping entre todos los equipos en la consola de Mininet sin éxito debido a que no está activada la aplicación de forwarding en el controlador

Para poder activar dicha aplicación de manera gráfica (web), basta con dirigirse a la pestaña de aplicaciones, buscar la relacionada con reenvío de paquetes y activarla (ver figura 9).

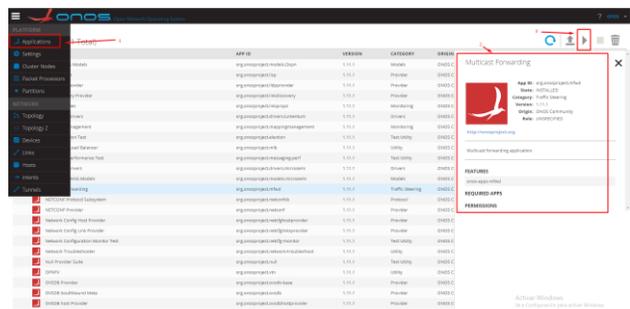


Figura 9. Inicio de la aplicación relacionada con el reenvío de paquetes de manera grafica

Al activar la aplicación de forwarding, la cual, como se había mencionado previamente permite el reenvío de paquetes para que toda la red pueda tener una comunicación, se puede establecer la conexión entre todos los equipos de la red como se ve en la figura 10. También se puede ver que ahora ONOS reconoce los equipos presentes en la red, y en la pestaña de topología se pueden ver los equipos conectados a cada switch (ver figura 11).

```
mininet> pingall
*** Pings: testing ping reachability
Des -> Inv Laptop1 Laptop2 Laptop3 Laptop4 PC1 PC2 PC3 PC4 PC5 PC6 Phone1 Phone2 Phone3 Phone4 Fruebas
Inv -> Des Laptop1 Laptop2 Laptop3 Laptop4 PC1 PC2 PC3 PC4 PC5 PC6 Phone1 Phone2 Phone3 Phone4 Fruebas
Laptop1 -> Des Inv Laptop2 Laptop3 Laptop4 PC1 PC2 PC3 PC4 PC5 PC6 Phone1 Phone2 Phone3 Phone4 Fruebas
Laptop2 -> Des Inv Laptop1 Laptop3 Laptop4 PC1 PC2 PC3 PC4 PC5 PC6 Phone1 Phone2 Phone3 Phone4 Fruebas
Laptop3 -> Des Inv Laptop1 Laptop2 Laptop4 PC1 PC2 PC3 PC4 PC5 PC6 Phone1 Phone2 Phone3 Phone4 Fruebas
Laptop4 -> Des Inv Laptop1 Laptop2 Laptop3 PC1 PC2 PC3 PC4 PC5 PC6 Phone1 Phone2 Phone3 Phone4 Fruebas
PC1 -> Des Inv Laptop1 Laptop2 Laptop3 Laptop4 PC2 PC3 PC4 PC5 PC6 Phone1 Phone2 Phone3 Phone4 Fruebas
PC2 -> Des Inv Laptop1 Laptop2 Laptop3 Laptop4 PC1 PC3 PC4 PC5 PC6 Phone1 Phone2 Phone3 Phone4 Fruebas
PC3 -> Des Inv Laptop1 Laptop2 Laptop3 Laptop4 PC1 PC2 PC4 PC5 PC6 Phone1 Phone2 Phone3 Phone4 Fruebas
PC4 -> Des Inv Laptop1 Laptop2 Laptop3 Laptop4 PC1 PC2 PC3 PC5 PC6 Phone1 Phone2 Phone3 Phone4 Fruebas
PC5 -> Des Inv Laptop1 Laptop2 Laptop3 Laptop4 PC1 PC2 PC3 PC4 PC6 Phone1 Phone2 Phone3 Phone4 Fruebas
PC6 -> Des Inv Laptop1 Laptop2 Laptop3 Laptop4 PC1 PC2 PC3 PC4 PC5 Phone1 Phone2 Phone3 Phone4 Fruebas
Phone1 -> Des Inv Laptop1 Laptop2 Laptop3 Laptop4 PC1 PC2 PC3 PC4 PC5 PC6 Phone2 Phone3 Phone4 Fruebas
Phone2 -> Des Inv Laptop1 Laptop2 Laptop3 Laptop4 PC1 PC2 PC3 PC4 PC5 PC6 Phone1 Phone3 Phone4 Fruebas
Phone3 -> Des Inv Laptop1 Laptop2 Laptop3 Laptop4 PC1 PC2 PC3 PC4 PC5 PC6 Phone1 Phone2 Phone4 Fruebas
Phone4 -> Des Inv Laptop1 Laptop2 Laptop3 Laptop4 PC1 PC2 PC3 PC4 PC5 PC6 Phone1 Phone2 Phone3 Fruebas
Fruebas -> Des Inv Laptop1 Laptop2 Laptop3 Laptop4 PC1 PC2 PC3 PC4 PC5 PC6 Phone1 Phone2 Phone3 Phone4
*** Results: 0% dropped (372/372 received)
mininet>
```

Figura 10. Comando ping de nuevo en consola de Mininet, ahora que se tiene la aplicación encargada del reenvío de paquetes, los computadores tienen con éxito comunicación.

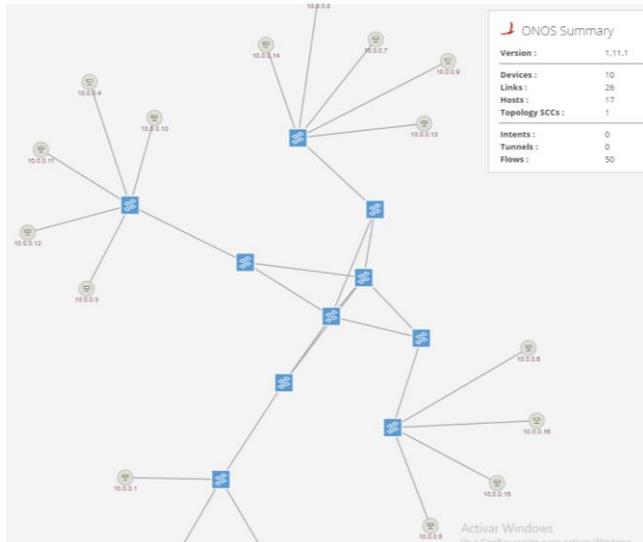


Figura 11. Topología de red después de correr el comando ping para descubrir todos los equipos de la red.

Ahora que la red definida por software ya tiene la capacidad de permitirle comunicación a los equipos que se encuentren dentro de ella, el siguiente paso es administrarla por medio del controlador ONOS, por ejemplo, si se quiere realizar lo planteado en la sección B del capítulo III bastaría con crear una aplicación de negocio específica para esa tarea y activarla en el controlador como se hizo con la aplicación de forwarding.

#### D. Resultado de las implementación

Se mostró la instalación y configuración de una red empresarial virtual implementada con una arquitectura SDN en donde se podría administrar toda la red por medio de aplicaciones SDN en un controlador ONOS.

Se creó una red virtual por medio de Mininet y Python en donde se simuló una red de negocio o trabajo típica, en donde se intentó establecer comunicación entre los equipos, inicialmente sin éxito, se solucionó dicho problema por medio de las aplicaciones de negocio que provee ONOS por defecto y se mostró los resultados de la comunicación establecida con éxito entre los equipos y como ONOS puede ver dicha relación de manera global.

Ahora que toda la red se encuentra con arquitectura SDN el siguiente paso sería crear las aplicaciones de negocio encargadas de administrar y gestionar toda la red.

## V. TRABAJO FUTURO

Dentro del trabajo realizado en este proyecto, no se abordó la configuración y uso de las LAN virtuales o VLANS. Ya que es muy común que se tengan redes virtuales de área locales (VLANS) en las organizaciones, por ejemplo, para una organización de desarrollo se podrían tener 3 redes virtuales, las cuales serían: la de desarrollo, pruebas y marketing. Sería muy útil poderlas implementar y administrar mediante la tecnología SDN. El hecho de poder administrar estas VLANS de manera centralizada y muy probablemente por medio de una interfaz gráfica web, es una funcionalidad muy útil que las organizaciones aprovecharían, pudiendo ahorrar en tiempo y costos de configuración y gestión.

La seguridad siempre es un aspecto fundamental en cualquier infraestructura y dado que con las redes definidas por software toda la administración se encuentra centralizada, el controlador se convierte en un recurso crítico para la infraestructura. Es fundamental revisar cuidadosamente todos los puntos críticos que conlleva una implementación SDN y se propone como trabajo futuro.

## VI. CONCLUSIONES

La administración de las redes actuales puede llegar a ser compleja y hasta tediosa, con una tecnología como SDN se puede convertir en una tarea más simple, rápida y económica dando un espacio para la innovación y el progreso.

Para la implementación de SDN se requiere una serie de productos de software y hardware diseñados para soportar esta tecnología, es fundamental que el hardware cuente con el protocolo OpenFlow para permitir la comunicación entre equipos y controlador.

La administración de la red centralizada es muy buena, pero esto involucra temas de seguridad muy grandes dado que desde el controlador podría afectar toda la red, por ejemplo, generando problemas de disponibilidad.

ONOS es un controlador muy bueno que actualmente está siendo muy utilizado, que provee una interfaz de usuario muy amigable y que permite tener una escalabilidad en la capa de aplicación muy amplia, siendo un controlador muy bueno para educación y para puesta en producción en un ambiente real de trabajo. ONOS permite administrar las aplicaciones de negocio de manera muy fácil, como se mostró con la aplicación de reenvío de paquetes.

Mininet es una herramienta muy útil a la hora de realizar pruebas, dado que, si no se tienen los equipos físicos, basta con realizar un script en Python que los cree.

SDN está tomando mucha fuerza actualmente y ya está siendo implementado por grandes empresas debido a la manera como replantea toda la administración de las redes, lo cual es innovador y seguramente será el futuro de la infraestructura de redes. Las redes definidas por software con el apoyo de nuevas tecnologías o protocolos como lo es IPv6 crearán una nueva generación de “redes modernas”.

## REFERENCIAS

- [1] C. Holloway, “Las redes al 2017: el desenfadado crecimiento de la información que generamos,” TecnoAmerica, 2013. [Online]. Available: <https://tecno.americaeconomia.com/articulos/las-redes-al-2017-el-desenfrenado-crecimiento-de-la-informacion-que-generamos>. [Accessed: 23-Aug-2018].
- [2] M. Luque, “La influencia de internet en la sociedad actual,” SoloCiencia, 2015. [Online]. Available: <http://www.solociencia.com/informatica/influencia-internet-sociedad-actual-origen-evolucion-historica.htm>. [Accessed: 23-Aug-2018].
- [3] M. Vázquez, “Las redes informáticas y el conocimiento sobre el fenómeno urbano,” Educational. pp. 4–12, 2000.
- [4] SearchDataCenter en Español, “Redes empresariales: Todo lo que necesita saber,” SearchDataCenter en Español, 2015. [Online]. Available: <http://searchdatacenter.techtarget.com/es/guia/Redes-empresariales-Todo-lo-que-necesita-saber>. [Accessed: 23-Aug-2018].
- [5] Open Networking Foundation, “SDN Architecture Overview,” Onf, pp. 1–5, 2013.
- [6] I. Networks Inc, “Introducción SDN,” Iquall Networks. Iquall Networks Inc, pp. 1–4, 2016.
- [7] Wikispaces, “Surgimiento de IPv6,” Wikispaces, 2017. [Online]. Available: <https://ipv6nuevastecredes.wikispaces.com/2.+SURGIMIENTO+DE+IPV6>. [Accessed: 23-nov-2018].
- [8] J. Morgan, “A Simple Explanation Of ‘The Internet Of Things,’” Forbes, 2014. [Online]. Available: <https://www.forbes.com/sites/jacobmorgan/2014/05/13/simple-explanation-internet-things-that-anyone-can-understand/#f1c5a711d091>. [Accessed: 22-Aug-2018].
- [9] C. Popoviciu, “IPv6, SDN: When worlds collide ... in a good way,” searchSDN, 2013. [Online]. Available: <http://searchsdn.techtarget.com/feature/IPv6-SDN-When-worlds-collide-in-a-good-way>. [Accessed: 10-Aug-2018].
- [10] D. Serrano and J. Guerri, “Redes Definidas por Software (SDN): OpenFlow,” Universitat politecnica de valencia, 2015.
- [11] L. J. Arizmendi, “Ideas básicas sobre Software Defined Network (SDN),” blogspot, 2013. [Online]. Available: <http://luisarizmendi.blogspot.com.co/2013/11/software-defined-network-sdn.html>. [Accessed: 15-Sept-2018].
- [12] SdxCentral, “Understanding the SDN Architecture,” Sdx, 2016. [Online]. Available: <https://www.sdxcentral.com/sdn/definitions/inside-sdn-architecture/>. [Accessed: 23-Aug-2018].
- [13] Open Networking Foundation, “SDN Architecture Overview,” Onf, no. 1, pp. 1–5, 2013.
- [14] SDx Central, “Open Network Operating System (ONOS),” SDx. .
- [15] Project Floodlight, “Floodlight,” Project Floodlight, 2017. .
- [16] SDx Central, “What is a Floodlight Controller?,” SDx Central. .
- [17] SdxCentral, “What is OpenFlow? Definition and how it relates to SDN,” SdxCentral, 2016. [Online]. Available: <https://www.sdxcentral.com/sdn/definitions/what-is-openflow/>. [Accessed: 23-Aug-2018].
- [18] M. Rouse, “Mininet,” searchSDN, 2013. [Online]. Available: <http://searchsdn.techtarget.com/definition/Mininet>. [Accessed: 23-Oct-2018].