[1]

Emulación de Software Defined Network

Esteban Salazar Zapata. esteban.salazar1@udea.edu.co, Rafael Luna Pérez. rafael.luna@udea.edu.co Universidad de Antioquia, Departamento de Ingeniería de Sistemas

***Resumen*— El imparable proceso de comercialización en los últimos años, las redes informáticas y la virtualización, han definido un nuevo paradigma del que surgen muchas oportunidades de mercado y de investigación; sin embargo un tanto restringidas por las redes tradicionales, que son controladas por las grandes industrias dominantes del sector, y que causan a demás limitaciones a lo que se puede lograr con ellas. Las redes definidas por software se han convertido en la puerta que al lograr ser permanentemente abierta, puede cambiar este paradigma con nuevos conocimientos, ampliando el horizonte de los sistemas de cómputo y la investigación, incluyendo las industrias que buscan ser independientes en el establecimiento de las condiciones y formas en que responden sus redes, además de la preferencia por los proveedores de hardware necesarios para las redes. Este documento muestra las tecnologías actualmente disponibles para el campo de la investigación SDN, concepto que busca revolucionar las redes actuales.**

***Palabras claves*— Topología de red, Internet, OpenFlow, Beacon, controlador, router, openFlow Switch, SDN,**

***Abstract*— The unstoppable process of commercialization in recent years, computer networks and virtualization, framed a new paradigm from what many market opportunities arise and research, however somewhat restricted by traditional networks, which are controlled by major key industries in the sector, also than a limitation to what can be achieved that these have caused. Software defined networks have become the door that when achieve be permanently open, it can change this paradigm completely with new insights, broadening the horizon of systems computer and research, including industries that seek to be independent in setting the conditions and ways in which they respond their networks, apart from the preference for hardware vendors necessary for networks. This document shows the technologies Currently available for the field of research SDN That concept Attempts to revolutionize today's networks.**

***Keywords*— Network Topology, Internet, OpenFlow, Beacon, controller), router, openFlow Switch, SDN,**

I. INTRODUCCIÓN

Las redes de datos en la actualidad son muy estáticas, esto se evidencia cuando se considera lo siguiente: Primero cada tipo y marca de router difieren en el tipo de configuración entre sí, variando desde el tipo de firmware hasta los comandos de instanciación para la implementación de las arquitecturas de red. Segundo, la integración en redes es vertical porque una empresa construye tanto el hardware como el sistema operativo y las aplicaciones. Tercero, para realizar un cambio en la red se requiere hacer cambios en el hardware.

La evolución de los sistemas estáticos a dinámicos es la razón de los grandes avances informáticos, hoy en día como cloud computing, cluster systems, entre otros; hicieron atractiva la idea de convertir las redes actuales estáticas, dada su configuración, a sistemas de redes que sean dinámicas y abiertas a una programación adaptable a las necesidades pertinentes de su implementación.

Desde su aparición, la atención se ha centrado en las redes de datos basadas en software o SDN por sus siglas en inglés (Software defined network), en ellas se busca separar el plano de control del plano de datos y así llevar todo el mundo de las redes de datos a nuevos niveles, con nuevos servicios y aplicaciones.

En este documento se muestran algunos conceptos básicos sobre SDN, el objetivo del proyecto integrador, y antecedents cercanos a nuestro objetivo.

[1].

II. REDES DEFINIDAS POR SOFTWARE (SDN)

El aumento hoy en el tráfico de datos es una de las razones por las que se hacen limitadas las redes actuales; dada su configuración estática, estas están construidas con funcionalidades fijas y con orientaciones a ser disponibles; es decir, se garantiza que el flujo de paquetes siempre esté disponible, administrando la forma cómo y hacia donde viajan los paquetes usando reglas y parámetros estáticos.

En las redes actuales, los dispositivos de red deciden a través de un parámetro fijo (tablas de enrutamiento, por dirección IP y MAC), el camino por el cual enviar los paquetes; ahora bien, ¿qué, si se quisieran enviar considerando parámetros dinámicos?, como el costo de ancho de banda, el enlace con menor latencia, la ruta más confiable o incluso parámetros externos a la red, tales como la temperatura, leyes que regulan el uso de la tecnología, entre otros; se hace imposible hacerlo a través de las redes tradicionales. SDN ofrece una versatilidad suficiente y realmente abre el cambio del paradigma sobre cómo se ven las redes hoy, controlando el flujo de datos de una manera programable, y con un control bajo demanda, es decir que sean las aplicaciones las que controlen cómo se mueven los datos tomando la información obtenida de la red para decidir según la circunstancia que hacer sobre el tráfico de datos.

SDN se centra en el concepto de poder separar el plano de control del plano de datos, es decir, separar el software de los dispositivos que conmutan los paquetes dentro de la red [11]. Es una propuesta muy atractiva que ha surgido con gran aceptación para hacer las redes mucho más potentes, extensibles, flexibles y programables, independientes de la intervención de la mano del hombre.

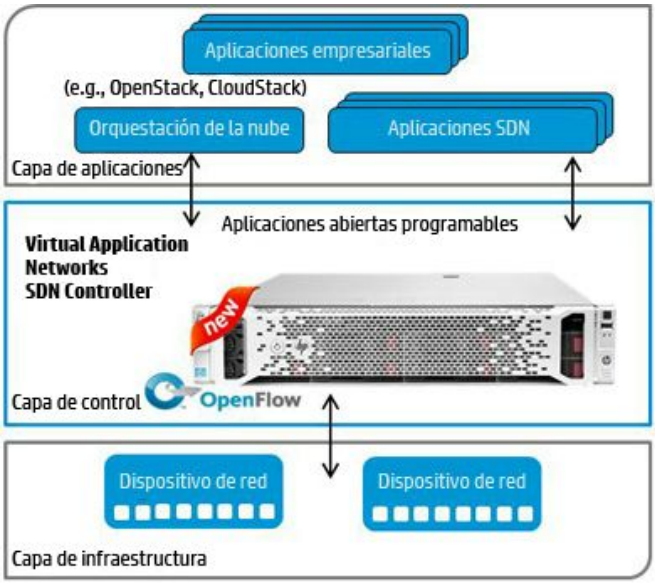

Imagen 1: Arquitectura SDN

III. SOFTWARE

OpenFlow

Es un estándar abierto que permite correr protocolos experimentales en redes locales. OpenFlow se agrega como una característica a los switches ethernet comerciales, routers, access Points; y provee un enganche estandarizado que habilita a los investigadores experimentar con los dispositivos de red sin que los proveedores tengan que exponer el funcionamiento interno de sus equipos.

Este protocolo ha sido diseñado para sostenerse en tres componentes: las tablas de flujo instaladas en los elementos

de datos para indicar a cada dispositivo lo que debe hacer en el tráfico de datos. El controlador el cual dialoga con los elementos de datos y les transmite la información que necesiten y por último los elementos de datos (switches) [11].

El protocolo OpenFlow en sí, consiste en un estándar abierto mediante el cual se especifican las entradas en la tabla de flujos. OpenFlow es una Southbound API, o un protocolo de comunicación entre la capa de control y la capa de infraestructura de red [10], permite que el switch se comunique con el controlador en una dirección ip usando un puerto a través de una conexión TCP.

El funcionamiento de este protocolo está dado porque al separar el plano de datos del plano de control, se puede tener un mejor control de la red, y por tanto, mayor eficiencia, El protocolo es el medio por el cual un dispositivo opera según la configuración que se haga en el controlador y en la imagen OPENFLOW SWITCH se muestra claramente cómo se ubica el protocolo entre él y el controlador.

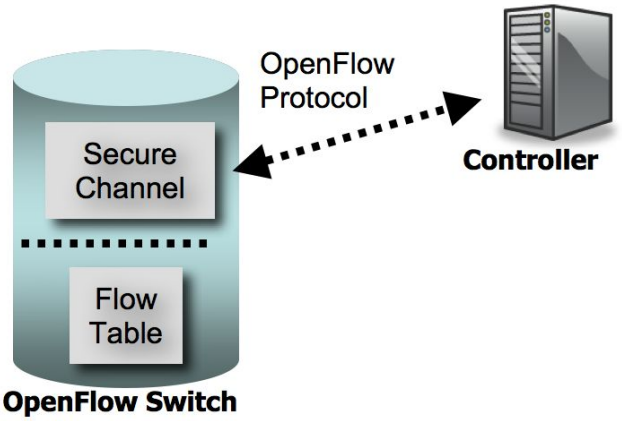

Imagen 2

IV. CONTROLADORES

Como ya se ha mencionado anteriormente, la visión de SDN es lograr mediante un proceso de abstracción obtener la toma de decisiones y definirla en el software para hacer las redes automatizables y menos dependientes del hardware, es aquí donde el controlador se constituye como el principal componente de la arquitectura SDN. El es quien toma las decisiones, implementa las reglas de la red, ejecuta las instrucciones que le proporcionan las diferentes aplicaciones y las distribuye a los diferentes dispositivos de capa física de la red [10].

Durante el tráfico de paquetes de la red, cuando un switch recibe un paquete, este se encarga de verificar de dónde viene y hacia dónde va, sin embargo cuando este no sabe qué acción realizar respecto al paquete, se lo envía al controlador quien analiza las cabeceras, inserta las reglas necesarias para que el switch pueda enrutar correctamente dicho paquete; el controlador descarta el paquete, en caso de que éste no sea un paquete ethernet.

Hay diferentes implementaciones de controladores. Desde un simple software que dinámicamente añada y suprima flujos, donde el administrador controla toda la red de switches OpenFlow y es el responsable del proceso de todos los flujos hasta una implementación con múltiples administradores, cada uno con diferentes cuentas y passwords, que les permite gestionar diferentes conjuntos de flujos. Es lo que se podría asimilar como una virtualización de una red con múltiples propietarios [10].

Beacon Controller

Beacon es un controlador OpenFlow de código abierto basado en Java creado en 2010. Ha sido ampliamente utilizado para la enseñanza , la investigación, y como la base del controlador Floodlight [9], este a su vez es la base para un controlador llamado OpenDayLight el cual es más robusto. Beacon es un controlador SDN muy bien escrito y organizado en Java y altamente integrado en el IDE de Eclipse. Fue el primer controlador que hizo posible que los programadores principiantes trabajasen con un ambiente SDN.

Mininet

Mininet es un emulador de red que ejecuta un conjunto de elementos de datos como switches y routers en un solo core de Linux. Se utiliza para representar virtualmente un sistema que parezca una red completa. [13]
Los programas que se ejecutan pueden enviar paquetes a través de lo que parece ser una interfaz de Ethernet real, con una velocidad de enlace y con retardo. Los paquetes se procesan por lo que parece un verdadero interruptor de Ethernet, un enrutador o middlebox.
Ahora bien, los dispositivos virtuales de MiniNet, enlaces, conmutadores y los controladores se crean en lugar de hardware utilizando software, en gran parte el comportamiento es similar a los elementos de hardware. La eficiencia de mininet permite que con solo un comando se ejecute la red [12].

Mininet utiliza hosts virtuales, switches y enlaces para crear una red en un solo núcleo del sistema operativo, y utiliza la pila de red real para procesar paquetes y conectarse a las redes reales. Además, las aplicaciones de red basadas en Unix/Linux, también se pueden ejecutar en los hosts virtuales. En una red OpenFlow emulada por Mininet, una aplicación de controlador real OpenFlow se puede ejecutar en una máquina externa o en el mismo equipo en el que se emulan los hosts virtuales

A pesar que mininet es un software simple de manipular y bastante rápido, tiene algunas limitaciones como el poco desempeño, la emulación solo de redes cableadas y el uso de solamente un controlador a la vez. Además [14]:

- El correr mininet sobre un solo sistema es conveniente pero limita los recursos.
- MiniNet utiliza un solo núcleo de Linux para todos los hosts virtuales, por lo que no se puede correr sobre un sistema que dependa de núcleos de otro sistema operativo.
- Las especificaciones de enrutamiento personalizado de un controlador debe ser desarrollado con las características que se requieran.
- Todos los hosts MiniNet comparten el sistema de archivos de host y el espacio PID; esto significa que es necesario tener cuidado si están ejecutando demonios que puede desbordar los recursos asignados al proceso.
- A diferencia de un simulador, MiniNet no tiene una fuerte noción de tiempo virtual; esto significa que mediciones de temporización se basan en tiempo real. Por ejemplo, 100 Gbps en redes no pueden ser fácilmente emulados.

V. ANTECEDENTES

[8]. En este artículo se lleva a cabo la creación de escenarios virtuales con Visual Network Descripcion el cual es una plataforma web que tiene una interfaz gráfica para la creación de topologías de red SDN a ser utilizadas en la simulación y análisis de las mismas, recreando la red del campus perteneciente a la universidad del Sinú.

[12]. En este artículo se lleva a cabo la simulación de una red con topología en estrella, la cual está compuesta por 12 host, 3 switch y un controlador, con el código implementado en python siguiendo el API ofrecido en mininet.

Mininet: una herramienta versátil para emulación y prototipado de Redes Definidas por Software [13]. En este artículo se habla la historia de SDN, también nos habla de otras ideas que surgieron para solucionar el problema de la falta de dinamismo en las redes, y por último nos habla de la herramienta Mininet y su importancia en la investigación de las redes basadas en el paradigma de Redes definidas por software.

VI. PLANTEAMIENTO

Los altos costes y dificultades que representan el llevar a cabo un proyecto experimental para estudiar el comportamiento de nuevas tecnologías de redes, como lo es el pasar de las redes tradicionales a unas redes basadas en software. Dan paso a la apreciación de los verdaderos beneficios que ofrece la virtualización, como medio de experimentación suficiente para determinar qué resultados arrojan los escenarios allí recreados.

Es aquí donde; como fue descrito anteriormente, mininet nos permite lograr los objetivos de nuestra investigación, que además de aportar a nivel educativo, es mostrar la integración horizontal que se logra al utilizar redes de datos basadas en software, figurar el futuro de las redes, al llevar estas a las redes virtualizadas.

Se pretende entonces:

- Emular, en un ambiente virtual, una SDN usando el software mininet para simular distintas topologías, y un controlador llamado Beacon, el cual usa el protocolo Openflow; con el que se puede controlar las redes según las políticas que se definan., disociadas de las redes físicas, para satisfacer una amplia gama de requisitos que no son posibles en las redes actuales.

- Visualizar el efecto del controlador sobre el tráfico de datos, haciendo flexible la red a un comportamiento dinámico.

- Mostrar que el controlador funcione de 2 formas: Una como un Hub, haciendo que el switch envía datos a todos los puertos menos del cual llegó. Y como un switch, porque cada vez que llega un paquete este lo envía al controlador, el cual verifica el destino y agrega a la tabla de MAC, si no existe, el origen y el puerto del paquete de entrada, y luego envía el paquete al switch para ser enviado a destino.

VII. MONTAJE

Infraestructura:
Un computador portátil con 8GB de RAM y un procesador AMD A10 QuadCore, en este se tiene un sistema operativo Windows 10, en el cual se instalaron y corrieron los siguientes programas:

Eclipse: Programa donde se corre el controlador Beacon.
VirtualBox: Software donde se corre la máquina virtualizada con sistema operativo Ubuntu, donde corre el programa mininet.
Xming: Software que se utiliza para acceder a los terminales de los nodos en las redes simuladas.
Putty: Software que se utiliza para acceder a la máquina virtual de Ubuntu por SSH desde Windows 10.

Red

Se simularon varias redes con diferentes topologías, con un controlador Beacon que se encuentra en la dirección ip = 192.168.0.37, el número de dispositivos en cada red difiere según la topología.

Topologías analizadas

Lineal: Todos los host están conectados a un switch y no hay conexión directa entre ninguno de ellos. Se simulan 8 switches y cada switch se conecta un host.

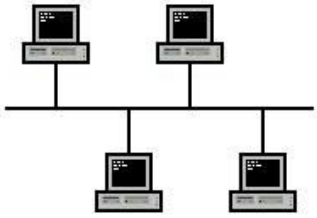

Imagen 3

En la imagen 4, se puede observar el comando para montar la red lineal y se aplica el test de alcanzabilidad haciendo ping entre todos los hosts con un éxito del 100 %.

```
mininet@mininet-vm: ~
                    node placement for --cluster (experimental!)
mininet@mininet-vm:~$ sudo mn --topo linear,8 --mac --switch ovsk --controller r
emote,ip=192.168.0.37
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8
*** Adding switches:
s1 s2 s3 s4 s5 s6 s7 s8
*** Adding links:
(h1, s1) (h2, s2) (h3, s3) (h4, s4) (h5, s5) (h6, s6) (h7, s7) (h8, s8) (s2, s1)
 (s3, s2) (s4, s3) (s5, s4) (s6, s5) (s7, s6) (s8, s7)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8
*** Starting controller
c0
*** Starting 8 switches
s1 s2 s3 s4 s5 s6 s7 s8 ...
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6 h7 h8
h2 -> h1 h3 h4 h5 h6 h7 h8
h3 -> h1 h2 h4 h5 h6 h7 h8
h4 -> h1 h2 h3 h5 h6 h7 h8
h5 -> h1 h2 h3 h4 h6 h7 h8
h6 -> h1 h2 h3 h4 h5 h7 h8
h7 -> h1 h2 h3 h4 h5 h6 h8
h8 -> h1 h2 h3 h4 h5 h6 h7
*** Results: 0% dropped (56/56 received)
mininet>
```

Imagen 4: Topología Lineal

El comando se entiende de la siguiente forma: sudo mn para crear la red, --topo linear,8 para que sea una topologia lineal de 8 hosts y 8 switches, --mac para asignar MACs automáticamente a los hosts, --swtich ovsk para que los switches sean de tipo OpenvSwitch y por ultimo --controller=remote,ip=192.168.0.37 para decirle a la red que el controlador es remoto y esta en la ip 192.168.0.37

Star: Los host están conectadas directamente a un switch central y todas las comunicaciones que han de hacerse, necesariamente son a través de este.

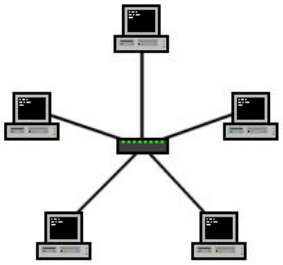

Imagen 5

En la imagen 6, se puede observar el comando para montar la red en estrella o simple y se aplica el test de alcanzabilidad haciendo ping entre todos los hosts con un éxito del 100 %

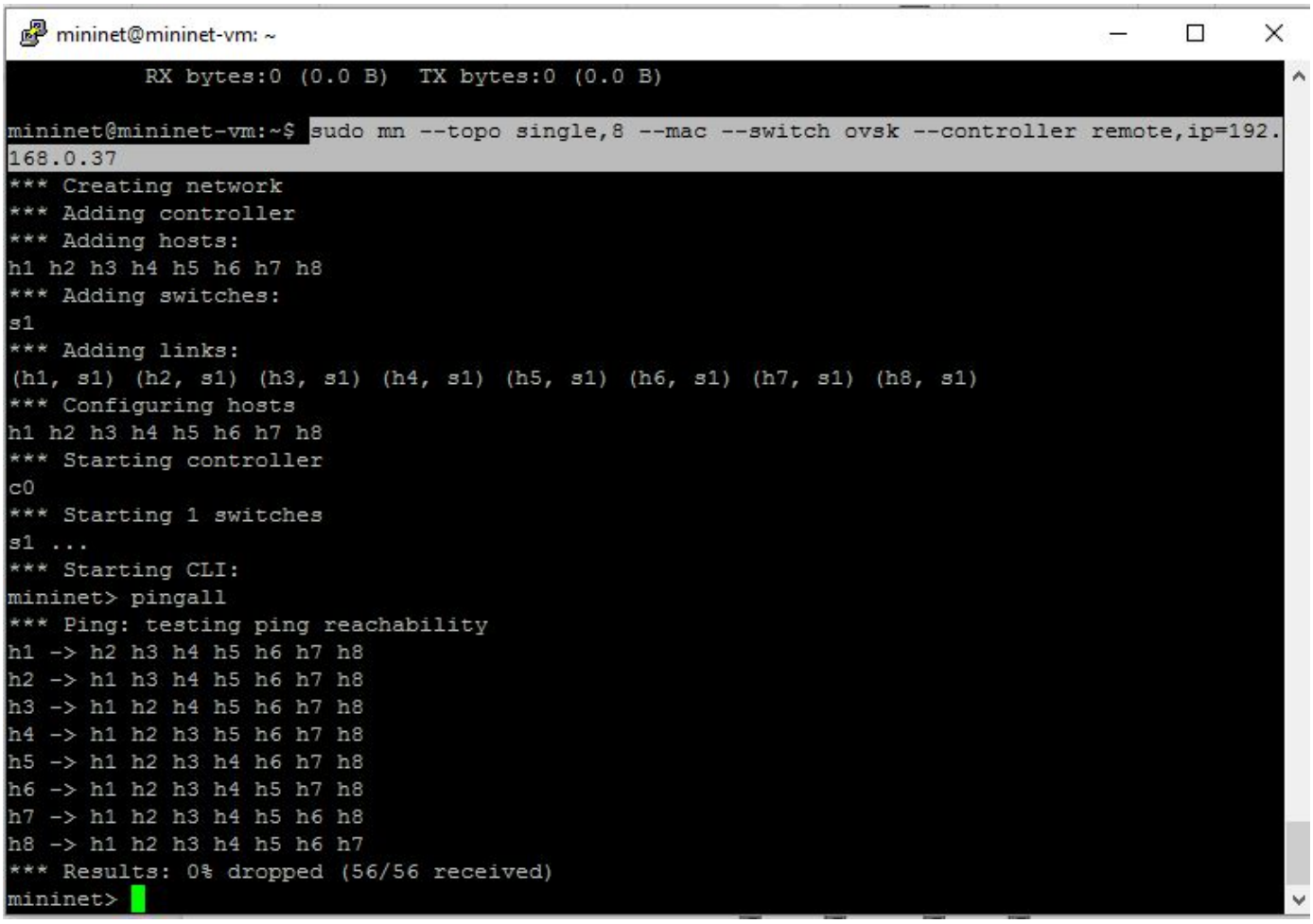

Imagen 6: Topología en Estrella

El comando se entiende de la siguiente forma: sudo mn para crear la red, --topo single,8 para que sea una topologia de estrella de 8 hosts y 1 switch, --mac para asignar MACs automáticamente a los hosts, --swtich ovsk para que los switches sean de tipo OpenvSwitch y por ultimo --controller=remote,ip=192.168.0.37 para decirle a la red que el controlador es remoto y esta en la ip 192.168.0.37

Tree: Los host están colocados en forma de árbol, como una serie de redes en estrella interconectadas pero sin un nodo central

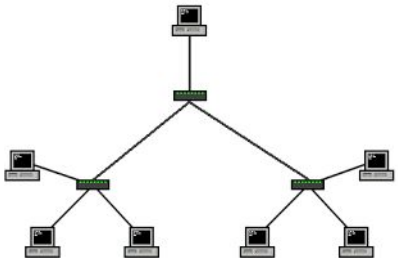

Imagen 7

En la imagen 8, se puede observar el comando para montar la red en árbol y se aplica el test de alcanzabilidad haciendo ping entre todos los hosts con un éxito del 100 %

```
mininet@mininet-vm: ~
completed in 621.237 seconds
mininet@mininet-vm:~$ sudo mn --topo tree,3 --mac --switch ovsk --controller remote,ip=192.16
8.0.37
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8
*** Adding switches:
s1 s2 s3 s4 s5 s6 s7
*** Adding links:
(s1, s2) (s1, s5) (s2, s3) (s2, s4) (s3, h1) (s3, h2) (s4, h3) (s4, h4) (s5, s6) (s5, s7) (s6
, h5) (s6, h6) (s7, h7) (s7, h8)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8
*** Starting controller
c0
*** Starting 7 switches
s1 s2 s3 s4 s5 s6 s7 ...
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6 h7 h8
h2 -> h1 h3 h4 h5 h6 h7 h8
h3 -> h1 h2 h4 h5 h6 h7 h8
h4 -> h1 h2 h3 h5 h6 h7 h8
h5 -> h1 h2 h3 h4 h6 h7 h8
h6 -> h1 h2 h3 h4 h5 h7 h8
h7 -> h1 h2 h3 h4 h5 h6 h8
h8 -> h1 h2 h3 h4 h5 h6 h7
*** Results: 0% dropped (56/56 received)
mininet>
```

Imagen 8: Topología en Árbol

El comando se entiende de la siguiente forma: sudo mn para crear la red, --topo tree,3 para que sea una topologia en arbol de 8 hosts y 7 switches, --mac para asignar MACs automáticamente a los hosts, --swtich ovsk para que los switches sean de tipo OpenvSwitch y por ultimo --controller=remote,ip=192.168.0.37 para decirle a la red que el controlador es remoto y esta en la ip 192.168.0.37

VIII. CONCLUSIONES

Las exigencias hacia las redes, como la movilidad y virtualización, generan la necesidad de responder a las condiciones de negocio actuales, cambiando las redes tradicionales por una nueva visión como Redes definidas por Software, que aprovechan los recursos actuales y no genera grandes traumatismos como podría ocurrir si se pensara en cambiar toda la red.

En este trabajo concluimos, que al aplicar SDN separando la capa lógica de la capa de datos, podemos visualizar la flexibilidad que se otorga a la red mediante el controlador, Beacon para este caso, dado que si el switch no sabe qué hacer con un paquete, el controlador se encarga de resolver el flujo de este según las políticas con las que fue programado. De igual manera la independencia que adquiere la red sobre la topología; es decir sin importar la topología seleccionada, la red sigue funcionando sin problemas como arrojaron los resultados durante la emulación.

Además se puede ver como después de tener la infraestructura montada, y con suficiente conocimiento del protocolo openflow y de programación, se puede lograr una red sencilla y dinámica en muy poco tiempo sin tener que cambiar la configuración en cada switch, como pasa en una red tradicional, ahorrando tiempo a los ingenieros encargados. Además con este nuevo tipo de red se obtiene más control sobre la red, pudiendose ofrecer nuevos servicios que mejorarían la red y optimizarán el uso de esta.

REFERENCES

[1] *N. Feamstar, J. Rexford y E. Zegura, «The Road to SDN: An Intellectual History of Programmable Networks,» 10 Octubre 2013. [En línea]. Available: http://www.cs.princeton.edu/courses/archive/fall13/cos597E/papers/sdnhistory.pdf.*

[2] *D. Kreutz, F. M. V. Ramos, P. Verissimo, C. S. Rothenberg, S. Azodolmolky y S. Uhlig, «Cornell University Library,» IEEE, 8 Octubre 2014. [En línea]. Available: http://arxiv.org/pdf/1406.0440v3.pdf.*

[3] *B. Valencia, S. Santacruz, L. Becerra y J. Padilla, «Entre ciencia e Ingeniería,» 2009. [En línea]. Available: http://biblioteca.ucp.edu.co/ojs/index.php/entrecei/article/viewFile/2492/2364.*

[4] *A. M. R. Calero y Á. Pachón, «Universidad ICESI,» [En línea]. Available: https://bibliotecadigital.icesi.edu.co/biblioteca_digital/bitstream/10906/68434/2/articulo-redes-sdn.pdf.*

[5] *U. Javed y A. Iqbal, «an-dash,» [En línea]. Available: http://andash.seecs.nust.edu.pk/andash_publications/SDN.pdf.*

[6] *V. A. M. Galué, «Departamento de Ingeniería de Sistemas Telemáticos - ETSIT - UPM,» 2012. [En línea]. Available: http://www.dit.upm.es/~posgrado/doc/TFM/TFMs2011-2012/TFM_Vanessa_Moreno_2012.pdf.*

[7] *R. M. Peralta, «Departamento de Electrónica - Universidad Técnica Federico Santa María,» [En línea]. Available: http://profesores.elo.utfsm.cl/~agv/elo323/2s14/projects/reports/RodrigoManriquez/RodrigoManriquez.pdf.*

[8] *G. E. Duarte y R. J. L. U., «EMULACIÓN DE ESCENARIOS VIRTUALES, EN UNA SDWLAN (SOFTWARE),» Revista Cientifica Electronica - Universidad del Sinú, 2016.*

[9] *D. Erickson, «The McKeown Group - Deparment of Computer Science, Stanford University,» [En línea]. Available: http://yuba.stanford.edu/~derickso/docs/hotsdn15-erickson.pdf.*

[10] *O. R. Hervás, «Universidad Politecnica de Catalunya,» [En línea]. Available: http://upcommons.upc.edu/bitstream/handle/2099.1/21633/Memoria.pdf.*

[11] *D. F. B. Gómez, «Dialnet,» 2013. [En línea]. Available: http://dialnet.unirioja.es/servlet/articulo?codigo=4897871.*

[12] *W. A. V. Vargas, «Academia,» [En línea]. Available: http://www.academia.edu/5730624/Emulaci%C3%B3n_de_una_red_definida_por_software_utilizando_MiniNet.*

[13] B. Valencia, S. Santacruz, L. Becerra y J. Padilla, «Univerisdad Catolica de Pereira,» 29 Mayo 2015. [En línea]. Available: http://biblioteca.ucp.edu.co/ojs/index.php/entrecei/article/viewFile/2492/2364.

[14] MarineFury «Network Simulation with Mininet,» 22 Junio 2015. https://projectmincpm.wordpress.com/2015/06/22/restrictions-to-mininet/