Evaluación de Convergencia del Protocolo OSPF en Redes Definidas por Software

Juan Ángel González[1], Fabio López Pires[2]

[1] Facultad de Politécnica. Universidad Nacional de Asunción
[2] Parque Tecnológico Itaipu
Paraguay
angelo_py@hotmail.com, fabio.lopez@pti.org.py

Resumen. El protocolo *Open Shortest Path First* (OSPF) es el protocolo de enrutamiento más utilizado en la actualidad. Las Redes Definidas por *Software* permiten implementar nuevos enfoques en el flujo de tráfico y enrutamiento. Este trabajo presenta una evaluación experimental del tiempo de convergencia del protocolo OSPF en Redes Tradicionales en comparación con Redes Definidas por *Software*, simulando diferentes condiciones de fallas simples y dobles en una red de datos, considerando una topología en anillo, con 5 nodos terminales, 5 enrutadores y 11 enlaces. Los resultados experimentales indican que en promedio, el tiempo de convergencia en las Redes Definidas por *Software* es hasta 82% menor ante fallas simples y hasta 78% ante fallas dobles.

Palabras Claves: Redes Definidas por *Software*, Enrutamiento Dinámico, Evaluación Experimental, Tiempo de Convergencia, OSPF.

1 Introducción

EL auge de las Tecnologías de la Información y de Comunicación (TIC), así como la expansión de las aplicaciones de banda ancha imponen estrictos requerimientos en términos de rendimiento de equipos y protocolos de enrutamiento en redes de datos, las cuales deben contar con rutas que garanticen el alcance a todos los nodos destinos, de manera a adaptarse a posibles cambios de topología, producidos por fallas en los enlaces o por flujo intenso de tráfico. Los enrutadores representan el punto clave en la infraestructura de red, debido a que a través de ellos, son interconectadas las redes y los protocolos de enrutamiento del tráfico son tradicionalmente ejecutados en los mismos.

Una de las características de un protocolo de enrutamiento, es su impacto en el rendimiento de extremo a extremo, adaptándose a cambios de topología cuando éstos ocurren. El factor de rendimiento que distingue a diferentes protocolos de enrutamiento es el **tiempo de convergencia** o tiempo de restablecimiento del enlace ante una falla o corte del mismo [1]. Cuanto menor es el **tiempo de convergencia**, se puede considerar que mejor es el protocolo de enrutamiento.

El objetivo de este trabajo es evaluar experimentalmente el **tiempo de convergencia** del protocolo OSPF en redes SDN en comparación con redes tradicionales, considerando diferentes escenarios de prueba. El resto de este trabajo está organizado de la siguiente manera: la Sección II, presenta una breve descripción del protocolo OSPF, en la Sección III, se presenta los trabajos relacionados más relevantes, mientras que la Sección IV describe el entorno experimental, propone la arquitectura y topología de ambos esquemas evaluados. La Sección V resume los resultados experimentales. Finalmente, las conclusiones y trabajos futuros son presentados en la Sección VI.

2 Protocolo *Open Shortest Path First* (OSPF)

El *Open Shortest Path First* (OSPF), es actualmente el protocolo de enrutamiento más utilizado en redes de datos, ya que ofrece simplicidad, adaptabilidad y buen rendimiento ante cambios de topología [3]. En las redes de datos tradicionales, los protocolos de enrutamiento se ejecutan en cada enrutador, donde se administran el plano de datos y el plano de control.

Cuando ocurren fallas de enlaces de red, la topología lógica existente en cada enrutador debe actualizarse, generando así tráfico de control que inunda la red, mediante los anuncios *Link State Announcement* (LSA). Esto ocurre constantemente de manera a mantener las tablas de rutas siempre actualizadas (ver Fig. 1).

Con la aparición de las *Software Defined Networks* (SDN) [10], el modelo de actualización de las tablas de rutas son más ágiles, ya que las SDN separan el plano de datos y el plano de control, gestionando ambos planos por separado.

Este nuevo modelo de gestión, incorpora al controlador**,** el nodo principal de la red, responsable de administrar y mantener comunicación constante con todos los conmutadores de la red, obteniendo así una visión centralizada y global de la red. [10].

Los diferentes tipos de mensajes que intercambian los enrutadores, son gestionados por el controlador, ya que trata estos mensajes como flujos de datos en una red SDN, cada enrutador necesita intercambiar información con otros enrutadores de manera a mantener actualizada la tabla de rutas, en las SDN éstos mensajes son captados por los conmutadores y lo envían al controlador. De ésta manera el controlador recibe todos los mensajes, actualiza la tabla de rutas y la topología lógica de la red.

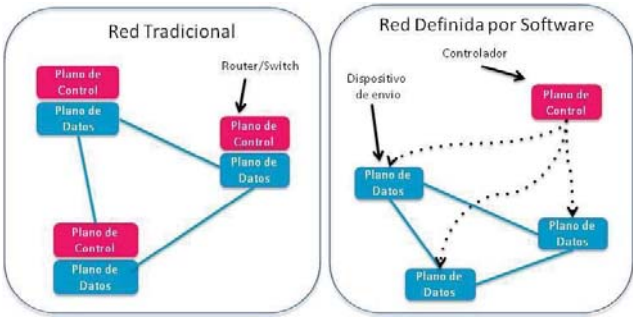

Fig. 1. Comparativo de Redes Tradicionales y SDN.

3 Trabajos Relacionados

En esta sección se presentan varias técnicas y métodos ejecutados para determinar el **tiempo de convergencia** y el rendimiento del protocolo OSPF en redes tradicionales y redes definidas por software los cuales son presentados en trabajos relacionados [1].

El **tiempo de convergencia** del protocolo OSPF es directamente influenciado por los tiempos de envío del paquete de saludo *(HELLO)* y el tiempo de cálculo del algoritmo de *Shortest Path First* (SPF), que son ejecutados en cada enrutador [1]. Estos parámetros pueden ser ajustados en cada enrutador de manera a obtener el menor tiempo posible de ejecución. Por ejemplo, el paquete de saludo *(HELLO)* es enviado periódicamente entre los enrutadores vecinos con la frecuencia establecida en el enrutador por el Intervalo de Saludo *(Hello-Interval)* [1]. Otros parámetros de rendimiento, pueden ser ajustados en el enrutador, de manera a mejorar el **tiempo de convergencia** del protocolo OSPF, teniendo en cuenta diferentes escenarios de fallas.

El enrutamiento en Redes Definidas por *Software* incluye un modelo de enrutamiento centralizado, el descubrimiento de la topología y el cálculo de las rutas son finalizados en el controlador [2]. El modelo de enrutamiento centralizado nos brinda mayor flexibilidad, debido a que nos ofrece condiciones más favorables para la optimización de la red [2]. La convergencia, representa el proceso de sincronización de las tablas de enrutamiento después de un cambio en la topología. La cantidad de tiempo es definido como “Intervalo” y es basado en el máximo tiempo esperado en la estabilización de la red después de un cambio en la topología.

La Ecuación (1) es aplicada a redes tradicionales y muestra que el tiempo de entrega de los mensajes LSA [2] y la ejecución del Algoritmo SPF [1], son factores claves en el **tiempo de convergencia** [1]. De acuerdo al tamaño de la red y retardo de los mensajes LSA [2], influyen directamente al **tiempo de convergencia** [1].

$$B + C + D + E = TCRT \quad (1)$$

dónde:
B = tiempo de descubrimiento del corte;
C = tiempo de entrega de mensajes LSA;
D = tiempo de ejecución del Algoritmo SPF;
E = tiempo de actualización de la tabla de rutas.
TCRT = tiempo de convergencia en Red Tradicional

La Ecuación (2) aplicada a redes SDN [10], muestra que el factor clave para el **tiempo de convergencia** [1], incluye el tiempo de actualización de la topología, el tiempo de ejecución del Algoritmo de SPF [1], y el tiempo de actualización de la tabla del flujo de rutas. Típicamente el controlador en las redes SDN [10], disponen de un enlace de alta velocidad con los conmutadores y esto mejora ampliamente el **tiempo de convergencia** [1].

La reproducción de video en tiempo real, exige de la red alto rendimiento y excelente calidad servicio QoS *(Quality of Service)*, éstas métricas son influenciadas por el retardo, la pérdida de paquetes y la fluctuación del retardo [3].

$$B + F + D + E = TCSDN \quad (2)$$

dónde:
F = tiempo de actualización de la topología.
TCSDN = tiempo de convergencia en SDN

En [1] se analiza el tiempo de convergencia de rutas en un entorno únicamente SDN, compuesto por *Mininet* y *Quagga*, con una topología *mesh* con 4 *routers* y 4 *hosts*. Se realizaron 10 experimentos por cada tipo de falla, optimizando algunos parámetros de rendimiento como el Intervalo de Saludo y el Tiempo de Ejecución del Algoritmo *SPF Delay*.

En [2] se analiza principalmente el tiempo de convergencia del protocolo OSPF, en redes tradicionales simuladas y redes SDN. Los entornos son simulados con varias máquinas virtuales VMWARE utilizando *Mininet* y *Quagga*. Disponen 2 topologías de red, una de 16 nodos y otra de 120 nodos, ambos con topología *mesh*. Se realizaron experimentos bajo diferentes parámetros de enlace, optimizando algunos parámetros de rendimiento del protocolo OSPF.

En [3] se analiza el tiempo de convergencia y calidad de servicios cuando se transmite tráfico multimedia sobre una red tradicional y una SDN. El entorno SDN está compuesto por *Mininet* y *Quagga*, con una topología en anillo, 4 máquinas físicas, interconectadas mediante túneles GRE, simulando máquinas virtuales en cada una de ellas, donde se tiene 4 *routers* y 6 *hosts*. El entorno tradicional, compuesto de *routers*, *switches* Cisco y PCs. Se realizaron experimentos a fin de determinar el rendimiento del protocolo OSPF en ambos entornos experimentales.

4 Entorno Experimental

En esta sección se describe la topología que se utilizarán en las pruebas experimentales, pero antes de describirlo, se expondrán tres subsecciones. La primera describe la arquitectura, plataforma y *software* utilizado para simular una red tradicional en un entorno virtualizado. La segunda describe la arquitectura de las redes SDN, mientras que la tercera describe la topología a ser simulada en ambos esquemas.

Las diferencias fundamentales de este entorno de trabajo, en comparación con los trabajos relacionados, son (1) topología de red y cantidad de nodos y (2) simulación de la red tradicional mediante virtualización de enrutadores, enlaces y nodos.

4.1 Simulación de Redes Tradicionales

De manera a contar con una infraestructura y plataforma que simule una red tradicional, se ha montado un servicio de simulación de redes virtuales Linux, sobre

plataforma Linux conocida como Virtual Network – User Mode Linux (VNUML) [8].

Esta plataforma simula un servidor Linux con las siguientes características: memoria base de 4Mb, memoria de video de 128 Mb, 2 discos virtuales de 16 Gb y de 8 Gb, soporte de red, audio y USB, con soporte VirtualBox. Utilizando ésta plataforma se ha construido el entorno virtual para simular una red tradicional, del cual obtenemos los siguientes beneficios: el usuario puede crear su propio escenario, varias máquinas virtuales pueden correr en una sola pc, compartición de configuraciones comunes entre máquinas virtuales, muy ligero y opera solo en modo consola.

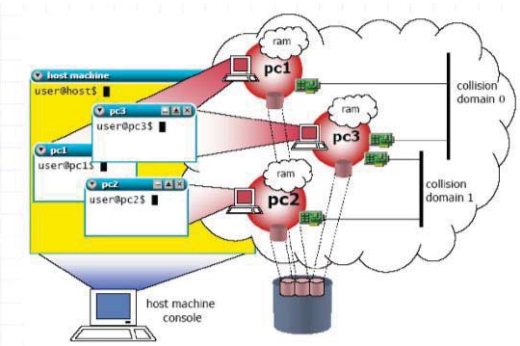

Fig. 2. Ejemplo de Simulación de Redes con VNUML.

Cada máquina virtual VNUML, puede desempeñar roles como host, enrutador entre otros, de manera a simular una red tradicional, ciertas máquinas virtuales deben ejecutar el rol host y otras como router. Asignar el rol de enrutador a una máquina virtual implica que esa máquina virtual deberá contar con procesos de enrutamiento, lectura, reenvío de paquetes, administración de mensajes de control, actualización de tablas de enrutamiento, entre otros, que son tareas inherentes a un enrutador en las redes tradicionales.

Quagga [6], es un conjunto de herramientas que permiten implementar en un sistema Linux los protocolos de enrutamiento (OSPF), (RIP) y (BGP), gestionando, para ello, la tabla de enrutamiento del propio núcleo del sistema.

Quagga [6] permite asignar el rol de enrutador, mediante la activación de demonios de enrutamiento y así poder intercambiar información de enrutamiento con otras máquinas del mismo rol. Quagga [6], puede configurar la interfaz de cada máquina virtual con direcciones, banderas, rutas estáticas y otros.

4.2 Redes Definidas por *Software*

En la Sección II se ha descrito como las SDN administran los mensajes del protocolo OSPF, en la Fig. 3 se puede ver la separación de las capas de la arquitectura SDN y su interconexión mediante una interfaz como **OpenFlow**, que comunica la capa de control con la capa de infraestructura. **OpenFlow**, es una interface de comunicaciones definidas entre la Capa de Control con la Capa de Infraestructura (Datos) en una arquitectura SDN. Admite el acceso directo a la manipulación de datos presente en los dispositivos por ejemplo, *switches*, enrutadores, entre otros [14].

OpenFlow utiliza el concepto de flujos para identificar el tráfico de red basado en reglas predefinidas que pueden ser programadas de forma estática o dinámica por el controlador SDN, lo cual permite definir como debe fluir el tráfico hacia los dispositivos de red [14].

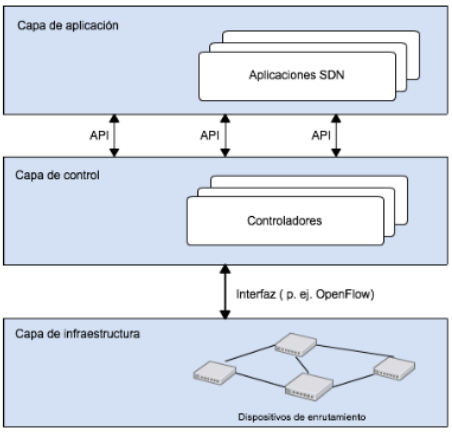

Fig. 3. Modelo de Arquitectura SDN.

4.3 Topología para Pruebas Experimentales

Se ha diseñado una topología que será simulada en ambos esquemas, presentados en las sub-secciones anteriores. A cada máquina virtual se han asignado roles específicos. Las máquinas virtuales con rol de router, utilizan el software de enrutamiento Quagga [6], y con el proceso activo OSPFD, de manera a utilizar el protocolo de enrutamiento dinámico OSPF.

En la simulación para redes tradicionales, cada máquina virtual desempeña un rol específico, el de *host* y *router*. La topología fue diseñada mediante el VNUML de Linux, con enlaces virtuales a los nodos vecinos [8]. Todos los *routers* están configurados en el área “0”. La Fig. 4 representa gráficamente la topología de la red tradicional.

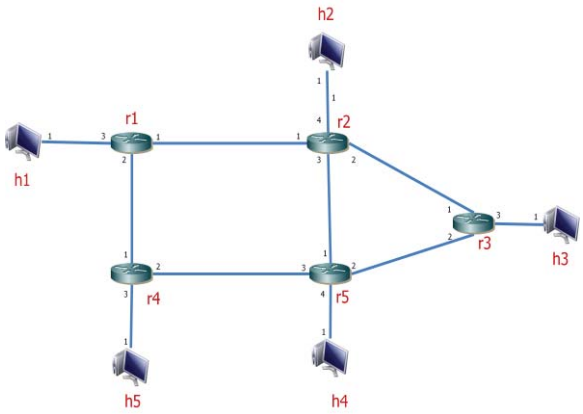

Fig. 4. Topología de red tradicional considerada en pruebas experimentales.

En la simulación para SDN, la topología fue diseñada con el software Mininet [4], que es una herramienta open-source de simulación de redes SDN, que permite crear host, switches, controladores y enlaces virtuales. Mininet corre sobre el estándar Linux, en este caso en particular, se ha instalado sobre Ubuntu v16.01, cada nodo de la red, es un switch virtual (Open vSwitch) y soportan **OpenFlow** y se han aplicado los mismos roles asignados en el modelo tradicional [14].

Todos los switches de la red son conectados al **controlador**, elemento central de las redes SDN. De manera a completar el entorno SDN [10], se ha utilizado una versión extendida de la herramienta Mininet denominada MiniNExT (Mininet ExTended) [12], que incluye una capa de extensión para facilitar la creación de redes complejas [4]. El motor de enrutamiento para cada nodo MiniNext es software de enrutamiento Quagga [6], en el cual se ha configurado y activado el proceso OSPFD de manera a utilizar el protocolo OSPF, similar al modelo tradicional.

Para la simulación en ambos esquemas, se ha implementado el protocolo IPv4, con dirección de red 10.10.0.0/24 y se han asignado direcciones IP a cada interface de la red propuesta.

5 Resultados Experimentales

En ésta sección se resumen los resultados de las pruebas experimentales, focalizadas en el **tiempo de convergencia** del protocolo OSPF, tanto en redes tradicionales como en redes SDN, bajo diferentes condiciones y parámetros que influyen en el rendimiento del protocolo OSPF.

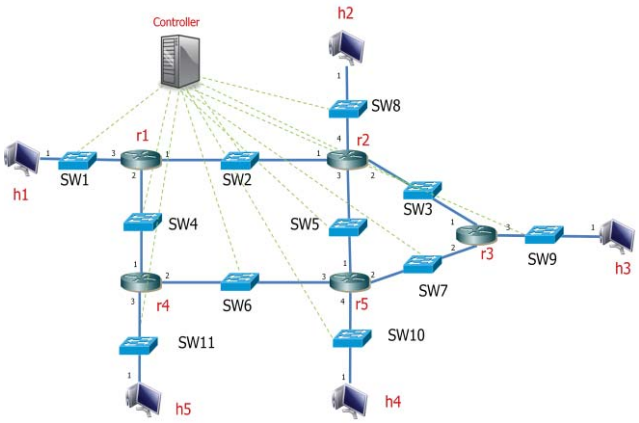

Fig. 5. Topología de la red SDN

5.1 Parámetros de rendimiento del protocolo OSPF

El protocolo OSPF es influenciado por diferentes parámetros de rendimiento, entre los cuales se pueden citar:

- **Intervalo de Saludo *(Hello Interval)*:** Cuando un enrutador inicia un proceso de enrutamiento OSPF envía un paquete de saludo *(Hello)* que contiene información acerca de las redes conectadas al enrutador y sigue enviando paquetes de saludo *(Hello)* a intervalos regulares. El valor por defecto del intervalo de saludo es de 10 segundos [1].
- **Algoritmo SPF:** El algoritmo SPF, representa una gran exigencia para la CPU del enrutador y el tiempo que lleva realizar los cálculos depende del tamaño del área. Si en la red hay un enlace inestable, que alterna entre los estados up y down, esto causa que los enrutadores OSPF de un área estén ejecutando constantemente el algoritmo para actualizar el árbol SPF, lo que dificulta una convergencia adecuada del enrutamiento. Esto se lo conoce como retardo del protocolo OSPF, *(SPF Delay)* [1].
- **Intervalo Muerto *(Dead Interval):*** determina si un vecino adyacente está desactivado. Por defecto, el intervalo muerto es de cuatro veces el valor del intervalo de saludo, por defecto 40 segundos [1].
- **Retardo de Transmisión *(Transmit Delay):*** es el tiempo estimado para enviar un paquete de actualización del estado del enlace (LSA), el valor por defecto es 1 segundo [1].
- **Intervalo de Retransmisión *(Retransmit Delay)*:** es el tiempo entre envíos de paquetes de actualización de del estado del enlace (LSA) entre enrutadores, el valor por defecto es de 5 segundos [1].

Parámetros	Falla Simple - 01 enlace cortado					Falla Doble - 02 enlaces cortados				
	Valor (A)	Valor (B)	Valor (C)	Valor (D)	Valor (E)	Valor (F)	Valor (G)	Valor (H)	Valor (I)	Valor (J)
Intervalo Hello < s >	10	8	6	4	3	10	7	5	4	3
SPF Delay										
Pérdida < ms >	400	200	100	80	50	400	150	80	50	30
Initial - Holdtime < ms >	600	400	300	200	100	600	300	200	120	80
Max - Holdtime < ms >	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000
Transmit Delay < 1 - 65530 > s	1	1	1	1	1	1	1	1	1	1
Dead Interval < 1 - 65530 > s	40	40	40	40	20	40	40	40	40	20
Retransmit Interval < 1 - 65530 >	5	5	5	5	5	5	5	5	5	5
Ancho de Banda < mBits >	10	10	10	10	10	10	10	10	10	10
Pérdida < ms >	0	0	0	0	0	0	0	0	0	0
Pérdida de Paquetes < % >	0	0	0	0	0	0	0	0	0	0

Tabla 1. Detalle de parámetros de rendimiento considerados.

En la tabla 1 vemos dos secciones en columnas que indican el tipo de falla o corte del enlace, para cada tipo de falla se ha modificado los mismos parámetros de rendimiento, de manera a verificar el comportamiento del protocolo, ante el tipo de falla y su rendimiento. Tenemos entonces desde la columna "Valor(A)" hasta la columna "Valor(E)" , indican los valores aplicados a los parámetros del protocolo OSPF que se utilizaron para realizar las pruebas de Falla Simple. Las siguientes columnas, desde "Valor(F)" hasta la columna "Valor(J)", indican los valores aplicados a los parámetros del protocolo OSPF que se utilizaron para realizar las pruebas de Falla Doble.

Falla Simple indica que existe un corte de enlace entre el origen y el destino.
Falla Doble indica que existen dos cortes del enlace entre origen y el destino.

5.2 Pruebas Experimentales

Todas las pruebas fueron realizadas utilizando el envío de paquetes ICMP de solicitud (ICMP echo request) y de respuesta (ICMP echo reply) de manera a comprobar el estado de la comunicación de cada host local con los demás host de la red. Antes de realizar la prueba se verificó el camino establecido por el protocolo, mediante el comando "traceroute". Luego se configuró cada enrutador según los parámetros de rendimiento indicados en la Tabla 1.

Posteriormente, se eligió aleatoriamente un enrutador ubicado en la ruta trazada desde el host origen al host destino; mediante línea de comandos del software Quagga se cortó el enlace, deshabilitando la interfaz del enrutador elegido. En ese momento se produce la convergencia de rutas del protocolo. Cada solicitud ICMP consistió en el envío de 64 bytes y secuencia de 20 envíos y cada test es guardado en un archivo de texto. El corte fue realizado en la secuencia ICMP 10 para falla simple y para falla doble, el corte fue realizado en la secuencia ICMP 8 y en la secuencia ICMP 15.

En la Tabla 2, se resumen resultados obtenidos, que incluyen en total 400 pruebas experimentales, compuestos de 200 pruebas de fallas simples (1 corte del enlace) y 200 pruebas de fallas dobles (2 cortes del enlace) para ambos esquemas (i.e. redes tradicionales y SDN). Los parámetros de rendimiento citados, influencian en el rendimiento y mejoran el **tiempo de convergencia** del protocolo OSPF.

Fallas del Enlace	Red Tradicional			Red SDN		
	Test hechos	ICMP Perdido	Tiempo Promedio (ms)	Test hechos	ICMP Perdido	Tiempo Promedio (ms)
Falla Simple - Parámetro (A)	20	8	0,904	20	2	0,138
Falla Simple - Parámetro (B)	20	1	1,013	20	0	0,120
Falla Simple - Parámetro (C)	20	1	1,008	20	0	0,222
Falla Simple - Parámetro (D)	20	1	1,622	20	2	0,244
Falla Simple - Parámetro (E)	20	2	1,996	20	1	0,456
Falla Doble- Parámetro (F)	20	16	0,752	20	1	0,205
Falla Doble- Parámetro (G)	20	2	1,098	20	0	0,165
Falla Doble- Parámetro (H)	20	3	0,958	20	1	0,208
Falla Doble- Parámetro (I)	20	7	0,517	20	0	0,141
Falla Doble- Parámetro (J)	20	2	0,500	20	0	0,103

Tabla 2. Resumen de resultados experimentales

6 Conclusiones y Trabajos Futuros

Este trabajo presenta algunos los parámetros que influencian en el rendimiento y la convergencia del protocolo OSPF. El **tiempo de convergencia** del protocolo OSPF [1], es influenciado por intervalos de tiempo del paquete de saludo *(HELLO)* y otros parámetros de rendimiento del protocolo, ante diferentes condiciones de fallas.

En las SDN, sin embrago, la congestión del tráfico por inundación de anuncios LSA, tráfico de control y mecanismo de enrutamiento, disminuyen debido a la incorporación del controlador.

Como se puede ver en la Tabla 2, la pérdida de paquetes ICMP en redes tradicionales totalizan 43 paquetes perdidos en las 200 pruebas realizadas, mientras que en las SDN totalizan solo 7 paquetes perdidos, en la misma cantidad de pruebas realizadas, lo que representa una disminución del 84% de paquetes perdidos.

Los tiempos de convergencia presentados en la Tabla 2, indican que en promedio, el tiempo de convergencia en las Redes Definidas por *Software* es hasta 82% menor ante fallas simples y hasta 78% ante fallas dobles, en ésta implementación.

Las SDN ofrecen mayor flexibilidad y alto rendimiento para adecuarse a eventuales cambios de topologías y repentinos que requieren las redes de datos actuales.

Como trabajo futuro se analizará el **tiempo de convergencia** de otros protocolos de enrutamiento como el BGP, con distintas topologías y cantidad de nodos, considerando ambos esquemas de redes tradicionales y SDN.

Referencias Bibliográficas

1. Cristina Loredana Duta, Laura Gheorghe, Nicolae Tapus "Analyse OSPF Convergence Time in the Presence of Single and Multiple Failures", EMERGIN 2015: The Seventh International Conference of Emerging Networks and Systems Intelligence, pp 3 –7, 2015.
2. Hailong Zhang, Jinyao Yan, "Performance comparison with Legacy Routing Protocol", 2015 International Conference on Cyber-Enabled Distribuited Computing and Knowledge Discovery, pp 2 – 4, 2015.
3. Albert Rego, Sandra Sendra, Jose Miguel Jimenez, Jaime Lloret, "OSPF Routing Protocol Performance in Software Defined Networks", 2017 Fourth International Conference on Software Defined Systems (SDS), pp 2 – 6, 2017.
4. Mininet, http://mininet.org.
5. RouteFlow, http://routeflow.github.io/RouteFlow.
6. Quagga Routing Suite, http://www.nongnu.org/quagga/.
7. Escuela Politécnica Federal de Lausana, Curso TCP/IP Networking, http://www.epfl.ch.
8. Virtual Network over Linux, http://web.dit.upm.es/vnumlwiki/index.php/Main_Page.
9. Chinmay Abhay Joglekar, "Route Manipulation using SDN in Quagga", Master Thesis, University of Colorado at Boulder
10. SDN Hub Tutorials, http://www.sdnhug.org
11. VirtualBox, http://www.virtualbox.org
12. MiniNExt, http://github.com/USC-NSL/miniNExT
13. Enrutamiento OSPF, https://sites.google.com/site/redeslocalesyglobales/4-configuracion-de-red/2-configuracion-de-routers/6-configuracion-del-encaminamiento/2-encaminamiento-dinamico/6-protocolo-ospf/4-problema-de-convergencia-en-ospf
14. Openflow, https://www.opennetworking.org/sdn-definition/.