Seguridad y rendimiento en redes híbridas SDN

Software Defined Networks

Autores: Ing. Lucas Ezequiel Bujedo, Gastón Alejandro Borja Perazzi, Emiliano Ortin Calabrese.
Tutor: Ing. Juan Galleguillo
lbujedo873@alumnos.iua.edu.ar, gborja333@alumnos.iua.edu.ar, eortin558@alumnos.iua.edu.ar
Depto. de Electrónica y Telecomunicaciones
Instituto Universitario Aeronáutico – Facultad de ingeniería
Córdoba, Argentina

***Resumen*—En este artículo se pretende informar sobre esta arquitectura de red emergente, sus ventajas y desventajas en relación con la seguridad sobre la red tradicional y componentes básicos. También se explicará el protocolo OPENFLOW, un protocolo emergente y abierto que facilita la implementación de SDN y los Switch OpenFlow. Por último se mostrará una implementación de red física definida por software con switches HP 2920-24G y controlador Floodlight.**

***Palabras Clave*—SDN, Openflow, Red híbrida, Switch Openflow, Controlador Floodlight.**

I. Introducción

Las redes de comunicación actuales enfrentan problemas que se van volviendo más críticos con el paso de los años, y hace tiempo se están buscando soluciones que se adapten tanto a las necesidades de los clientes como a las de las grandes empresas. Dentro de estos problemas podemos encontrar la gran cantidad de datos que se necesitan enviar por la red, la necesidad de seguridad avanzada, mayor eficiencia en el procesado de paquetes para no generar cuellos de botella en los enrutadores ni colisiones, propiedades de escalabilidad acordes al crecimiento de la población y más incógnitas para las cuales las redes actuales resultan obsoletas [5].

Aquí es donde entran en juego las Redes Definidas por Software (SDN) y sus aplicaciones con respuestas a todas las interrogantes que presentan las redes actuales, brindando un cambio de paradigma con respecto, no solo a las tecnologías nuevas a utilizar, sino también, cómo se deben aplicar.

Existen muchas herramientas que sirven para probar su seguridad en forma análoga a la seguridad de redes convencionales. El objetivo de este trabajo, es explicar los aspectos a tener en cuenta sobre la seguridad de este nuevo tipo de redes y marcar la diferencia con sus predecesoras.

II. Redes Definidas por Software

Esta nueva arquitectura de red presenta una característica clara que es la separación de las funciones de control y gestión de las funciones de "forwarding", dando como resultado una red dinámica, manejable, centralizada, adaptable y rentable,

ideal para las necesidades de las empresas y clientes de hoy en día.

Al entender este concepto de la separación del plano de control y el plano de datos, podemos imaginar a la red como un gran conjunto de dispositivos de conmutación (switch, routers, etc), conectados entre sí por líneas de transmisión y, separado de estos, la capa de control por encima compuesta por un controlador que es el "cerebro" de la red y toma las decisiones sobre los paquetes que circulan por ella.

Dentro de las características que presenta esta nueva arquitectura y que sacan ventaja a las redes convencionales se encuentran [1]:

- **Gestión centralizada:** Permite gestionar la totalidad de la red desde un único punto, eliminando fallas de seguridad, cuellos de botellas, errores de configuración, etc. que traen consigo los múltiples puntos de decisión.
- **Agilidad:** Se puede reservar ancho de banda para servicios especiales o QoS con mayor rapidez y adaptarse a los cambios y prioridades de la red.
- **Bajos costos:** Al no tener que optar por la homogeneidad de productos y tener un proveedor genérico, se abaratan los costos de la red.
- **Programación automática:** SDN permite a los administradores, por medio de programas automatizados que ellos mismos pueden escribir ya que no depende de software propietario, gestionar y optimizar los recursos de la red de forma rápida y dinámica.
- **Fácilmente programable**: Brinda la posibilidad de programar o configurar el plano de control con mayor facilidad al estar separado del plano de red.
- **Escalabilidad:** Con esta arquitectura, es muy fácil proyectar una red que crezca en gran medida sin perder ninguna de sus características.

- **Basado en estándares abiertos y proveedores neutrales:** Como se implementa a través de estándares abiertos, simplifica el diseño de la red y las operaciones porque las instrucciones son proporcionadas por el controlador, en lugar de múltiples dispositivos, proveedores específicos o protocolos.

- **Seguridad:** Al estar la red centralizada por el controlador, no se corre el riesgo de poseer agujeros de seguridad en las configuraciones de los switchs.

Todas estas características que poseen las SDN y no las redes tradicionales nos llevan a pensar que en un futuro no muy lejano, esta arquitectura será ampliamente aceptada por todos los sectores del mercado de telecomunicaciones, favoreciendo tanto a los clientes que buscan velocidad, seguridad, y robustez en sus conexiones, como a las empresas que buscan brindar un servicio de alta calidad y confiabilidad con el menor costo de operación y de inversión posible.

III. Arquitectura lógica de las SDN

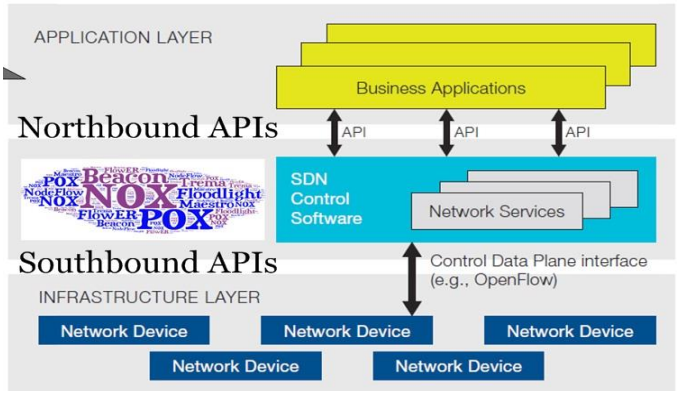

Figura 1: Arquitectura lógica SDN [12]

Examinando las redes definidas por software en profundidad, podemos separarlas en tres capas bien diferenciadas. La capa inferior o capa de hardware, compuesta por el conjunto de switchs, routers, hosts, medios de transmisión, etc, que pueda contener el sistema. La capa del medio recibe el nombre de capa de control, compuesta por el controlador SDN, que se encarga de las decisiones a tomar sobre los paquetes en la red y como deben responder los dispositivos de la capa de hardware. En esta capa se encuentra el sistema operativo de red (NOS) y posee una visión global de la red. Se comunica con la capa inferior por medio de las SouthBound APIs, que permite el diálogo entre el controlador y los dispositivos de red. Entre estas APIs se encuentra OPENFLOW.

La capa superior o capa de aplicaciones está compuesta por las aplicaciones de usuario y se comunica con el controlador por medio de las NorthBound APIs. Esta parte es muy importante y es la que saca ventaja sobre las redes tradicionales. Estas APIs incorporan los patrones de uso de la red, esto quiere decir que, la red actuará de diferente manera de acuerdo a la aplicación que se necesite. Entonces la capa de aplicación comunica los patrones de uso de la red de una aplicación por medio de las NorthBound APIs al controlador, este toma la decisión más inteligente y se comunica con la capa inferior por medio de Openflow para indicar que hacer con los paquetes [8].

IV. Openflow

Es una tecnología de switching y se define como "un protocolo emergente y abierto de comunicaciones que permite a un servidor de software determinar el camino de reenvió de paquetes que debería seguir una red de switches"[8]. Su propósito es ser el comunicador entre el controlador y los distintos dispositivos de la red, logrando que la red se vea como un todo y sea el controlador el que tome las decisiones sobre los paquetes que viajan en ella, dejándole a los dispositivos de encaminamiento la función de reenvío solamente. Los switch convencionales reciben los paquetes, buscan coincidencias en sus tablas de flujos, y realizan la acción preestablecida. La diferencia con los switch OpenFlow radica en estas tablas, los switch convencionales poseen una tabla de flujo creada por el proveedor de dicho instrumento, lo que nos ata a los protocolos y decisiones que se hayan preestablecido sin poder realizar cambios. Por el contrario, con OpenFlow, los usuarios pueden configurar las tablas de flujo a su antojo y decidir cómo serán tratados los paquetes según las necesidades de cada red y usuario. Si algún paquete no encuentra coincidencia en su tabla de flujo, se enviará al controlador que lo procesará y devolverá al switch, agregando una nueva entrada de flujo a su tabla, creando una red dinámica, ágil, centralizada y automatizada [9].

Está claro que el protocolo OpenFlow está totalmente pensado para la implementación de las SDN y se presenta en la actualidad como el primer estándar desarrollado para las redes definidas por software. A pesar de sus características, la Open Networking Foundation, organización desarrolladora y promotora de Software Defined Networks, sigue trabajando para poder agregarle más beneficios para mejorar la interacción entre el controlador y los Switch OpenFlow.

V. Controlador SDN

Como ya se vió más arriba, el controlador es el núcleo de estas redes, es la parte más importante, el encargado de la "inteligencia" de la red. Él es quien toma las decisiones como la distribución de recursos, decisiones sobre paquetes que no coinciden con las tablas de flujo de los switchs, creación, modificación, o eliminación de entradas de flujo, ejecuta las instrucciones que le proporcionan las distintas aplicaciones,

básicamente, centraliza el funcionamiento de la red que es la característica básica de las SDN.

Un controlador puede ser, desde un simple software en una computadora, encargado de gestionar las tablas de flujos de los dispositivos de una red, quedando en mano de un solo administrador encargado de la gestión, hasta múltiples administradores de red que pueden gestionarla por medio de distintas cuentas y con la posibilidad de poder configurar distintos conjuntos de flujos de la red.

Dentro de los principales controladores OpenFlow que hay hoy en día están [8]:

- **NOX/POX:** Un controlador open-source que brinda una plataforma para escribir software de control de la red en C++ o Phyton.
- **Floodlight:** Controlador Open-Source basado en Java.
- **Beacon:** Controlador basado en JAVA.
- **Trema:** Completa plataforma OpenFlow para Ruby/C.
- **Maestro:** Plataforma de control escalable escrita en JAVA.
- **SNAC:** Controlador que usa una interfaz web para controlar la red.
- **IRIS:** Controlador basado en JAVA.

En este proyecto se utilizó Floodlight, por su interfaz web gráfica, facilitando nuestra investigación.

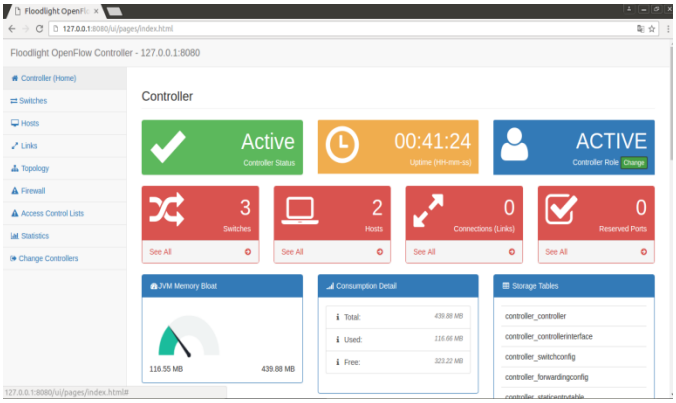

Figura 2: Interfaz gráfica Floodlight V1.2

VI. Switch OpenFlow

Estos dispositivos cuentan con tres partes:

- Una o más tablas de flujo que indican cómo debe ser procesado cada paquete que entre en el switch.
- Un canal especial o seguro que comunique el switch con el controlador. Esta comunicación se realiza por medio del protocolo OpenFlow.
- Un controlador que sea quien gestione las tablas de flujo del switch de acuerdo a las necesidades del usuario.

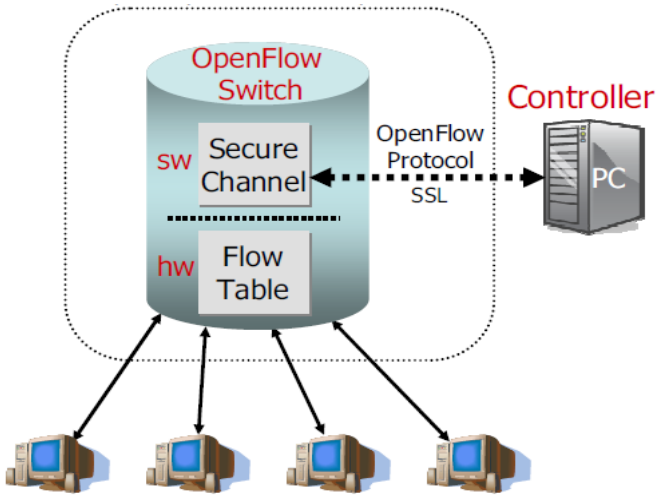

Figura 3: Arquitectura Switch OpenFlow[11]

Cada switch puede poseer una o más tablas de flujo utilizadas para decidir como redireccionar los flujos entrantes. Para eso, cada una de las entradas en las tablas de flujo poseen tres campos [9]:

- **Cabecera:** Se encarga de definir de que flujo se trata.
- **Acciones:** Que se realizarán cuando el paquete entrante encuentre coincidencia con alguna entrada de flujo de la tabla.
- **Estadísticas:** Que monitorizan el paso de flujos, número de paquetes, número de bytes, etc.

Dentro de las acciones que se pueden realizar, podemos encontrar tres:

- Reenviar paquete a través de un puerto.
- Encapsular el paquete y enviarlo al controlador para decidir qué hacer con él. Esto normalmente sucede cuando llega un paquete de un flujo nuevo.
- Eliminar paquete.

Con todo esto, gracias al protocolo OpenFlow, el switch solo debe recibir el paquete, buscar alguna coincidencia en la tabla de flujo y realizar la acción indicada, si no hay coincidencia se envía el paquete al controlador para que este decida si agregar una nueva entrada a las tablas de flujo o eliminar el paquete. Tomando en cuenta esto, se puede observar que en el único lugar donde hay retardo es cuando llega un nuevo flujo que no encuentra coincidencia en la tabla y debe ser examinado por el controlador, luego de que se agregue una nueva entrada de flujo, los próximos paquetes del mismo tipo no deberán ser enviados al controlador puesto que ya se ha agregado una entrada para decidir qué hacer con ellos. Esto también nos agrega seguridad, ya que podemos configurar el controlador para que elimine cualquier paquete que se considere peligroso para la red.

VII. Capa de aplicaciones

La consecuencia de este nuevo principio de diseño incrementa la flexibilidad en la gestión de la red. Por analogía a los teléfonos móviles, esta evolución puede ser comparada con el cambio de los teléfonos con funciones restringidas a la flexibilidad de los Smartphone con un sistema operativo y la habilidad de instalar varias aplicaciones.

La capa de aplicación permite a las business applications modificar e influenciar la forma en que la red se comporta para proveer servicios a los clientes. Esto requiere de la definición de una API, para permitir a desarrolladores terceros que desarrollen y vendan aplicaciones a los operadores de red.

El desarrollo de esa API todavía no fue definida por la ONF, pero es requerida para garantizar la interoperabilidad entre business applications y los controladores de red de diferentes proveedores.

Esta capa nos permite proporcionar mayor seguridad a la red mediante firewalls definidos por software dando la posibilidad de mitigar diferentes tipos de ataques como:

- **Ataques volumétricos como inundaciones SYN:** Estos ataques consisten en enormes cantidades de paquetes TCP solamente con las banderas SYN configuradas. Esto puede obstruir el ancho de banda, así como rellenar las colas de conexión en sistemas concretos que pueden tenerse en cuenta.
- **Ataques a aplicaciones y servicios específicos:** Estos ataques se dirigen a servicios web con series de peticiones HTTP muy particulares (usando cadenas específicas de agentes de usuarios con variables específicas de *cookies* y similares).
- **Ataques DDoS enfocados contra el comportamiento del protocolo:** Estos ataques llenan las tablas de estado de los dispositivos de la red.

Los interruptores programados para SDN pueden ser capaces de actuar como primera línea de defensa en la identificación de patrones particulares y en umbrales de volumen de paquetes de una sola fuente o varias fuentes dentro de un plazo determinado. Estos interruptores pueden dejar ir el tráfico o reorientarlo usando otras técnicas y protocolos. La mayoría de los ruteadores y otras plataformas de redes carecen de este nivel de control granular.

Los controladores pueden ejecutar secuencias de comandos (*scripts*) y comandos que pueden actualizar rápidamente las direcciones MAC e IP y el filtrado de puertos, lo que permite una respuesta rápida y cambios a las políticas y normas de tráfico. Esto también libera otros dispositivos de red de la carga de manejar grandes cantidades de tráfico.

VIII. Implementación

El proyecto se basa en la implementación de una red híbrida entre switches OpenFlow y switches tradicionales con el objetivo de mostrar ventajas y desventajas de la aplicación de la seguridad en esta tecnología en diferentes topologías.

Para ello utilizamos tres switches HP 2920-24G J9726A, proporcionados por el Instituto Universitario Aeronáutico, que son híbridos por lo cual pueden funcionar con el protocolo OpenFlow y/o con protocolos tradicionales. Como controlador se utilizó Floodlight en su versión 1.2 (la más actual al momento de escribir este paper) que incorpora una interfaz web gráfica e intuitiva, que también incluye varios módulos complementarios entre los cuales se utilizó la funcionalidad de Firewall.

Para este proyecto se utilizaron tres configuraciones de red diferentes en las cuales se alternó el uso del protocolo OpenFlow entre los switches de frontera y el switch núcleo.

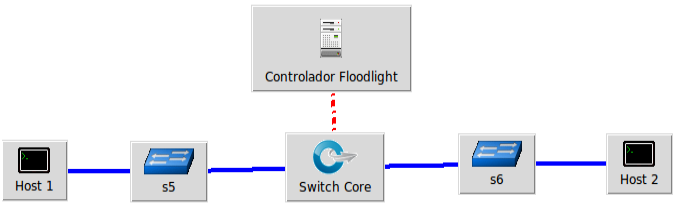

Figura 4: Primera topología

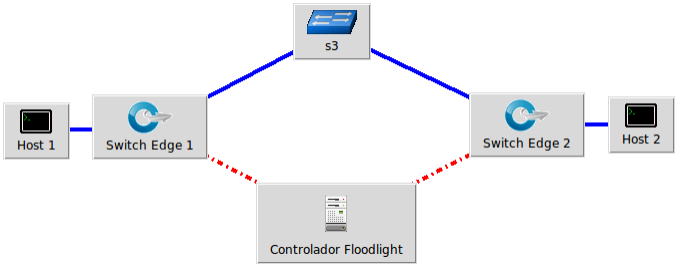

Figura 5: Segunda topología

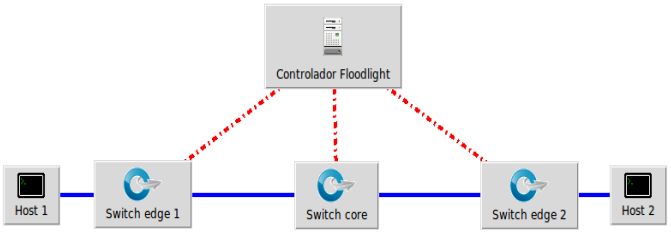

Figura 6: Tercera topología

El módulo de Firewall nos permite filtrar flujos de paquetes basado en diferentes criterios mostrados en la Tabla 1.

Field name	Possible values	Description
switchid	xx:xx:xx:xx:xx:xx:xx:xx	Switch identification number
src-inport	Short	Input switch port number
src-mac	xx:xx:xx:xx:xx:xx	Source MAC-address
dst-mac	xx:xx:xx:xx:xx:xx	Destination MAC-address
dl-type	ARP or IPv4	protocol
src-ip	A.B.C.D/M	Source IP-address
dst-ip	A.B.C.D/M	Destination IP-address
nw-proto	TCP or UDP or ICMP	Protocol
tp-src	short	Source port number
tp-dst	short	Destination port number
priority	Int	Priority of the rule (less is more important)
action	allow or deny	Allow or deny set of network flows which are match rule

Tabla 1: Parámetros de Firewall

Se tomó en consideración el tiempo que tardan los paquetes en ser procesados a lo largo de la red en las distintas topologías como parámetro del rendimiento. Para realizar esta prueba enviamos paquetes ICMP (Ping) desde un host final a otro observando el tiempo de respuesta en las distintas configuraciones.

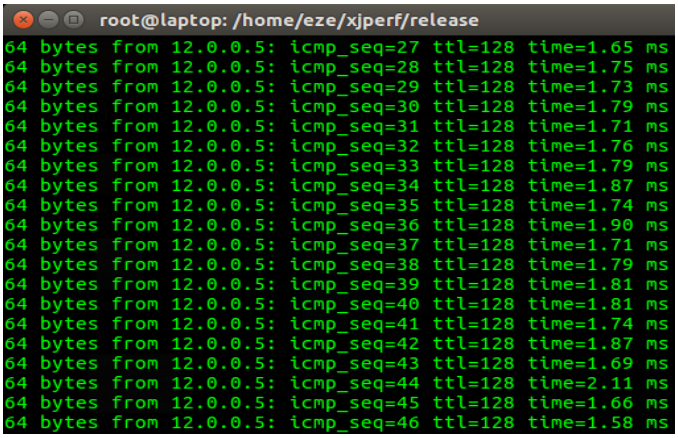

Figura 7: Ping con primera topología

En esta topología en donde solamente se aplica el Firewall al switch núcleo, se obtuvo la mejor performance de procesamiento de paquetes pero solo se consigue un nivel de seguridad alto en el centro de la red dejando que paquetes posiblemente maliciosos puedan circular en los otros nodos de la red.

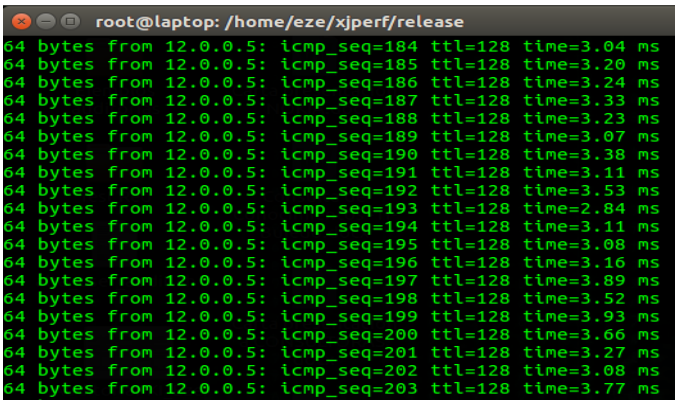

Figura 8: Ping con segunda topología

En la segunda topología disminuyó la velocidad de procesamiento de paquetes con respecto a la anterior, debido al doble procesamiento por parte de los switches OpenFlow, pero se incrementó la seguridad en la entrada a la red lo que genera un nivel de seguridad en la totalidad de la misma ya que los paquetes posiblemente maliciosos no lograrían circular hacia el interior.

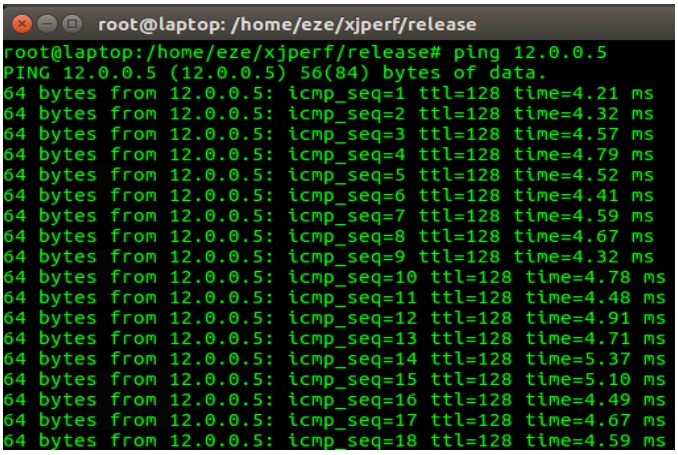

Figura 9: Ping con tercera topología

En la última topología se logra la mayor seguridad ya que los paquetes se filtran en cada nodo OpenFlow pero se obtiene el tiempo de procesamiento más alto de las tres configuraciones debido a la suma de tiempos de retardo en los procesamientos de cada paquete en todos los switches.

IX. Conclusión

Este nuevo paradigma de las redes llamado SDN trajo consigo un nuevo futuro positivo para las telecomunicaciones. Con todas las ventajas antes mencionadas como programación facilitada, agilidad, gestión centralizada, bajos costos, escalabilidad, etc. Provee soluciones a problemas que son muy comunes en la actualidad y para los cuales no se encontraban respuestas.

Esta nueva tecnología proporciona opciones de seguridad que las redes convencionales no. El hecho de poder aislar nodos de forma instantánea agiliza de gran manera el trabajo del administrador de red. Estas aplicaciones se implementan con solo correrlas en el controlador, y nos ofrecen un control más granular de los datos que circulan en la red, permitiendo el filtrado de paquetes según parámetros específicos.

De este trabajo podemos observar que de las tres configuraciones probadas la que aporta mejor equilibrio entre seguridad y rendimiento es la topología número dos dado que frente a un ataque de DoS o de inundación SYN los paquetes no logran propagarse por la red más allá de los switches frontera. Esto es muy importante a la hora de mantener la conexión de la red, ya que en caso de que el ataque tenga éxito solo se quedaría sin servicio una parte de la misma, la que es controlada por el switch atacado.

REFERENCIAS

[1] Open Networking Fundation. Disponible en: https://www.opennetworking.org

[2] Ing. Andres Ballesteros e Ing. Lucas Ezequiel Bujedo Tesis “ Seguridad en redes definidas por software”

[3] Ing Gaston Borja paper uEA 2015 “Redes definidas por software”

[4] James F. Kurose & Keith W. Ross, Redes de computadoras, un enfoque descendente, 5° ed., Pearson, 2010.

[5] Joseph D. Sloan, Network Troubleshooting Tools, 1° ed., O'Reilly, 2001.

[6] William Stallings, Comunicaciones y redes de computadores, 7° ed., Person-Prentice Hall, 2004.

[7] Thomas D. Nadeau & Ken Gray, SDN Software Defined Network, 1° ed., O'Reilly, 2013.

[8] Óscar Roncero Hervás, Tesis “Software Defined Network”, Universidad Politécnica de Catalunia, España. Disponible en: http://upcommons.upc.edu/pfc/bitstream/2099.1/21633/4/Memoria.pdf

[9] Wikipedia, OpenFlow. Disponible en: http://es.wikipedia.org/wiki/Openflow

[10] Wikipedia, Redes definidas por software. Disponible en: http://es.wikipedia.org/wiki/Redes_definidas_por_software

[11] Departamento de ciencias de la computacion de Stanford. Disponible en: http://yuba.stanford.edu/cs244wiki/index.php/Overview.

[12] Principia Technologica. Disponible en: http://principiatechnologica.com/2014/03/25/investigacion-de-redes-sdn-parte-iii-la-arquitectura-sdn/

[13]Floodlight, Disponible en: http://www.projectfloodlight.org/floodlight/