



**UNIVERSIDAD POLITÉCNICA SALESIANA  
SEDE QUITO  
CARRERA DE INGENIERÍA ELECTRÓNICA**

**COMPARACIÓN DE UNA RED CONVENCIONAL CON UNA RED DEFINIDA POR  
SOFTWARE (SDN)**

Trabajo de titulación previo a la obtención del  
Título de Ingeniero Electrónico

AUTORES: Andrés Orlando Quishpe Sánchez  
Bryan Daniel Toapanta Rodríguez

TUTOR: Luis Germán Oñate Cadena

Quito-Ecuador

2022

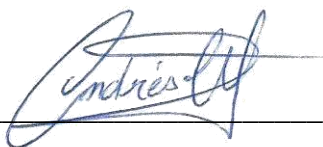
## CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Andrés Orlando Quishpe Sánchez, con documento de identificación No. 1723306054 y Bryan Daniel Toapanta Rodríguez con documento de identificación No. 1725445454; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 28 de julio del año 2022.

Atentamente,



---

Andrés Orlando Quishpe Sánchez  
1723306054



---

Bryan Daniel Toapanta Rodríguez  
1725445454

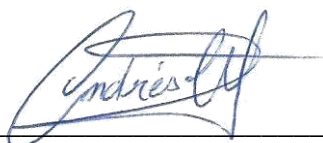
## **CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Nosotros, Andrés Orlando Quishpe Sánchez, con documento de identificación No. 1723306054 y Bryan Daniel Toapanta Rodríguez con documento de identificación No. 1725445454, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Artículo Académico: “Comparación de una red convencional con una red definida por software (SDN)”, el cual ha sido desarrollado para optar por el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana

Quito, 28 de julio del año 2022.

Atentamente,



Andrés Orlando Quishpe Sánchez

1723306054



Bryan Daniel Toapanta Rodríguez

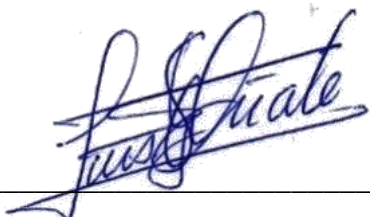
1725445454

## CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Luis Germán Oñate Cadena con documento de identificación No. 1712157401 docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: COMPARACIÓN DE UNA RED CONVENCIONAL CON UNA RED DEFINIDA POR SOFTWARE (SDN), realizado por Andrés Orlando Quishpe Sánchez con documento de identificación No. 1723306054 y por Bryan Daniel Toapanta Rodríguez con documento de identificación No. 1725445454, obteniendo como resultado final el trabajo de titulación bajo la opción Artículo Académico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 28 de julio del año 2022.

Atentamente,



Ing. Luis Germán Oñate Cadena MSc.

1712157401

## **DEDICATORIA**

Inicio dedicando este trabajo de titulación a Dios por brindarme salud y fortaleza para culminar con éxito mi carrera, a mis padres Orlando Quishpe y Cecilia Sánchez por ser siempre el mejor ejemplo. Papá, gracias por ser el que siempre creyó en mí, por tener el mejor consejo y por brindarme la mano cuando más lo necesite, por enseñarme lo verdaderamente importante y valioso. Mamá, que con tu amor nunca dejaste que me rindiera, que siempre me guiaste desde mis primeras letras para formarme como un hombre de bien y porque me enseñaste que todo esfuerzo tiene su recompensa.

Andrés Orlando Quishpe Sánchez

A mis adorados padres, quienes me han dado todo su amor y me han apoyado siempre en todo objetivo planteado a lo largo de mi vida, quienes me han forjado con reglas y también con ciertas libertades, por enseñarme a crecer y a que si caigo debo levantarme, por apoyarme y guiarme. Madre mía, fuiste, eres y serás un pilar fundamental, siempre tienes palabras de aliento, jamás falta un buen consejo, mi admiración para ti por haber logrado tanto en el ámbito personal, así como en el ámbito profesional. Padre querido, nuestro gran apoyo, quien siempre me dio la confianza para seguir adelante y que no pierde ocasión para recordarme que confía en mí y en mis capacidades.

Bryan Daniel Toapanta Rodríguez

## **AGRADECIMIENTO**

Agradezco a la Universidad Politécnica Salesiana por haberme dado la oportunidad de formar parte de sus alumnos. Agradecer a mis profesores en especial al Ing. Luis Oñate que con su conocimiento guio nuestro trabajo de titulación y a mis amigos Luis, Daniel, Bryan, por el apoyo y paciencia.

Andrés Orlando Quishpe Sánchez

"No podemos tener una actitud de amor sin ser agradecidos por ello"

Gracias a Dios por darme esta maravillosa familia, a mis padres, por su apoyo incondicional, a la Universidad Politécnica Salesiana, por darme la oportunidad de vivir grandes experiencias dentro de sus aulas, gracias a cada profesor, por ser parte de esta formación integral, sobre todo al Ing. Luis Oñate quien nos acogió y guio para dar este último y a la vez importante paso en nuestra vida estudiantil, a mis amigos Jorge, Óscar, Douglas, Daniel por su apoyo, gracias a mi amigo y compañero de proyecto Andrés, por su ayuda y soporte en muchas ocasiones difíciles y por último a Stefany por su preocupación para llegar a culminar esta etapa de mi vida y motivación en su momento.

Bryan Daniel Toapanta Rodríguez

# Comparación de una red convencional con una red definida por software (SDN)

Andrés Orlando Quishpe Sánchez  
Ingeniería Electrónica Y  
Telecomunicaciones  
Universidad Politécnica Salesiana  
Quito, Ecuador  
aquizhpe@est.ups.edu.ec

Bryan Daniel Toapanta Rodríguez  
Ingeniería Electrónica Y  
Telecomunicaciones  
Universidad Politécnica Salesiana  
Quito, Ecuador  
btoapantar@est.ups.edu.ec

Luis Germán Oñate Cadena  
Ingeniería Electrónica Y  
Telecomunicaciones  
Universidad Politécnica Salesiana  
Quito, Ecuador  
lonatec@ups.edu.ec

**Resumen** – Si bien en un inicio la gestión de redes convencionales podía ser cubierta con una pequeña demanda de administradores, su crecimiento paulatino ha dado lugar a que presenten inconvenientes como la falta de flexibilidad con respecto a la programabilidad y a la gestión de sus elementos. Teniendo en cuenta que las redes definidas por software o también conocidas como Software Defined Networking (SDN) nacen como un modelo de conocimiento de estudio que se enfocan de una manera diferente en la gestión y control de redes frente al actual funcionamiento de las redes convencionales. Este artículo se enfoca en comparar una red convencional frente a una red SDN utilizando como medios de emulación los softwares GNS3 y Mininet, tomando como métricas de comparación: la latencia, pérdida de paquetes y throughput, para ello se ha establecido un escenario de red de pruebas con enlaces redundantes y varios hosts. Al analizar los datos arrojados por las diferentes pruebas se pudo evidenciar un mejor desempeño de la red definida por software (SDN), ya que presentó un retardo promedio de un 82,32% menor a la red convencional, determinando así la eficiencia de las redes SDN frente a una red convencional.

**Palabras clave** - Redes convencionales, Redes SDN, GNS3, Mininet, pérdida de paquetes, latencia, throughput.

**Abstract** - Although at the beginning the management of conventional networks could be covered with a small demand for administrators, its gradual growth has given rise to drawbacks such as the lack of flexibility concerning programmability and the management of its elements. Taking into account that networks defined by software or also known as Software Defined Networking (SDN) were born as a study knowledge model that focuses differently on the management and control of networks compared to the current operation of conventional networks. This article focuses on comparing a conventional network against an SDN network using GNS3 and Mininet software as emulation means, taking as comparison metrics: latency, packet loss, and throughput, for which a network scenario of tests with redundant links and multiple hosts. When analyzing the data produced by the different tests, it was possible to show a better performance of the software-defined network (SDN), since it presented an average delay of 82.32% less than the conventional network, thus determining the efficiency of the networks. SDN versus a conventional network.

**Keywords** - Conventional networks, SDN networks, GNS3, Mininet, packet loss, latency, throughput.

## I. INTRODUCCIÓN

El modo de gestión que se emplea en las redes de comunicación se ha visto afectado por el enorme crecimiento que se da en los últimos años, para el año 2017 se contaba alrededor de 4 mil millones de conexiones [1]. Donde la manera de recolectar, transportar, almacenar y procesar la información [2], se convierte en una tarea difícil de cumplir por la gran cantidad de dispositivos que forman parte de la red, a esto se suma que la red debe atender a más dispositivos que aumentan de manera constante.

El objetivo de este artículo fue realizar la experimentación sobre la eficiencia de una red definida por software SDN en relación con una red convencional con equipos de hardware basándose en la pérdida de paquetes, latencia y throughput, con el propósito de determinar cuál de los tipos de redes es más eficiente al momento de converger. En lo que se refiere a la arquitectura de las redes definidas por software, se divide en plano de datos y plano de control, como consecuencia, la administración de estas redes no se lleva a cabo en cada uno de los equipos *in situ* [3], como lo es en el modelo de red convencional, en donde se considera a la infraestructura como un conjunto de elementos independientes y a su vez relacionados entre sí, que realizan la transferencia de datos entre ellos aplicando protocolos y reglas que estarán determinadas en la mayoría de los casos por el hardware [4]. Mininet es una herramienta que surge como el resultado del trabajo en conjunto de varios expertos, con el fin de proporcionar un medio capaz de crear prototipos de redes definidas por software SDN [5] en este entorno se encuentra una colección de controladores, enlaces virtuales, host y switches [6], entre los beneficios que brinda esta herramienta se encuentra la representación de topologías, desde las más simples hasta las más complejas y la ejecución de controladores externos de manera local o remota [5]. Las redes SDN se han convertido en la solución en empresas por su bajo costo de operación, resolver problemas y ser adaptivas dependiendo de la demanda [7].

Este artículo se compone de siete secciones que permiten llegar a determinar que red es más idónea, la sección II describe a las redes definidas por software, la sección III describe el

entorno de simulación, la sección IV muestra la comparación de las redes SDN y la red convencional, la sección V muestra el análisis y resultados obtenidos en Mininet y GNS3 y en la sección VI se presenta las conclusiones obtenidas de la investigación.

## II. REDES DEFINIDAS POR SOFTWARE

### A. Redes definidas por Software (SDN)

El beneficio que tienen los administradores de red al usar redes definidas por software es el controlar y programar desde un solo dispositivo toda la red. El modelo de las redes (SDN) se basa en tres planos: aplicación, control y datos, como se puede ver en la figura 1. La Arquitectura tiene como fin desagregar los planos de control y datos en dispositivos de red como enrutadores y conmutadores[1].

Este tipo de red tiene como propósito incorporar ciertas facilidades como: un diseño simplificado y un control optimizado en contra posición a una red tradicional. Partiendo de este hecho se puede apreciar la autonomía del software con respecto al fabricante, además de la disminución de equipos ya que no es necesario la configuración de distintos protocolos, básicamente su funcionamiento es en base a ciertas instrucciones por parte del controlador[8].

Cabe destacar que, aunque su funcionamiento no este asociado fundamentalmente a ciertos equipos físicos, es imprescindible disponer de un firmware con Open Flow en los dispositivos de red.

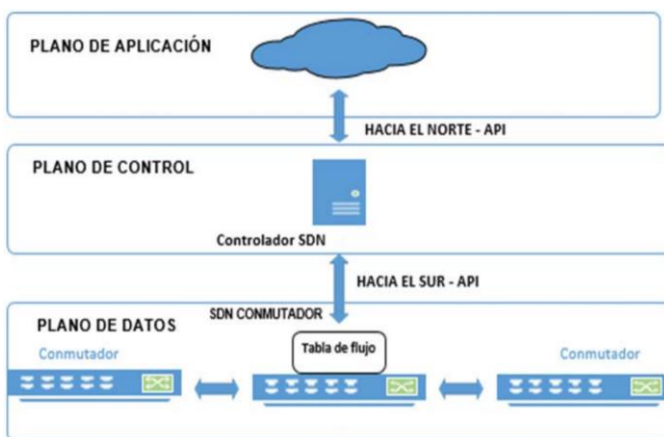


Fig. 1. Descripción general de la arquitectura SDN [9].

### B. Arquitectura de las redes definidas por software

1) *Plano de Aplicación:* Las aplicaciones de esta capa pueden considerarse el cerebro de la red debido a que son las encargadas de establecer la lógica de control [10]. Además, permiten establecer comunicación entre el plano de aplicación

y plano de control, de esa manera se puede prestar servicios como: enrutamiento, ingeniería de tráfico, multidifusión, seguridad, control de acceso, gestión en el ancho de banda y calidad de servicio (QoS) [11].

2) *Plano de Control:* Por medio de este plano se obtiene una vista centralizada de la topología, es el lugar donde se aloja el controlador, siendo este el componente central de la red. Toma la decisión de cómo los paquetes son enviados por uno o más elementos[10]. Además, la arquitectura no ve ningún valor en la proliferación de bloques y no todas las responsabilidades del controlador SDN pueden ser asignadas a componentes funcionales específicos [12].

3) *Plano de datos o infraestructura:* Formado por un conjunto de equipos, mismos que forman parte de una red convencional, con la diferencia que en redes SDN dichos equipos se encargan del reenvío de paquetes. La inteligencia de la red forma parte de un control lógicamente centralizado llamado Sistema Operativo de Red o NOS por sus siglas en inglés [10]

## III. ENTORNO DE SIMULACIÓN

### A. Mininet

Mininet es un emulador de gran flexibilidad que permite la creación de nuevas topologías con nuevas características de software. Para la simulación de los elementos de una red SDN: controlador, hosts y switches OpenFlow. Mininet utiliza el núcleo del sistema operativo de Linux [3].

Es importante destacar que Mininet incorpora un controlador con distribución OpenFlow el mismo que sirve como referencia pudiéndolo comparar con un switch clásico [5], de esta manera se logra separar a los planos de control y datos.

### B. Controlador POX

POX es un controlador que usa Python como lenguaje de programación que nace a partir del controlado NOX, con el fin de cubrir los requerimientos de las redes SDN usando Python en Windows, Mac o Linux[13]. Debido a que la presente investigación requirió la lectura de ciertas métricas para evaluar el rendimiento de la red, se decidió usar el controlador POX, ya que presenta facilidades al momento de implementar ciertas funcionalidades que permitían la optimización de la red. Además, que esta herramienta dispone de un banco de datos en donde se puede hallar explicaciones y funcionalidades de sus diferentes módulos, permitiendo una comprobación rápida de la red.

Los componentes utilizados por el controlador POX [14] [15] deben ser invocados al momento de iniciar con el controlador, estos componentes son programados en Python y cumplen con funciones propias como: la memorización de las direcciones



MAC, controlar los bucles que presenta la red así también como la activación o desactivación de puertos para evitar las alteraciones al momento de controlar un ciclo y la identificación de la topología de red a partir del envío de mensajes LLDP.

### C. OpenFlow

OpenFlow es un protocolo abierto que tiene como fin operar redes SDN, constituyéndose un elemento esencial en la arquitectura de estas redes. Permite programar hardware ya sean físicos o virtuales debido a que separa el plano de datos del plano de control originando mayor eficiencia [16]. El tipo de control es granular, ocasionando una tendencia a ejecutar cambios dinámicos dentro de la red tienda a ejecutar cambios dinámicos, sea en aplicación, usuario y sesión [17]. El mecanismo de operación radica en el uso de flujos para identificar el tráfico en la red, mismos que se encuentran predefinidas de forma estática o dinámica en el software de control SDN [17].

## IV. COMPARACIÓN DE UNA RED CONVENCIONAL Y UNA RED SDN

### A. Topología en Mininet

Para la presente investigación se planteó como entorno de prueba una red con enlaces redundantes, como se puede observar en la figura 2. La red SDN está compuesta por los siguientes dispositivos: 5 Open vSwitches 2.13.1, 6 hosts virtuales (H 1- H 2- H 3- H 4- H 5- H 6), 1 Controlador POX.

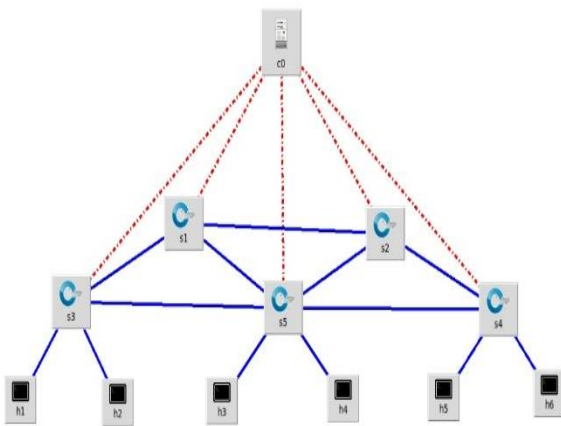


Fig. 2. Topología implementada.

### A. Simulación de la red SDN

Para la simulación de la red SDN se hizo uso de la herramienta Mininet la cual trabaja en un entorno de Linux simulado dentro de una máquina virtual (Virtual Box). Para tener una mejor interacción entre la máquina host y la máquina virtual, fue necesario la instalación del software VcXsrv en conjunto con el

software PuTTY por medio de SSH. Generalmente, la instalación de un Sistema Operativo que se basa en Linux proporciona un grupo de interfaces así como una tabla de enrutamiento[18]. Debido a ello, las entradas de las tablas pueden ser modificadas, así como habilitar o deshabilitar las entradas de las interfaces, sin embargo, todas son utilizadas por el sistema operativo.

Los Network namespaces son útiles para configurar entornos virtuales. Permitiendo que cada instancias o espacios de los dispositivos se encuentren separados contando así con sus propias interfaces de red y tablas de enrutamiento logrando un trabajo independiente [18]. Estas características de los Network namespaces son un punto clave para el desarrollo de esta investigación, ya que permite emular los diferentes procesos de red de los open vSwitches sin estar restringidos a un número en específico, así como todos los hosts virtuales teniendo un comportamiento diferenciado dentro de una misma máquina virtual por medio de interfaces de red virtual. De forma predeterminada un Root Network namespace se encuentra incluido dentro de cada máquina Linux el cual contiene la operatividad funcional de red de la máquina. Todos los procesos que se ejecutan en ella comparten las mismas funcionalidades [18].

El funcionamiento de conexión se basa en que cada extremo de una interfaz de red virtual tendrá un Network namespace diferenciado, en pocas palabras las tramas recibidas son enviadas de un extremo al otro y viceversa.

Para el caso de estudio se implementó un controlador POX por ello al momento de la redacción del código en Python se especificó los siguientes parámetros del controlador:

- IP del controlador: 127.0.0.1
- Puerto de comunicación: 6633 puerto de comunicación estándar TCP (switch y controlador)
- Tipo de controlador: remoto

Dentro del código de la red se declaró la dirección IP de los diferentes hosts, así como el ancho de banda, esto debido a que Mininet por defecto trabaja en un ancho de banda del orden de los Gbits/sec, por ello se realizó un primer flujo de datos TCP en GNS3.

1) *Latencia:* El valor de latencia se obtuvo realizando un ping entre todos los hosts de la topología como se muestra en la figura 2. Para obtener el valor promedio de latencia de la red se planteó realizar un envío de 60 paquetes desde el host 1 hacia los demás hosts, luego desde el host 2 con todos los demás, incluido el host 1, este proceso se repitió para el resto de los hosts, de los datos recopilados se obtuvo los valores RTT promedios los cuales son: mínimo, máximo, average o promedio todos medidos en milisegundo (ms).

2) *Perdida de paquetes*: Cuando se habla de pérdida de paquetes en un sistema simulado con una conexión ya establecida, la entrega de paquetes será de un 100 por ciento, por lo que para determinar un porcentaje de pérdida de paquetes se realizó una desconexión en el enlace que conecta los switch S1-S4.

3) *Throughput*: Para la determinación del throughput se utilizó la herramienta iPerf, misma que ayuda para mediciones de ancho de banda en redes IP. Al mismo tiempo, se puede ajustar los parámetros relacionados con la sincronización, búferes y protocolos (TCP, UDP) con IPv4 [19]. Para la presente investigación se realizó la medición del throughput tanto para protocolo TCP como para UDP, para ello se configuró el host1 como servidor y el host6 como cliente, enviando 60 paquetes en intervalos de 1 segundo para ambos protocolos.

### B. Simulación redes convencionales

Para la implementación de la red en GNS3 se instalaron dos máquinas virtuales con el SO Ubuntu, siendo estas asignadas al host1 y al host6, es decir se ubicaron en los extremos de la red propuesta, esta implementación fue necesaria para tener acceso a las herramientas de medición de ping y throughput descritas en la sección IV inciso B. Como se puede observar en la figura 3. La red convencional está compuesta por los siguientes dispositivos: 6 switches capa 3 modelo “c3725, adventerprisek9-mz.124-15. T14.image”, 5 switch funcionando como switch (ESW 1-ESW 5) y el ESW 6 como router, 6 hosts (PC 2 – PC 5) hosts normales, host 1 y host 6 máquinas virtuales con Sistema operativo ubuntu-20.04.3 de 64 bits.

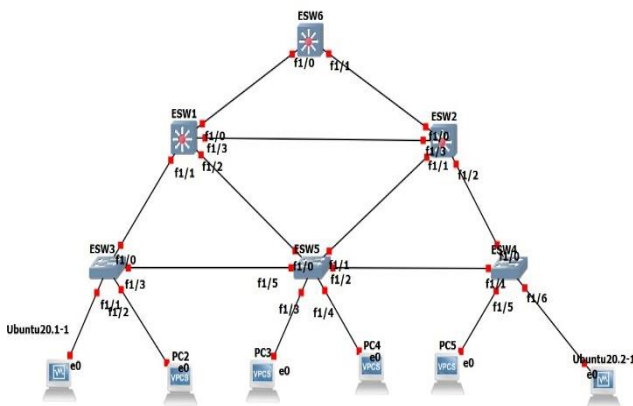


Fig. 3. Topología implementada en GNS3.

Para la red convencional simulada en GNS3 se utilizó Switches de capa3, la configuración se realizó implementado VLANs, además, del protocolo VTP el cual se configuró como modo servidor en el switch 6 permitiendo que las VLANs se copien en los demás switches que estarán en modo cliente, en las conexiones entre switches se configuró el protocolo STP y para

la comunicación entre los diferentes hosts se configuró un enrutamiento InterVLAN en el switch6.

Para determinar los valores de latencia, throughput, y pérdida de paquetes se realizó el mismo procedimiento de medición implementado en Mininet.

## V. ANÁLISIS Y RESULTADOS

### A. Latencia

Como primer parámetro de medida del rendimiento de una red se ha estudiado la latencia o retardo, dicho parámetro es la suma de retardos que aparecen en la red tras el establecimiento de los flujos de paquetes, es decir, el tiempo de retardo de los paquetes en llegar a su destino, para este caso en particular se necesitó conocer el retardo generado por cada una de las ramas.

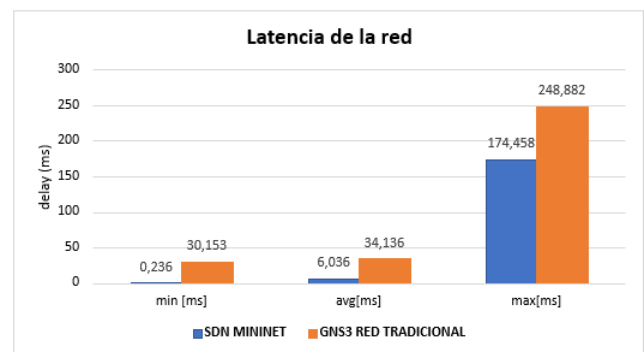


Fig. 4. Comparación de valores promedios de latencia.

Para la simulación en Mininet y GNS3 se obtuvo los resultados presentados en la figura 4, como punto de comparación se tomó el valor promedio AVG (ms) obteniendo un resultado de latencia para la red SDN de 6,036 (ms) en comparación con la latencia de una red convencional de 34,136 (ms), lo cual representa una reducción de latencia en un 82,32% observando un resultado óptimo de la red SDN en comparación de una red convencional.

### B. Perdida de paquetes

Para establecer la pérdida de paquetes dentro de las redes en cuestión se procedió a la desconexión de un enlace, con el fin de observar como la red actúa al momento de encontrar una nueva ruta, pasado el tiempo de espera muchos paquetes son eliminados y esto ocasiona que no lleguen a su destino. En la figura 5 se puede apreciar que en las redes SDN, al desconectar un enlace, el porcentaje de pérdidas de paquetes es de un 18,33% a diferencia con la red convencional, que llega a perder un 75% de los paquetes. Una vez restablecido el enlace se pudo observar que la red SDN ya no genera pérdidas

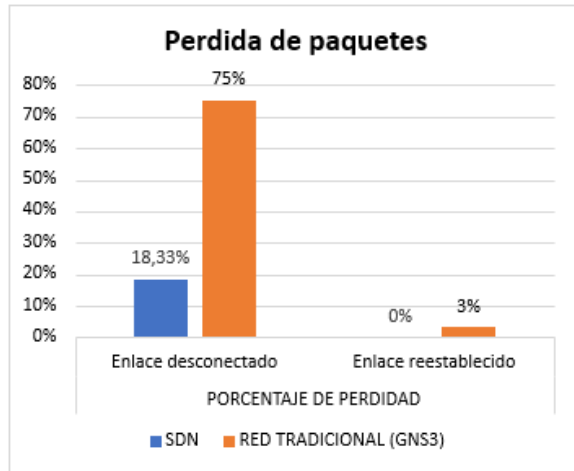


Fig. 5. Comparación de pérdida de paquetes.

### C. Throughput

Como tercer parámetro de comparación se evaluó el valor de Throughput para ambos tipos de redes, el parámetro Throughput representa la cantidad de datos efectiva enviada a través de un enlace, generalmente este valor es inferior al ancho de banda, ya que este correspondería al valor teórico.

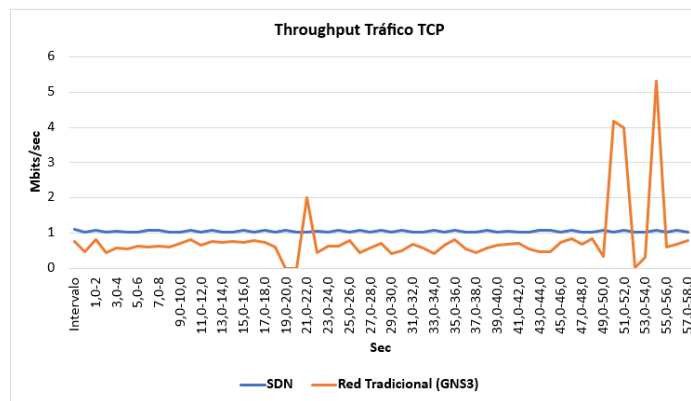


Fig. 6. Presentación de valores de throughput para protocolo TCP.

La figura 6 muestra los resultados obtenidos luego de realizada la medición en un tiempo igual a 60 segundos con un intervalo de 1 segundo, para ambos casos, la red SDN y convencional se realizó un primer envío de datos con un protocolo TCP entre el host 1 y el host 6 estando estos equipos ubicados en extremos opuestos de la red, es decir por la ruta más larga con el objetivo de utilizar todo el ancho de banda disponible, pudiéndose observar como en la red SDN el valor de Throughput se mantiene casi constante sin demasiada variación.

En valor promedio obtenido en la medición dio un valor de 1,05 Mbits/sec para redes SDN, por otro lado, en la red convencional implementada en GNS3 se pudo observar como el Throughput baja hasta un poco más de la mitad del ancho de banda, presentando en ocasiones picos muy altos o bajos. El valor

promedio obtenido fue de 0,622 Mbits/sec para la red convencional.

Como puede observarse en la figura 7 el rendimiento promedio de la red SDN está por encima de la red convencional.

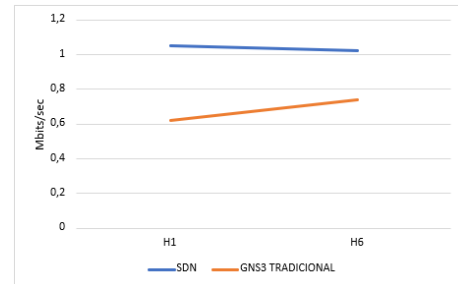


Fig. 7. Comparación de throughput de la red SDN respecto a una red convencional.

Debido a que el tráfico TCP requiere de mensajes de comprobación es decir esto implica el reenvío de varios de paquetes dentro de la red como consecuencia el rendimiento o throughput podría verse afectado, por ello con el objetivo de mejorar la transmisión de información se realizó una conexión UDP con ayuda de la herramienta iPerf cambiando su línea de comando ya anteriormente descrita, de igual manera la toma de datos se la hizo entre los hosts que se encuentran en el extremo de la topología en lugares opuestos (H1-H6) actuando como servidor y cliente respectivamente, se enviaron un total de 60 paquetes en un intervalo de 1 segundo.

En la figura 8 se puede visualizar como con este protocolo tanto la red SDN como la red convencional se mantienen mayormente estables, presentando ciertas variaciones en rangos constantes, ambas presentando un throughput promedio de 1.05 (Mbits/sec).

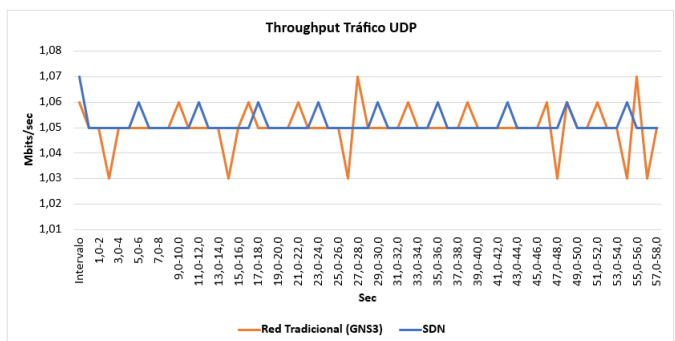


Fig. 8. Presentación de valores de throughput para protocolo UDP.

## VI. CONCLUSIONES

Tras las pruebas realizadas se pudo comprobar como los parámetros de rendimiento de la red (latencia, pérdida de paquetes, throughput) presentaron mejores valores, la red SDN ofrece valores promedio más bajos en comparación de una red convencional obteniendo un porcentaje de 99,22% menos de latencia para el valor mínimo, un 82,32% menos de latencia

para el valor promedio AVG y un 29,9% menos para su valor máximo. Con respecto a la pérdida de paquetes, se obtuvo un porcentaje de pérdida de 18,33% en comparación a la red convencional con un 77% evidenciando como tras la detección de pérdidas de comunicación el controlador es informado sobre el evento fortuito, el cual analiza y genera rutas alternas causando un mínimo descarte de datos. Para determinación del throughput se efectuó el envío de datos TCP y UDP, para el caso de la conexión TCP en las redes SDN se obtuvo valores promedio de 1.05 y 1.02 Mb/s los cuales representan un 95% aproximado del ancho de banda asignado. Para el envío de paquetes mediante UDP se obtuvo como resultado un valor promedio de 1.05 Mb/s para ambas redes, el cual representa el 95.45% del ancho de banda, observando como la red convencional tuvo una mejora en la transmisión de datos efectiva. Todos estos indicadores favorables para la red SDN se debe a la conformación de flujos dentro de la red implementada a través del controlador POX el cual posee varios componentes que permiten una gestión rápida del comportamiento del flujo de datos utilizando un componente de descubrimiento para así construir un árbol de expansión, además se puede descubrir la topología a través del envío de mensajes LLDP de los conmutadores OpenFlow, derivando todo esto en un menor reenvío de paquetes y por ende una convergencia más rápida de la red dejando de lado la revisión de puertos físicos cada vez que un paquete de datos alcanza un switch o un conmutador en una red convencional.

## VII. BIBLIOGRAFÍA

- [1] M. Á. Barrera Pérez, N. Y. Serrato Losada, E. Rojas Sánchez, and G. Mancilla Gaona, "State of the art in software defined networking (SDN)," *Visión electrónica*, vol. 13, no. 1, pp. 178–194, 2019.
- [2] J. J. Regalado Jalca et al., *Redes de computadoras*. 2018.
- [3] G. I. Bermudes, *Diseño De Una Red Sdn En La Dirección Provincial De Pichincha Del Consejo De La Judicatura*. 2018.
- [4] P. Alcívar and M. Navia, "Comparison between traditional network and software defined network: Case of study ESPAM MFL | Comparativa entre red tradicional y red definida por software: Caso de estudio ESPAM MFL," *RISTI - Rev. Iber. Sist. e Technol. Inf.*, vol. 2020, no. E29, pp. 79–90, 2020.
- [5] B. Valencia, S. Santacruz, and L. Y. B. J. J. Padilla, "Mininet : una herramienta versátil para emulación y prototipado de Redes Definidas por Software | Mininet : a versatile tool for emulation and prototyping of Software Defined Networking," no. 17, pp. 62–70, 2015.
- [6] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop," pp. 1–6, 2015.
- [7] Daily Herald, "Cisco challenged as Facebook favors software to move data," 10-Jun-2014. [Online]. Available: <https://www.dailyherald.com/article/20140610/business/140619927/>. [Accessed: 15-Feb-2022].
- [8] E. N. Hardware, "Escuela politécnica nacional," 2013.
- [9] C. Bermúdez, Cristian; Rodríguez, Jesús; Dussan, "Beneficios de las Redes Definidas por Software y el Protocolo Openflow," 2020, pp. 1–15, 2020.
- [10] V. Asmel Marrero, "Universidad Central ' Marta Abreu ' de Las Villas Universidad Central ' Marta Abreu ' de Las Villas," pp. 30–34, 2013.
- [11] I. F. Akyildiz, A. Lee, P. Wang, M. Luo, and W. Chou, "A roadmap for traffic engineering in software defined networks," *Comput. Networks*, vol. 71, pp. 1–30, 2014.
- [12] Open Networking Foundation, "SDN Architecture," *Onf*, no. 1, pp. 1–68, 2014.
- [13] U. P. D. E. Cartagena, "Programación de redes SDN mediante el controlador POX," pp. 1–85, 2015.
- [14] Brianlinkletter, "Using POX components to create a software defined networking application | Open-Source Routing and Network Simulation," 21-Sep-2015. [Online]. Available: <https://www.brianlinkletter.com/2015/09/using-pox-components-to-create-a-software-defined-networking-application/>. [Accessed: 14-Feb-2022].
- [15] McCauley, "Instalación de POX — Manual de POX Documentación actual," 2015. [Online]. Available: <https://noxrepo.github.io/pox-doc/html/#openflow-spanning-tree>. [Accessed: 15-Feb-2022].
- [16] D. Blandón Gómez, "Openflow: El protocolo del futuro," *Páginas Rev. académica e Inst. la UCPR*, no. 93, p. 6, 2013.
- [17] Pitt Dan, "Understanding OpenFlow, VXLAN and Cisco's ACI | Network Computing," Apr-2015.
- [18] T. Jos and A. Curso, "Autor : Álvaro Vázquez Rodríguez Tutor : José Carlos López Ardao," 2020.
- [19] REPARAR.INFO, "Uso de iPerf para probar la velocidad y el ancho de banda de la red (rendimiento) - Reparar Info." [Online]. Available: <https://reparar.info/uso-de-iperf-para-probar-la-velocidad-y-el-ancho-de-banda-de-la-red-rendimiento/> [Accessed: 15-Feb-2022].