



XVIII SIMPOSIO DE INGENIERÍA ELÉCTRICA (SIE-2019)

Título

Evaluación del desempeño y escalabilidad de controladores SDN distribuidos.

Title

Performance evaluation and scalability of distributed SDN controllers.

Daniel Iglesias de la Torre¹, Félix Álvarez Paliza², Julio César Pérez García³

1 –Daniel Iglesias de la Torre. UCLV, Cuba, danielit@uclv.cu

2 –Félix Álvarez Paliza. UCLV, Cuba, fapaliza@uclv.edu.cu

3 –Julio César Pérez García. UCLV, Cuba, juliocpg@uclv.cu

Resumen: Las Redes Definidas por Software son un paradigma emergente que puede superar las limitaciones de la infraestructura de red actual. La idea clave de SDN es separar la lógica de control de red de los dispositivos subyacentes que reenvían el tráfico.

Dada la naturaleza de misión crítica y a gran escala de las redes a nivel de operador, el plano de control de SDN debe ser diseñado de manera altamente disponible y escalable. Para este fin, más de un controlador SDN debe gestionarse como un grupo distribuido y varios problemas de escalabilidad, disponibilidad y recuperación ante fallas surgen. Este trabajo se centra en evaluar el desempeño de las redes SDN mediante grupos distribuidos de diferentes controladores. Para ello fueron analizados los fundamentos teóricos de este tipo de redes, los controladores disponibles en la industria y los criterios para su selección. Además, se hace un análisis de las herramientas de simulación, evaluación y desarrollo existentes y al mismo tiempo son identificadas las métricas de desempeño utilizadas en la comparación.

Palabras Clave: Redes Definidas por Software; Escalabilidad; Desempeño.

Abstract: *Software Defined Networks are an emerging paradigm that can overcome the limitations of today's network infrastructure. The key idea of SDN is to separate the network control logic from the underlying devices that forward the traffic.*



Given the mission-critical and large-scale nature of networks at the operator level, the SDN control plane must be designed in a highly available and scalable manner. For this purpose, more than one SDN controller must be managed as a distributed group and several problems of scalability, availability and failure recovery arise. This work focuses on evaluating the performance of SDN networks through distributed groups of different controllers. For this, the theoretical foundations of this type of networks, the controllers available in the industry and the criteria for their selection were analyzed. In addition, an analysis of the existing simulation, evaluation and development tools is made and at the same time the performance metrics used in the comparison are identified.

Keywords: *Software Defined Networks; Scalability; Performance.*

1. Introducción

Los sistemas de control de las redes IP tradicionales que se utilizan hoy en día son distribuidos, por lo que cada elemento de la red es una entidad separada con su propio plano de control y de reenvío. Tener un plano de control y reenvío en cada elemento de red significa que todos los dispositivos de la red toman su propia decisión al usar información que se comparte entre sí.

Los sistemas de control distribuidos tienen sus beneficios. Cada elemento de red es autónomo; por lo tanto, no hay un único punto de fallo. Además, este es un sistema maduro que se ha utilizado durante décadas y con el que todos los administradores de redes se han familiarizado. En una gran red tradicional con muchos dispositivos, los administradores de red deben configurar manualmente cada uno de estos dispositivos de red uno por uno. El proceso es tedioso y propenso a errores humanos [1]. Además, los diferentes productos de diversos fabricantes suelen tener diferentes interfaces de configuración. Por ejemplo, Cisco usa Cisco IOS, mientras que los productos Juniper tienen Junos OS. En algunos casos, los diferentes productos de un único proveedor tienen diferentes interfaces de configuración. Los administradores de red se ven obligados a aprender todas estas interfaces diferentes [2].

Para hacer frente a estos problemas surge el paradigma de Redes Definidas por Software (SDN, *Software Defined Networking*) [3]. Este es un nuevo concepto evolutivo para la arquitectura de red, que separa el plano de control del plano de datos. La separación ayuda a una mejor gestión de la red con un manejo eficiente del tráfico de red en diferentes planos de la arquitectura de las redes definidas por software (SDN). El plano de datos en SDN reenvía el tráfico de red basado en las instrucciones del plano de control. El controlador SDN construye la inteligencia de la red mediante la observación de las



entidades del plano de datos de reenvío y otros agentes SDN. Sin duda, el control centralizado ayuda a una mejor gestión de la red; sin embargo, se convierte en un cuello de botella cuando se trata de intercambio de grandes volúmenes de datos. Por otra parte, debido a la arquitectura centralizada del controlador, experimenta sobrecarga mientras aumenta el número de usuarios. En consecuencia, el controlador se convierte en un obstáculo para la correcta prestación del servicio. Si el propio controlador falla, el conmutador que había estado gestionando ya no se puede controlar. Por otra parte, el controlador SDN actúa como un único punto de fallo porque todas las decisiones de envío dependen directamente de él [4]. Una vez que el controlador SDN o el enlace conmutador-controlador fallan, toda la red puede colapsar.

La escalabilidad, la fiabilidad, la interoperabilidad y la tolerancia a fallos sigue siendo un reto en las arquitecturas de red centralizada. Sin embargo, el aspecto positivo de SDN es que es centralizado, pero altamente flexible y programable al mismo tiempo. Los aspectos de programabilidad de la red de SDN la hacen única. Por otra parte, las redes SDN soportan múltiples controladores SDN distribuidos y conectados entre sí. Estos sirven como controladores de copia de seguridad en el momento de un fallo. Por otra parte, varios controladores permiten la compartición de la carga cuando un único controlador está abrumado con numerosas peticiones de flujo. Además, varios controladores pueden reducir la latencia, aumentar la tolerancia a fallas y escalabilidad [5]. Sin embargo, el principal problema con este enfoque es el de mantener la consistencia entre los diversos controladores distribuidos. Las aplicaciones de red [6] serán tratadas de forma inadecuada por los controladores distribuidos a causa de la inconsistencia entre los controladores relativa a la visión global de los estados de la red [7]. Además, varios controladores pueden crear problemas de gestión de los recursos, incluyendo la distribución de estado, el intercambio de datos, la consistencia y el retardo de propagación entre varios controladores. De esta forma se puede ver limitado el tiempo de convergencia de la red, así como que se afecta la capacidad del controlador para responder a los diferentes eventos de red en un tiempo mínimo [8].

El presente artículo expone la importancia de la implementación de grupos de controladores SDN distribuidos para enfrentar los problemas de escalabilidad, fiabilidad, interoperabilidad y tolerancia ante fallas presentes en las arquitecturas centralizadas actuales. En virtud de ello se ha definido como objetivo, evaluar el desempeño de



controladores SDN con una propuesta de escalabilidad distribuida. Esta evaluación de desempeño se realiza en términos de razón de transferencia y latencia en el plano de control, tiempo de arranque de los conmutadores, tiempo de descubrimiento de topología y tiempo de recuperación ante fallos.

2. Escalabilidad en Redes Definidas por Software

La escalabilidad en el plano de control de SDN se puede establecer a través de dos enfoques generales. El primer enfoque está relacionado con la topología con subcategorías de diseños centralizados y enfoques distribuidos [9]. En esta categoría, se estudia la relación entre la topología de arquitecturas y problemas de escalabilidad. El enfoque distribuido son los diseños de controladores distribuidos, diseños de controladores jerárquicos y diseños híbridos. La reducción de la carga de trabajo en un controlador dará lugar a un mejor rendimiento del controlador con respecto a la escalabilidad. Por lo tanto, la distribución de la carga de trabajo del plano de control entre controladores está directamente relacionado con la escalabilidad. Los diseños híbridos representan los estudios que aprovechan el plano de datos delegando algunas funciones de control limitadas a los conmutadores para dividir la carga de trabajo del plano de control. Este enfoque es híbrido debido a la participación de ambos el plano de control y plano de datos en el control de la red. Este difiere con los diseños distribuidos y jerárquicos en la forma en que los conmutadores están involucrados en el procesamiento de decisión y control de la red.

En el segundo enfoque, se revisa la relación entre los diversos mecanismos utilizados para optimizar los controladores y los problemas de escalabilidad. La mejora de los controladores con respecto a su desempeño por algunas técnicas de optimización se traduce en un mejor rendimiento en escalabilidad. Además, la reducción de eventos resultantes del mecanismo de enrutamiento del controlador es otra manera de aumentar la escalabilidad en el plano de control dado que el proceso de enrutamiento produce una carga considerable al controlador. En la figura 1 se muestra la relación entre las diferentes clasificaciones.

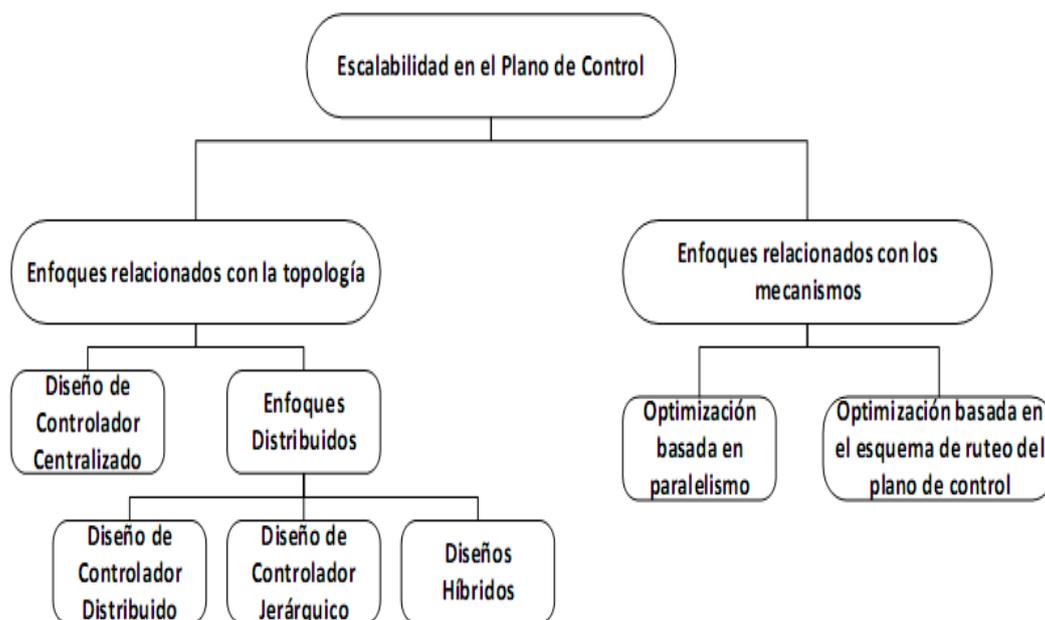


Figura 1: Relación entre los diferentes enfoques de escalabilidad en el plano de control, modificado de [10] .

2.1 Diseño de Controlador Distribuido

En esta estructura, cada controlador gestiona una subred / dominio de toda la red. Hay dos estrategias para arquitecturas de controlador distribuido para implementar la visión de red de controlador. En la estrategia de controlador local, cada controlador tiene su propia visión local de la red y cada una de sus redes locales vecinas se resumen como un nodo lógico. En la estrategia de visión global, por el contrario, cada controlador tiene una visión global de toda la red. En ambos casos, los controladores necesitan comunicarse a través de canales controlador-controlador para intercambiar información de estado con respecto a sus dominios.

2.2 Métricas para la evaluación de controladores SDN

En las redes SDN, el rendimiento del controlador es una de las principales preocupaciones para el diseño de redes más escalables. Hay muchos estudios que exploran el rendimiento de los diferentes controladores con respecto a la carga de trabajo de la red, implementaciones, arquitecturas y así sucesivamente [5], [11]. Aunque los estudios evalúan el desempeño de los controladores en cuanto a escalabilidad, se proponen en relación con diversos parámetros de rendimiento, tales como el tiempo de instalación de trayectoria, utilización del enlace, y así sucesivamente, dependiendo del objetivo del estudio. Las métricas más importantes y que son consideradas para la evaluación del



desempeño de escalabilidad, son la razón de transferencia exitosa del plano de control (*throughput*) refiriéndose al número de solicitudes de flujos manejados por segundo y la latencia de configuración de flujo (*delay*), que se refiere a la demora para responder las solicitudes de flujos, en el contexto SDN. Por lo tanto, el término escalabilidad o particularmente escalabilidad del plano de control en el contexto SDN, se caracteriza por las antes mencionadas métricas.

En una arquitectura SDN centralizada, la gestión de la red es comprometida en caso de fallas de la red. Esto se debe a que, cuando un conmutador o enlace falla en la red física, no solo afecta la comunicación entre él y el controlador, sino que también afecta a los conmutadores en la ruta de control. Mientras que, en el caso de una falla en el controlador, todos los dispositivos son desconectados del plano de control. Este problema de punto único de fallo es mitigado por los enfoques distribuidos, pero el proceso de detección y recuperación de falla afecta el tiempo de respuesta de la red para reestablecer a los flujos afectados y el tiempo de actualización de información de estado de la red en el plano de control. Por lo que el tiempo de recuperación ante fallos en el controlador (*failover time*) es una de las métricas de resiliencia más importantes en este tipo de arquitectura.

Además, para lograr mayor profundidad en la evaluación se considerarán otras métricas de desempeño de escalabilidad como el tiempo de arranque de los dispositivos (*boot up time*) y el tiempo de descubrimiento de topología (*topology discovery time*).

3. Resultados y discusión

Para la evaluación estableció un escenario basado en un grupo de controladores SDN distribuidos que consta de dos tipos diferentes de controladores. Cada controlador distribuido está configurado dentro de un grupo de tres nodos; los controladores en cada grupo están configurados en modo activo con uno de los controladores actuando como controlador maestro y los otros como esclavos. Este diseño se implementa usando los controladores ONOS [12, 13] y OpenDaylight [14, 15], el emulador de red Mininet [16] y las herramientas de evaluación de desempeño CBench [17] y de captura de tráfico Wireshark. El escenario fue configurado en servidores virtuales que fueron ejecutados en Ubuntu 16.04. Estas pruebas se realizan en varias fases, en las cuales se escala el número de conmutadores, con el objetivo de investigar cuál de estos controladores provee la más alta razón de transferencia en el plano de control, menor latencia en el plano de control,



menores tiempos de recuperación ante fallos en el controlador maestro y menor tiempo de arranque de los conmutadores y de descubrimiento de topología.

3.1 Resultados de las pruebas de razón de transferencia en el plano de control

En la figura 2, se representa la razón de transferencia en Flujos/segundo en el eje de las ordenadas y el número de conmutadores gestionados por el controlador en el eje de las abscisas, ambos en una escala lineal.

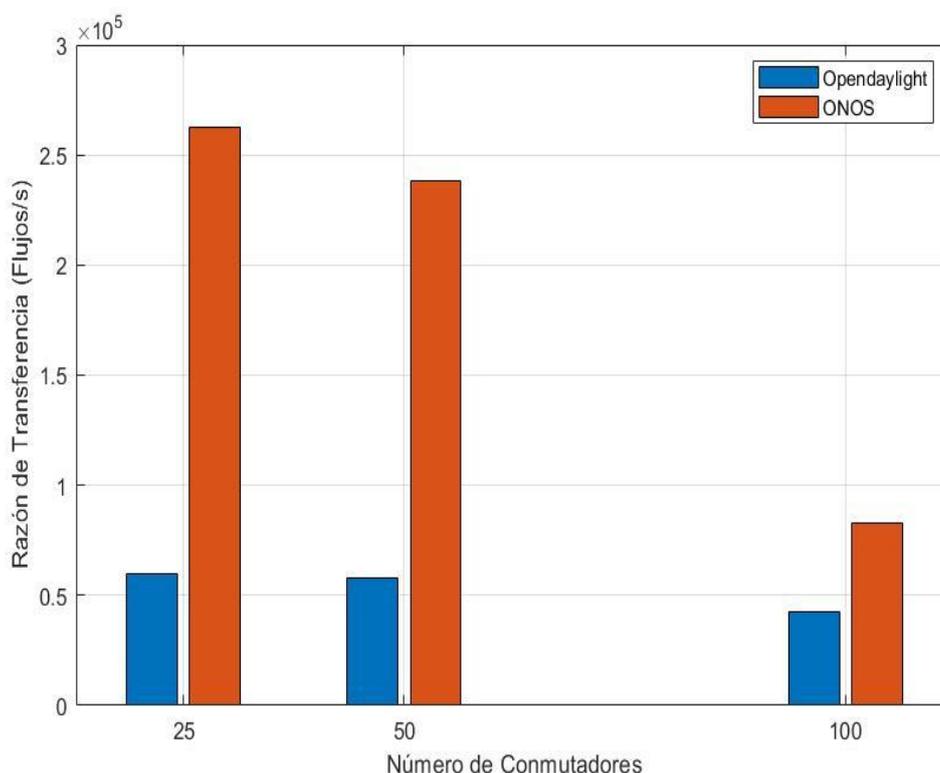


Figura 2: Razón de transferencia en el plano de control variando el número de conmutadores.

La figura anterior muestra que la mejor escalabilidad en cuanto a razón de transferencia la logra el controlador ONOS. ONOS presenta una razón de transferencia de 262450 Flujos/s con 25 conmutadores que disminuye ligeramente a 237980 Flujos/s con 50 conmutadores y a 82900 Flujos/s con 100 conmutadores. Aunque esta disminución entre los 25 y los 100 conmutadores es significativa, su alta razón de transferencia duplica en cada uno de los casos a las alcanzadas por el controlador Opendaylight con 59700 Flujos/s, 57940 Flujos/s y 42250 Flujos/s. Estos resultados se deben a que los



controladores ONOS y Opendaylight son controladores con capacidades de procesamiento multihilos,

3.2 Resultados de la prueba de latencia en el plano de control

En la figura 3, se representa la latencia de configuración de flujo en microsegundos en el eje de las ordenadas y el número de conmutadores gestionados por el controlador en el eje de las abscisas, ambos en una escala lineal.

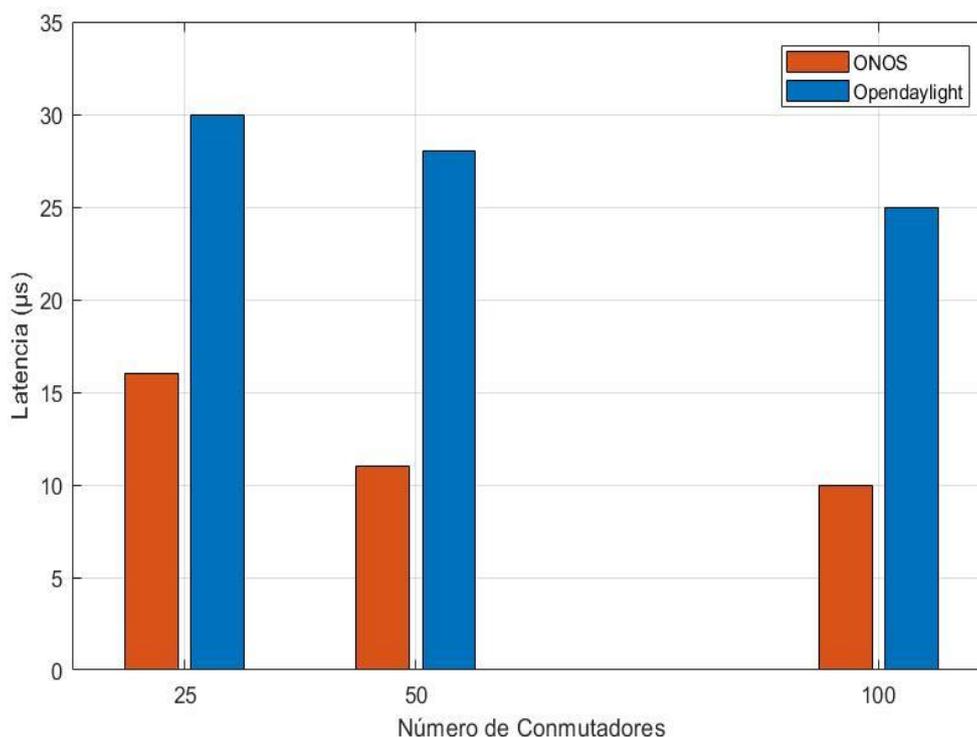


Figura 3: Latencia en el plano de control variando el número de conmutadores.

La figura 3 muestra que todos los controladores presentan la misma tendencia a la disminución de la latencia a medida que el número de conmutadores aumenta. La menor latencia de configuración de flujo la presenta el controlador ONOS que alcanza valores de 16 μs para 25 conmutadores, 11 μs para 50 conmutadores y 10 μs para 100 conmutadores. En el caso de Opendaylight se presentan latencias ligeramente mayores, con valores de 30 μs , 28 μs y 25 μs respectivamente. Este resultado se debe a que a medida que el número de conmutadores aumenta, la razón de mensajes *PACKET-IN* generados por Cbench aumenta. Como resultado, el controlador recibe una mayor



cantidad de mensajes *PACKET-IN* los cuales son procesados en paralelo. Esto a su vez resulta en una mayor cantidad de mensajes *PACKET-OUT* generados por el controlador, que subsecuentemente disminuye la latencia de configuración de flujo. Esto se debe a las funciones de dosificación de tareas (*Task Batching*) implementadas por cada controlador.

3.3 Resultados de la prueba de tiempo de arranque de los conmutadores

En la figura 4, se representa el tiempo de arranque de los conmutadores en milisegundos en el eje de las ordenadas y el número de conmutadores en el eje de las abscisas, ambos en una escala lineal.

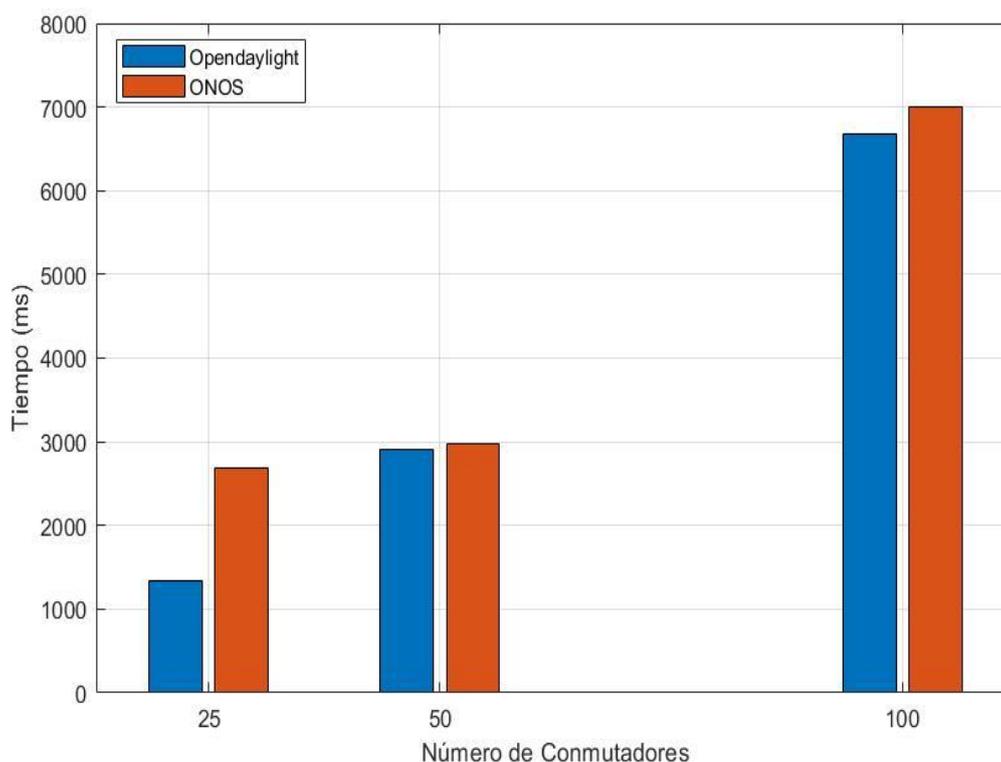


Figura 4: Tiempo de arranque de los conmutadores.

La figura 4 muestra que los controladores Opendaylight presenta un mejor desempeño en cuanto al tiempo de arranque de los conmutadores. Opendaylight alcanza tiempos de arranque de 1331 ms para 25 conmutadores, 2904 ms para 50 conmutadores y 6676 ms, superando en cada uno de los casos a los tiempos alcanzados por ONOS, que logra tiempos de 2693 ms, 2980 ms y 7008 ms respectivamente.



3.4 Resultados de la prueba del tiempo de descubrimiento de topología

En la figura 5, se representa el tiempo de descubrimiento de topología en milisegundos en el eje de las ordenadas y el número de conmutadores en el eje de las abscisas, ambos en una escala lineal.

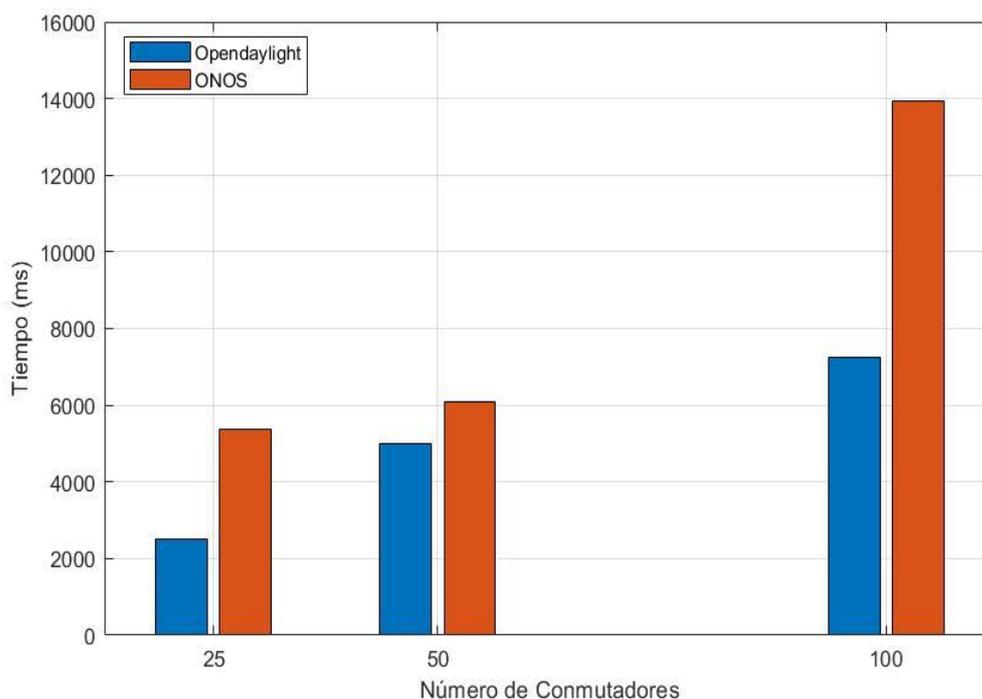


Figura 5: Tiempo de descubrimiento de topología.

La figura 5 muestra que el controlador Opendaylight presenta el mejor desempeño en cuanto al tiempo de descubrimiento de topología. Opendaylight solo necesita 2498 ms para descubrir una topología lineal de 25 conmutadores, 5011 ms para descubrir una topología lineal de 50 conmutadores y 7242 ms para descubrir una topología lineal de 100 conmutadores. Mientras que, el desempeño de ONOS se ve disminuido al necesitar 5380 ms, 6080 ms y 13924 ms respectivamente para cada una de las topologías.

3.5 Resultados de la prueba del tiempo de recuperación ante fallos

En la figura 6, se representa el tiempo de recuperación ante fallos en milisegundos en el eje de las ordenadas y el número de conmutadores gestionados por el controlador en el eje de las abscisas.

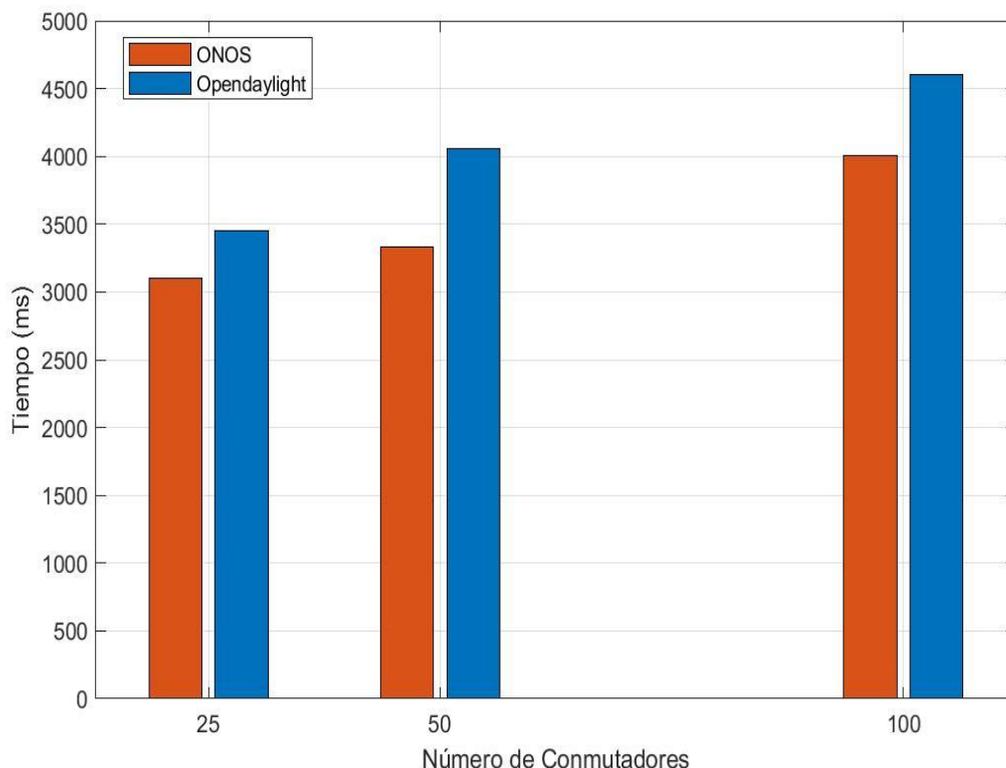


Figura 6: Tiempo de recuperación ante fallas.

La figura 6 muestra que la mejor resiliencia la alcanza el controlador ONOS. ONOS presenta tiempos de recuperación ante fallas de 3102 ms, 3331 ms y 4003 ms para 25, 50 y 100 conmutadores respectivamente. Estos resultados superan en cada caso a los obtenidos por el controlador Opendaylight con 3454 ms, 4056 ms y 4605 ms para 25, 50 y 100 conmutadores respectivamente.

4. Conclusiones

En el presente artículo se propone la implementación de grupos de controladores SDN distribuidos para enfrentar los problemas de escalabilidad, fiabilidad, interoperabilidad y tolerancia ante fallas presentes en las arquitecturas centralizadas. Para ello se realizó la evaluación del desempeño de los controladores ONOS y Opendaylight. Los resultados de los experimentos demostraron que el controlador ONOS presenta un mejor desempeño para la prueba de razón de transferencia en el plano de control que Opendaylight. En ambos controladores la razón de transferencia más alta se obtuvo para 25 conmutadores y disminuyó gradualmente para 50 y 100 conmutadores. ONOS alcanzó un máximo de



262450 Flujos/s, mientras Opendaylight solo alcanzó 59700 Flujos/s. En el caso de la latencia de configuración de flujos en el plano de control, en ambos casos se presentó la menor latencia para 100 conmutadores, dado que ambos controladores emplean funciones de dosificación de tareas. ONOS con 10 μ s tuvo mejor desempeño que Opendaylight con 25 μ s. En los casos del tiempo de arranque y del tiempo de descubrimiento de topología, fue el controlador Opendaylight quien mostró mejor desempeño. Aunque ambos obtienen sus menores tiempos para 25 conmutadores, ONOS duplica el tiempo de Opendaylight. Finalmente, el mejor desempeño en cuanto a resiliencia lo obtiene en cada caso el controlador ONOS. Los dos controladores presentan el menor tiempo de recuperación ante fallos para 25 conmutadores. La menor diferencia entre ambos es en este caso de 252 ms, mientras que para 50 conmutadores se obtiene la mayor diferencia de 725 ms.

5. Referencias bibliográficas

- [1] C. Doerr, R. Gavrilă, F. Kuipers, and P. Trimintzios, "Good practices in resilient internet interconnection," *ENISA Report*, Jun, 2012.
- [2] H. Krishna, "Providing end-to-end bandwidth guarantees with openflow," 2016.
- [3] S. Khan, A. Gani, A. W. A. Wahab, A. Abdelaziz, K. Ko, M. K. Khan, *et al.*, "Software-defined network forensics: Motivation, potential locations, requirements, and challenges," *IEEE Network*, vol. 30, pp. 6-13, 2016.
- [4] V. Yazici, M. O. Sunay, and A. O. Ercan, "Controlling a software-defined network via distributed controllers," *arXiv preprint arXiv:1401.7651*, 2014.
- [5] A. Tootoonchian, S. Gorbunov, Y. Ganjali, M. Casado, and R. Sherwood, "On Controller Performance in Software-Defined Networks," *Hot-ICE*, vol. 12, pp. 1-6, 2012.
- [6] M. Yang, Y. Li, D. Jin, L. Su, and L. Zeng, "Opportunistic spectrum sharing in software defined wireless network," *Journal of Systems Engineering and Electronics*, vol. 25, pp. 934-941, 2014.
- [7] S. Azodolmolky, *Software defined networking with OpenFlow*: Packt Publishing Ltd, 2013.
- [8] A. Abdelaziz, A. T. Fong, A. Gani, U. Garba, S. Khan, A. Akhuzada, *et al.*, "Distributed controller clustering in software defined networks," *PloS one*, vol. 12, p. e0174715, 2017.
- [9] O. Bliat, M. Ben Mamoun, and R. Benaini, "An overview on SDN architectures with multiple controllers," *Journal of Computer Networks and Communications*, vol. 2016, 2016.
- [10] M. Karakus and A. Duresi, "A survey: Control plane scalability issues and approaches in software-defined networking (SDN)," *Computer Networks*, vol. 112, pp. 279-293, 2017.



II Convención Científica Internacional 2019
Universidad Central "Marta Abreu" de Las Villas
CIENCIA, TECNOLOGÍA Y SOCIEDAD. PERSPECTIVAS Y RETOS

- [11] P. Isaia and L. Guan, "Performance benchmarking of SDN experimental platforms," in *NetSoft Conference and Workshops (NetSoft), 2016 IEEE*, 2016, pp. 116-120.
- [12] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, *et al.*, "ONOS: towards an open, distributed SDN OS," in *Proceedings of the third workshop on Hot topics in software defined networking*, 2014, pp. 1-6.
- [13] A. S. Muqaddas, A. Bianco, P. Giaccone, and G. Maier, "Inter-controller traffic in ONOS clusters for SDN networks," in *Communications (ICC), 2016 IEEE International Conference on*, 2016, pp. 1-6.
- [14] A. Hemid, "Facilitation of the OpenDaylight Architecture," *Accessed on*, vol. 3, p. 18, 2016.
- [15] J. Medved, R. Varga, A. Tkacik, and K. Gray, "Opendaylight: Towards a model-driven sdn controller architecture," in *2014 IEEE 15th International Symposium on*, 2014, pp. 1-6.
- [16] T.-Y. Huang, V. Jeyakumar, B. Lantz, N. Feamster, K. Winstein, and A. Sivaraman, "Teaching computer networking with mininet," in *ACM SIGCOMM*, 2014.
- [17] R. Sherwood and Y. KOK-KIONG, "Cbench: an open-flow controller benchmarker," URL <http://archive.openflow.org/wk/index.php/Oflops>, 2010.