Hamut'ay
Revista cuatrimestral de divulgación científica
UNIVERSIDAD ALAS PERUANAS
http://revistas.uap.edu.pe/ojs/index.php/HAMUT
ISSN: 2313-7878
Vol. 7 (3). Setiembre-Diciembre. Hamut'ay 2020. Lima-Perú

Análisis Comparativo de Protocolos de Comunicación para Redes definidas por Software

Comparative analysis of communication protocols for Software Defined Networks

Mocha Guacho, Geovanny[1]
https://orcid.org/0000-0002-2433-8650

Celleri Pacheco, Jennifer[2]
https://orcid.org/0000-0001-7041-8777
Universidad Técnica de Machala, Ecuador

Recibido: 10-10-2020
Aceptado: 23-12-2020

Cita Recomendada
Mocha-Guacho, G. & Celleri-Pacheco, J. (2020). Análisis comparativo de protocolos de comunicación para redes definidas por software. Hamut´ay, 7 (3), 39-50
http://dx.doi.org/10.21503/hamu.v7i3.2190

Resumen

Las redes definidas por software (en adelante SDN) son una innovación, que tiene como principal objetivo romper el paradigma de las redes tradicionales y presentar redes administrables o programables que se basan en el desacoplamiento de los planos de control y datos. Es destacable el territorio que van ocupando las SDN, sin embargo, no existen trabajos en los que se evidencien análisis comparativos de los protocolos hacia el sur (southbound) entre los planos antes mencionados. Al ser una tecnología que ha empezado a tener acogida, es necesario realizar estudios que permitan conocer y guiar a una fácil selección de los protocolos southbound a la hora de su implementación con la finalidad de mejorar el rendimiento. El propósito del presente estudio es analizar los protocolos de comunicación southbound, para lo cual se han seleccionado a OVSDB y OpenFlow. Se usó el método experimental para medir el rendimiento de la red mediante el análisis de la tasa de transferencia de paquetes dentro de la red. Para sustentar el método experimental, se empleó un ambiente de emulación Mininet sobre la que se implementó una SDN en una topología Lineal y Simple. Se obtuvo resultados que permitieron validar la hipótesis del estudio, comprobando que el protocolo de comunicación southbound pueden inferir en el rendimiento de la red. Como resultado del análisis se logró evidenciar que OpenFlow presenta mejores resultados de rendimiento en el envío de paquetes que OVSDB.

Palabras Clave: Mininet, Protocolos de comunicación southbound, Redes definidas por software, rendimiento de red.

1. Ingeniero de sistemas por la Universidad Técnica de Machala, estudiante del Máster en Software de la UTMACH. Sus áreas de interés son: desarrollo de software y aplicaciones móviles. Email: gmocha_est@utmachala.edu.ec
2. Docente titular de la UTMACH, Directora del grupo de investigación GIDCOWEB, Coordinadora de Maestría en Software, Coordinadora del Congreso Internacional TECDES, Coordinadora de la Red Temática de Ciencias de la Computación del Ecuador, Ingeniera de Sistemas con maestría en Informática Empresarial y doctorando en NTICs en la Universidade da Coruña. E-mail: jcelleri@utmachala.edu.ec

CC BY NC ND

Abstract

Software-defined networks (hereinafter SDN) are an innovation, whose main objective is to break the paradigm of traditional networks and present administrable or programmable networks that are based on the decoupling of control and data planes. The territory that the SDN is occupying is remarkable, however, there are no studies in which comparative analyzes of the protocols towards the south (southbound) between the aforementioned planes are evident. As it is a technology that has begun to be accepted, it is necessary to carry out studies that allow us to know and guide an easy selection of southbound protocols at the time of their implementation in order to improve performance. The purpose of this study is to analyze the southbound communication protocols for which OVSDB and OpenFlow have been selected. The experimental method was used to measure network performance by analyzing packet transfer rate within the network. To support the experimental method, a Mininet emulation environment was used on which an SDN was implemented in a Linear and Simple topology. Results were obtained that allowed to validate the hypothesis of the study, verifying that the southbound communication protocol can infer the performance of the network. As a result of the analysis, it was possible to show that OpenFlow presents better performance results in sending packages than OVSDB.

Key words: Mininet, Southbound Communication Protocols, Software Defined Networks, Network Performance

Introducción

El uso del internet se ha incrementado como consecuencia de la transformación digital (conocido también como la Nueva Era Digital) a la que se ha visto sometida a muchas empresas públicas y privadas (Fernández Torres et al., 2019). Ante la exigencia de adaptarse a estas tecnologías y el crecimiento de las competencias, se ha llevado a considerar a la infraestructura TI como un activo importante dentro de las organizaciones (Peña Casanova & Anías Calderón, 2019; Peña & Anías Calderón, 2018). La infraestructura TI, como se menciona en los trabajos de Peña Casanova & Anías Calderón, 2019; Célleri-Pacheco et al., 2018 mejoran los modelos de negocio generando beneficios y ahorro de los costos sobre las soluciones tecnológicas.

Este incremento del uso del internet también ha sido de consideración por parte del Ministerio de Telecomunicaciones y de la Sociedad de la Información. Y desde este ministerio se está impulsando la "necesidad de actualizar continuamente sus redes fijas y móviles a las últimas tecnologías, las cuales proveen mayores velocidades de subida y bajada de datos a los usuarios" (Ministerio de Telecomunicaciones y de la Sociedad de la Información, 2016). Ante esto, se ha considerado dentro de los objetivos del ministerio "Completar y fomentar el despliegue de infraestructura de Telecomunicaciones". Por tanto, es necesario desplegar una infraestructura de red con nuevas tecnologías.

Las redes tradicionales son una "estructura cerrada y estática que no se puede adaptar en tiempo real a la demanda de las aplicaciones" (Centeno et al., 2014). Una de las características es que este tipo de arquitectura de red se componen de tres planos (Pereira & Gamess, 2017) que son: control, datos y gestión, las mismas que están embebidas en los mismos dispositivos de red, lo que provoca una alta complejidad en su administración (Muro et al., 2017), y aumenta conforme crece el tamaño de la red (Molina & Fernando, 2017). Entre otras desventajas es que la implementación y mantenimiento de este tipo de red es costosa y limitada. Estos problemas presentados por las redes tradicionales, ha motivado a ir en busca de nuevas propuestas considerando el avance de la tecnología y el éxito del internet (Latifis, 2011). Es por ello que se busca innovar las redes mediante la administración de la red por medio de la programación (Benzekki et al., 2016).

De lo expuesto anteriormente, surgen las redes definidas por software, que se presentan como un paradigma innovador (Centeno et al., 2014) y que ha sido acogido por varias empresas (Parashar et al., 2019). Este nuevo paradigma se basa en el desacoplamiento del plano de control y datos. En el plano de control, se ubica un elemento denominado Controlador SDN, el cual se encarga de la administración de la red (Rams et al., 2017). En el trabajo de Gustavo Pereira se define a las SDN como un nuevo enfoque con "capacidad de inicializar, controlar, cambiar y gestionar el comportamiento de reenvío del tráfico de una red mediante APIs abiertas" (Pereira & Gamess, 2017). De esta forma se pretende centralizar la red desde un controlador (Kumari & Sairam, 2019; Saraswat et al., 2019) logrando conseguir redes administrables o programables y eficientes (Muro et al., 2016; Pérez & Marín, 2015). Sin embargo, pese a ser una gran novedad presentan problemas de seguridad y amenazas sobre la propia arquitectura (Krishnan et al., 2019). El terreno que han ido tomando las SDN, han generado que se empiecen a realizar estudios de este paradigma, logrando crear propuesta híbrida entre la red tradicional y SDN (Peña & Anías, 2018). Sin embargo, no existen trabajos en los que se evidencien análisis comparativos de los protocolos de comunicación hacia el sur (southbound) entre los planos antes mencionados. Al ser una tecnología que ha empezado a tener acogida, es necesario realizar estudios que permitan conocer y guiar a una fácil selección de los protocolos southbound a la hora de su implementación.

El objetivo de este estudio es implementar los protocolos de comunicación southbound en una arquitectura SDN mediante el uso de herramientas de emulación como Mininet para medir el rendimiento de la red entre los planos de control y datos. Consiste en realizar una descripción de los protocolos de comunicación Southbound empleados para el intercambio de información entre el plano de control y datos en una arquitectura SDN, de los cuales se eligen a OVSDB y OpenFlow considerando las características de la herramienta Mininet. Finalmente se realiza la implementación y ejecución de los protocolos de comunicación Southbound en una topología Lineal y Simple. Es importante destacar que en este estudio no tiene como finalidad determinar cuál es el mejor protocolo, sino, ser un punto de partida para la selección de un adecuado protocolo. La hipótesis que sigue el presente estudio es que, si se selecciona un adecuado protocolo de comunicación para SDN, entonces se logra optimizar la comunicación entre el software y hardware de una red. Conocer los protocolos de comunicación Southbound existentes permite facilitar la selección adecuada de un protocolo para redes basadas en software.

Redes Definidas por Software

Las redes tradicionales no pueden adaptarse a las nuevas demandas de aplicaciones y servicios que se ha extendido por el crecimiento de la demanda del internet (Ghonaim et al., 2018; Latifis, 2011), esto ha causado una compleja administración de la red (Muro et al., 2017). Además, este tipo de arquitectura no permite responder de forma eficaz a cambios constantes (Manzano et al., 2017) complicando el despliegue de nuevos servicios y aplicaciones (Gilces & Villamar, 2019). Por lo mencionado anteriormente, se ha motivado a reestructurar las redes tradicionales para mejorar su control, siendo una gran alternativa, hacer programable la red (Muro et al., 2016) por medio de su centralización en un controlador. Como consecuencia han surgido varias propuestas como las que se observan en la Figura 1.

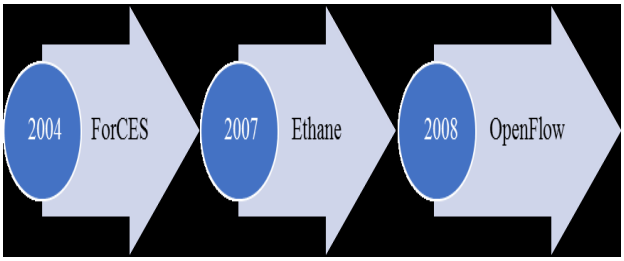

Figura 1: Propuesta para separación del plano de control y datos

Fuente: Elaboración propia (2020)

En el año 2004, surge Forwarding and Control Element Separation con la intención de separar

el control y reenvió de datos (Yang et al., 2004). FORCES es parte de un trabajo presentado por el IETF (Internet Engineering Task Force). Otra de las alternativas que se presenta es Ethane, desarrollado por la Universidad de Stanford en 2006 (Barrera et al., 2019). Se considera como una arquitectura de red para empresas (Casado et al., s. f.). Ethane se basa en la misma idea de tener un controlador centralizado que permita la admisión y enrutamiento de los flujos de datos. Finalmente, surge OpenFlow con su primera versión, la cual fue publicada en el 2008. OpenFlow se basa en el mismo paradigma de ForCes que es la separación del plano de control y datos. Ante esto, surge las redes definidas por software o denominadas SDN por sus siglas, que tiene como objetivo la separación del plano de control y el plano de datos (Gómez, 2013; Valencia, B. et al., 2015) lo cual permite programar la red en base a los requerimientos del usuario (Trois et al., 2016) sin generar costosos cambios en el hardware (Rodriguez et al., 2016; Tri-Hai Nguyen & Myungsik Yoo, 2017). Por los beneficios que presenta este nuevo paradigma es considerado para los sistemas de 5ta generación (Ordonez-Lucena et al., 2017). Las SDN se basan en una arquitectura de 3 planos (Huang et al., 2019), los cuales son: plano de aplicación, plano de datos y plano de control. Este nuevo paradigma en redes permite añadir dinámicamente nuevas características y servicios en forma de aplicaciones (Rams et al., 2017). En el trabajo de Magri Hicham se menciona que las SDN pueden ayudar a los operadores de redes móviles a respaldar mayor tráfico y disminuir los costos operativos (Hicham et al., 2018). Sin embargo, hay que destacar que el concepto de las SDN no son nueva, debido a que, desde el siglo pasado se ha venido concibiendo la idea de generar comandos para administrar la red (Alcívar & Navia, 2020).

Para el 2011, nace ONF (Open Networking Foundation) (Barrera Pérez et al., 2019), una comunidad fundada por: Deutsche Telekom, Facebook, Google, Microsoft, Verizon, y Yahoo. En la mayoría de trabajos que emplean la arquitectura SDN (Ampuño Avilés & Chávez Cristóbal, 2015; Carlos et al., 2014; Chico et al., 2014; Ibáñez Moruno et al., 2016; Trejos & Alzate, 2016), se habla del uso del protocolo OpenFlow.

OpenFlow, en la actualidad, existen varios protocolos que pueden ser empleados para la comunicación entre el plano de control y de datos, entre algunos están: ForCes, OVSDB, NETCONF Y BGP.

En el estudio de Rodriguez-Natal et al., (2015), se analiza a Locator/ID Separation Protocol (LISP) como un protocolo southbound aprovechando que las características se adaptan a la arquitectura de las redes SDN al cumplir con las características básicas de SDN: desacoplamiento de datos de control, la programabilidad de la red y el control centralizado.

En el trabajo de (Giraldo & Echeverry, 2018) se menciona varios casos de éxitos, entre los que destacan: Datacenter Interno de NEC, Empresa de Transporte en Metro - East Japan Railway Compay, Empresa Genesis Hosting Solutions, Hospital Universitario - Nagoya City University Hospital, entre otros.

Tabla 1: Propuesta de aplicación de redes SDN

Autor/Trabajo	Propósito	Aplicación
(Navarro et al., 2019)	Realizar una evaluación de la transmisión adaptativa de secuencias de video utilizando la técnica DASH en redes definidas por software. Empleo Mininet para la emulación	Transmisión de videos
(Trejos & Alzate, 2016)	Realizar el estado del arte sobre juegos en la nube, con el fin de generar documentación y presentar campos de investigación en esta tecnología.	Cloud gaming
(Jiménez Velásquez, 2019; Sanabria et al., 2018)	Implementar y evaluar una estrategia que garantice el mantenimiento de QoS para transmisión de video en tiempo real por medio de la administración aplicando SDN.	Transmisión de videos
(Ciungu, 2016)	programación de un algoritmo de detección de intrusos como una aplicación SDN	Detección de intrusos

(Barrientos-Avendaño et al., 2019)	Generar una infraestructura de granja inteligente que permita dar soporte a todos los procesos	proyectos de IOT
(Alves et al., 2018) **(Romero Amondaray et al., 2020)**	Diseño y la implementación de un marco SDN seguro para WSN	Redes de sensores
(Romero-Gázquez & Bueno-Delgado, 2018)	Una solución de arquitectura de software de código abierto basada en OpenDaylight (ODL), un controlador de red definida por software (SDN), para orquestar un escenario de IoT industrial.	IoT industrial

Fuente: Elaboración propia (2020)

En los últimos años, como se observa en la Tabla 1, se han considerado a las redes SDN para proyectos relacionados a IOT, transmisión de videos, juegos y otros. Uno de los usos que ha sido de gran demanda es la trasmisión de video generando demanda en el ancho de banda (Valencia, J. et al., 2020). Ante esta acogida, nace la necesidad de evaluar los protocolos de comunicación (Barrera et al., 2019) para analizar su comportamiento respecto a la trasmisión de paquetes entre los elementos de la red.

Como se menciona en el trabajo de (Gonzalez et al., 2018; Valencia, B. et al., 2015) las redes SDN se han convertido en un eje de atención para los investigadores en las áreas de las telecomunicaciones sugiriendo que se proponga nuevas líneas de investigación.

Materiales y métodos.

Participantes

En una arquitectura SDN los planos de aplicación, control y datos, están enlazados por medio de protocolos de comunicación o APIs: SouthBound y NorthBound entre las más conocidas. Sin embargo, también se incluyen a Westbound y Eastbound como en la arquitectura mostrada en el trabajo de Valencia, B. et al., (2015). Por lo tanto, para el presente trabajo se ha considerado centrar el enfoque de estudio sobre los protocolos de comunicación SouthBound.

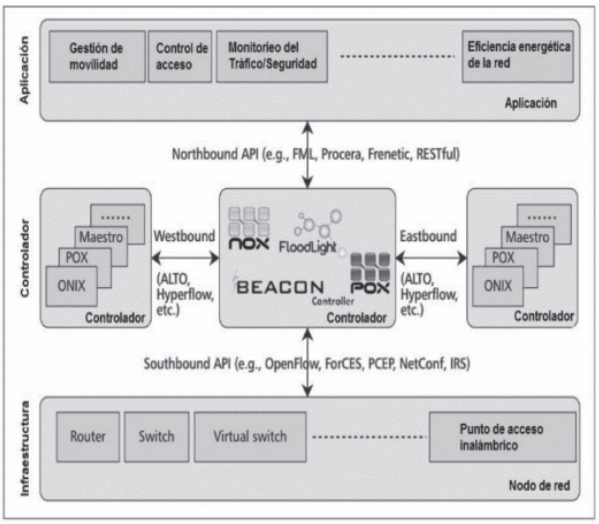

Figura 2: Arquitectura de red SDN
Fuente: Tomado del trabajo de Valencia, B. et al., (2015)

Como se puede observar en la Figura 2, los protocolos de comunicación southbound permiten el envío de información entre el plano de control (controlador) y el plano de datos (elementos de red). Son considerados como una interfaz que permite el controlador establezca el comportamiento de los equipos de red (Pereira et al., 2019). Para lo cual, se realizó una exploración sobre bases teóricas que permitió conocer los protocolos de comunicación southbound existentes. No obstante, hay que mencionar que no solo existen protocolos de comunicación (Tabla 2), sino APIs que permiten esta comunicación entre los planos antes mencionados.

Instrumento

Para el desarrollo del estudio, considerando el análisis de los protocolos southbound en redes definidas por software, se consideran los siguientes instrumentos:

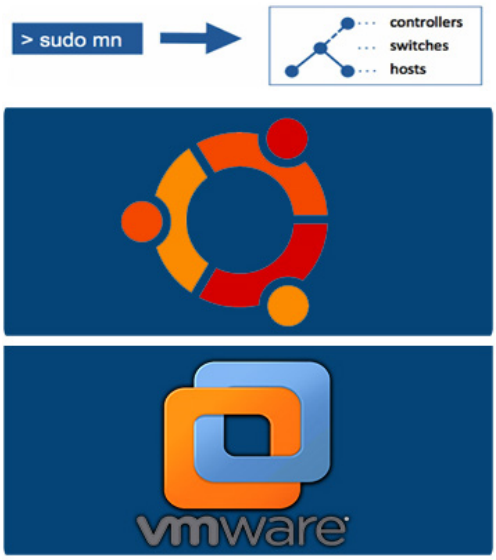

Figura 3: Materiales empleados para el estudio
Fuente: Elaboración propia (2020)

Mininet: Se seleccionó esta herramienta de emulación, considerando que soporta una red con arquitectura SDN, además, brinda funciones importantes para la creación de prototipos de la red SDN y permite la conexión con controladores reales (Valencia, B. et al., 2015).

Ubuntu: para preparar el ambiente sobre el que se ejecutará Mininet, se ha considerado el uso del SO Ubuntu como lo presenta el sitio web de Mininet.

VMware: Para ejecutar un ambiente de emulación, es necesario considerar una máquina virtual, sobre la cual se ejecutará el sistema operativo que soportará Mininet.

Tipo y Diseño

Para el presente estudio de los protocolos de comunicación empleados en las redes definidas por software, se realiza una investigación exploratoria y descriptiva. El enfoque exploratorio se aplica, como menciona Hernández, Fernández y Baptista, (2014), para examinar un tema o problema de investigación poco estudiado. Por medio del método exploratorio se obtiene información de los protocolos de comunicación existentes para redes SDN. El enfoque descriptivo, como menciona el mismo autor, permite describir personas, proceso, objeto o cualquier otro fenómeno. Por lo tanto, en este estudio se emplea el enfoque descriptivo para especificar las características de los protocolos de comunicación para redes SDN.

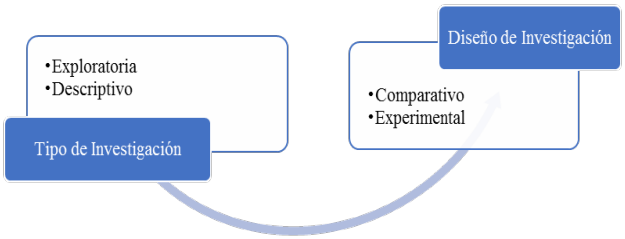

Figura 4: Tipo y diseño de investigación
Fuente: Elaboración propia (2020)

Una vez descrito cada uno de los protocolos de comunicación Southbound, se ha seleccionado dos de ellos. Para complementar este estudio de investigación, se ha empleado el método comparativo para analizar el rendimiento de la red considerando la tasa de transferencia en el envío de paquetes entre hosts aplicando los dos protocolos de comunicación seleccionados (OPENFLOW y OVSDB). Finalmente, para poner a prueba la hipótesis del estudio, se ha empleado el método experimental. Por medio de este diseño, se pretende observar las variables de investigación, es decir, si los protocolos de comunicación pueden incidir en el rendimiento de una SDN.

Procedimiento

Se ha planteado un proceso sistemático que consta de tres fases para el cumplimiento del desarrollo de la emulación de una SDN aplicando los protocolos de comunicación Southbound. A continuación, en la figura 1 se resume cada fase.

Figura 5: Proceso sistemático del estudio
Fuente: Elaboración propia (2020)

Recopilación de Información

En esta fase se detalla el resumen del análisis bibliográfico sobre los protocolos de comunicación Southbound. Los protocolos que se detallan en la Tabla 2, son basados en la revisión de la literatura realizada para la construcción del estado del arte y antecedentes conceptuales. Asimismo, se describen varios de los protocolos de comunicación empleados en las redes definidas por software.

Tabla 2: Protocolos Southbound

Ítem	Protocolo	Descripción
1	ForCES	Su funcionamiento se basa en una arquitectura maestro-esclavo donde los elementos de reenvío (FE) son esclavos y permiten que el elemento de control maestro (CE) los controle (Haleplidis et al., 2015). Esto quiere decir, que FE está a cargo de procesar y manejar los paquetes, mientras que CE se encarga de la administración y ejecución del enrutamiento de los paquetes.
2	BGP	Se lo clasifica como protocolo de ruta vectorial o un protocolo de vector de distancia. Se basa en su tabla de enrutamiento con las direcciones que se puede alcanzar, y se las van asociando a una métrica de costo (valor que representa la conexión a cada router).
3	OpenFlow	Considerado en muchas aplicaciones para protocolos hacia el sur (Alves et al., 2018). Es un protocolo a través del cual un controlador lógicamente centralizado puede controlar un interruptor OpenFlow. El enrutamiento de los paquetes se basa en las tablas de flujo (uno o más) de cada conmutador OpenFlow.
4	OVSDB	El protocolo de administración OVSDB usa JSON. Permite que las aplicaciones se conecten a la base de datos Open vSwitch, en donde se encuentra la configuración.
5	NetConf	Proporciona mecanismos para instalar, manipular y eliminar la configuración de los dispositivos de red. Emplean el lenguaje XML (Extensible Markup Language) para los datos de configuración y los mensajes de protocolo. Tienen como objetivo la reducción de la complejidad y mejora del rendimiento de la red.

Fuente: Elaboración propia (2020)

Además, se pudo encontrar que no solo existen protocolos de comunicación, sino que existen API y framework que pueden ser utilizadas en otros tipos de propuestas.

Tabla 3: Resumen de Protocolos Southbound

	Protocolo	API	Framework	Controlador Compatible
ForCES	X		X	Vendedor específico
OpenFlow	X			Opendaylight, RYU, ONOS, Openvirtex, POX
OVSDB	X			Opendaylight, RYU, ONOS, Opencontrail
Netconf		X		Opendaylight, RYU, ONOS, Opencontrail
Xmpp		X		Opencontrail
BGP LS	X			Opendaylight
BGP	X			Opencontrail

Fuente: Elaboración propia (2020)

Como se puede observar en la Tabla 3, se resumen los protocolos hacia el sur, no todos son protocolos, algunos son API como en el caso de XMPP que es empleado para mensajería instantánea haciendo envió de información por medio del formato XML. Otro caso a destacar es ForCES, que no solo se considera un protocolo sino también un Framework con la particularidad que es proporcionado por un vendedor especifico, lo que quiere decir, que son implementados en dispositivos propios.

Por lo tanto, para seguir a la siguiente fase solo se consideran a los protocolos: ForCES, OpenFlow, OVSDB, BGP LS y BGP.

Diseño de escenario para laboratorio

En esta fase, se analizó el soporte que puede brindar Mininet a los protocolos definidos en la fase anterior. Es decir, se realizó una revisión teórica de la herramienta para identificar que protocolos pueden ejecutarse en Mininet. Como resultado de la revisión se obtuvo que OpenFlow y OVSDB están soportados por la herramienta de emulación.

Tabla 4: Protocolos Southbound soportado por Mininet

Protocolos	Controlador Compatible	Soportado por MININET
ForCES	Vendedor específico	
OpenFlow	Opendaylight, RYU, ONOS, Openvirtex, POX	X
OVSDB	Opendaylight, RYU, ONOS, Opencontrail	X
BGP LS	Opendaylight	
BGP	Opencontrail	

Fuente: Elaboración propia (2020)

Se han diseñado dos topologías (lineal y simple) sobre las cuales los protocolos seleccionados (Tabla 4) fueron experimentados bajo controladores diferentes.

- Protocolo OpenFlow con el controlador POX
- Protocolo OVSDB con el controlador RYU

La topología lineal se empleó para experimentar con los protocolos, y para validar los resultados

obtenidos se experimentó en una topología distinta (simple).

Pruebas y análisis de resultados

Para los diseños de las topologías descritas en la fase anterior, se configuro un servidor HTTP, luego se procedió a comprobar la conectividad. Se estableció al Host 2 como cliente web haciendo una petición GET al servidor web (Host 1) como se observa en la Figura 6.

Figura 6: Petición GET desde cliente a servidor HTTP
Fuente: Consola de Mininet (2020)

Se obtuvo éxito en el envío de paquetes ICMP realizado desde el H2 (cliente) al H1 (servidor) como se evidencia en la Figura 7.

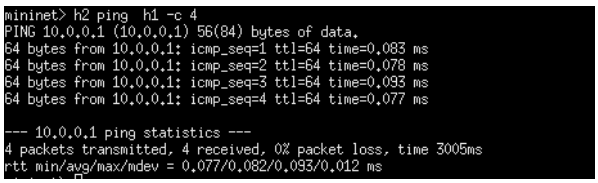

Figura 7: Resultado del ping realizado entre H2 y el servidor HTTP en RYU
Fuente: Consola de Mininet (2020)

Además, se verificó la petición HTTP en Wireshark con igual prueba de solicitud desde el cliente hacia el servidor HTTP como se aprecia en la Figura 8.

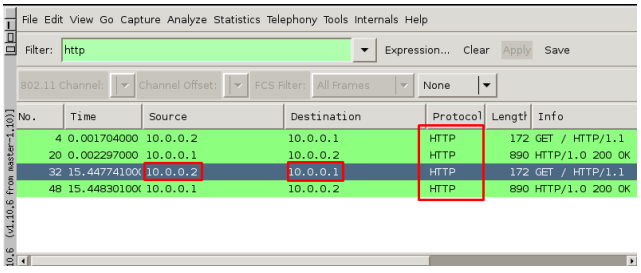

Figura 8: Verificación de petición HTTP
Fuente: Consola de Mininet (2020)

Se realizó el envío de paquetes ICMP entre los hosts H2 y H3 para verificar el comportamiento de la red. En este punto, se consideraron los siguientes criterios:

- Enviar paquetes de distintos tamaños (256, 512, 1024 y 2048 bytes)
- Realizar 5 pruebas de conectividad para cada tamaño de paquete.
- Cada prueba con 10 paquetes enviados de un mismo tamaño.

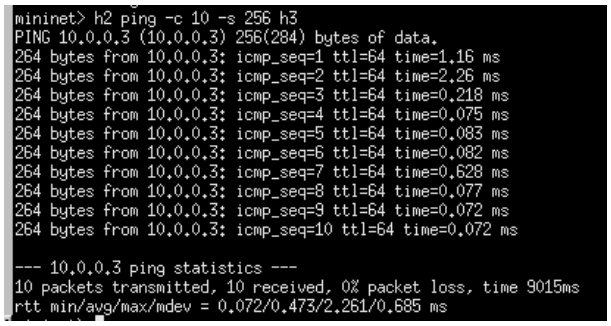

Figura 9: Resultado del ping realizado entre H2 y H3
Fuente: Consola de Mininet (2020)

Los criterios mencionados, se establecieron para la ejecución de los protocolos OpenFlow y OVSDB en las topologías Lineal y Simple.

Resultados

Una vez realizadas las pruebas en la topología linear (lineal), aplicando el proceso que se describió anteriormente se han obtenido los siguientes resultados:

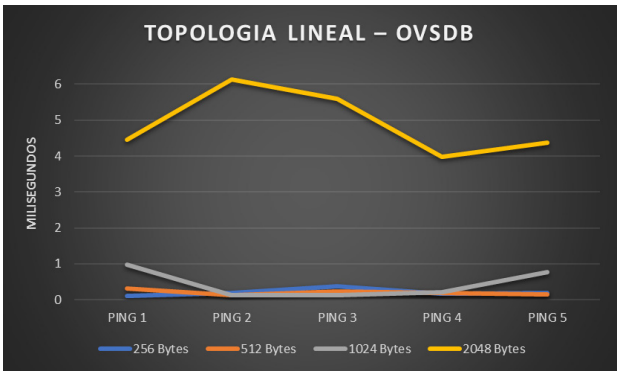

Figura 10: Resultado de OVSDB con topología lineal
Fuente: Elaboración propia (2020)

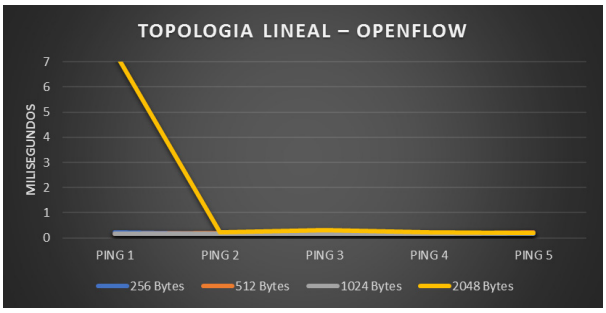

Figura 11: Resultado de OpenFlow con topología lineal
Fuente: Elaboración propia (2020)

Como se pude observar en la Figura 10 y 11 los resultados permiten evidenciar que el protocolo OpenFlow presenta mejores resultados de tasa de transferencia en el envío de paquetes con diferentes cargas (256, 512, 1024 y 2018 bytes) entre hosts.

En el caso del protocolo OVSDB (Figura 10), se han obtenido valores de hasta 6 milisegundos, a diferencia del protocolo OpenFlow que arroja mejores valores, llegando a estar por debajo de 1 milisegundo.

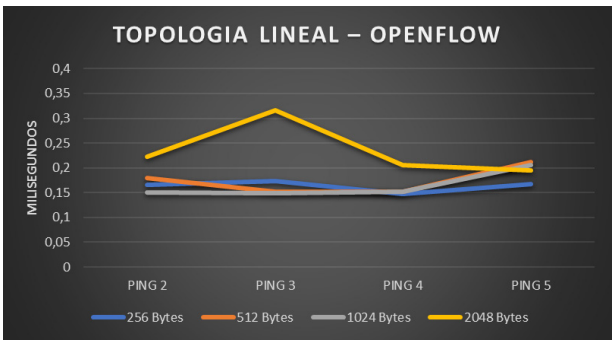

Figura 12: Resultados con la topología Lineal con OpenFlow
Fuente: Elaboración propia (2020)

Para mejorar la visualización de los resultados del protocolo OpenFlow, se han omitido los valores del primer envío de paquetes ICMP, y como se observa en la Figura 12 los resultados obtenidos en la topología propuesta no superan los 0.35 milisegundos.

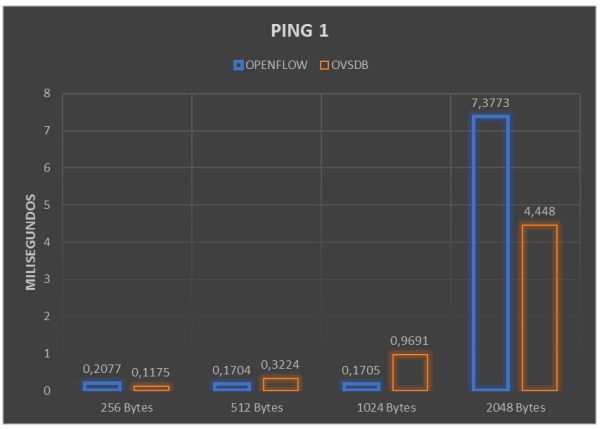

Figura 13: Resultados con diferentes tamaños de paquetes ICMP
Fuente: Elaboración propia (2020)

Otro dato importante que se observa en el comportamiento de estos protocolos, es que en el primer envío de paquetes con carga de 2048 bytes toma valores altos, es decir, mayor latencia en relación a los siguientes paquetes que se envían, y esto está relacionado al tiempo de convergencia de la red. Además, es importante mencionar que la topología lineal incluye un Switch al que se conecta cada Host lo que podría aumentar la latencia.
Sin embargo, OpenFlow a pesar del retardo inicial, logra estabilizarse con el paso del tiempo hasta su convergencia, a diferencia del protocolo OVSDB que en paquetes de 2048 bytes mantiene valores similares a los de los primeros paquetes enviados.

Para validar el mejor comportamiento del protocolo OpenFlow, se realizó otra prueba en una topología diferente, se escogió una topología Single (simple), obteniendo los siguientes resultados.

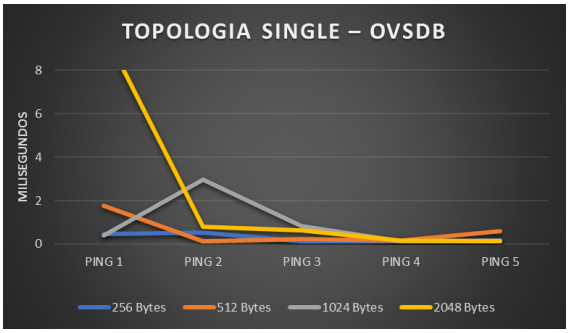

Figura 14: Resultado de OVSDB con topología single o simple
Fuente: Elaboración propia (2020)

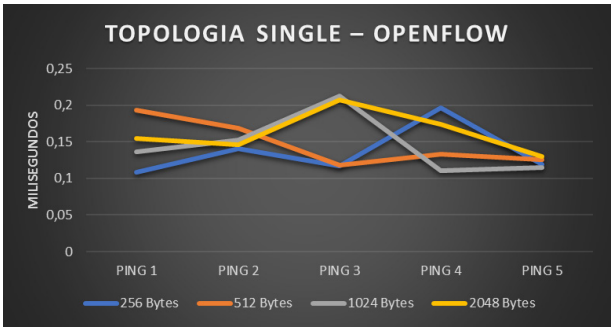

Figura 15: Resultado de OpenFlow con topología single o simple
Fuente: Elaboración propia (2020)

En este caso, ambos protocolos presentaron resultados, en general, similares al presentado en la topología Lineal. Como se evidencia en la Figura 14, el protocolo OVSDB alcanza un valor superior a 8 milisegundos (9,815ms) en el primer envío de paquetes ICMP con carga de 2048 bytes, a diferencia del protocolo OpenFlow que otorga valores que no superan 0.25 milisegundos en las 5 pruebas realizados con los diferentes tamaños de paquetes.

Conclusiones

El análisis de los protocolos de comunicación permite conocer alternativas con respecto a la implementación de una SDN considerando la importancia de la comunicación entre el plano de control y de datos. Estos protocolos pueden inferir en un mejor rendimiento en el envío de paquetes, sin embargo, esto también puede variar según el diseño de la topología. Como se pudo evidenciar en el presente estudio, al tener una topología simple con un switch se puede proporcionar un mejor rendimiento en él envió de los paquetes. A diferencia de una topología lineal, en la que se le agrega un switch para la conexión de cada host, y que por tanto agrega latencia necesaria para procesar el paquete desde el switch origen hacia el controlador, y posteriormente al switch del destino para que reenvíe al host receptor. Con este análisis, se deduce que el diseño de la topología juega un papel importante no solo en la selección del protocolo, sino para la implementación de la SDN.

Otro hallazgo en este estudio es que el protocolo OpenFlow ofrece una mejor comunicación. Observando los resultados, la tasa de transferencia mejora con el uso del protocolo OpenFlow en ambas topologías diseñadas.

El proceso de selección de la herramienta de emulación, en este caso Mininet, ha permitido evidenciar su gran capacidad para el despliegue de este tipo de arquitecturas. También se puede destacar, que presta un sinnúmero de comandos que pueden ser ejecutados desde los hosts creados.
Con este estudio se pretende generar una guía para introducir a nuevos análisis acerca de los otros componentes de una SDN, como son controladores, los protocolos NorthBound, además del análisis de seguridad en cada uno de los planos. Incluso, se puede desarrollar el software con el que se pretende interactuar desde el plano de aplicaciones hacia el controlador.

Otro trabajo que puede desarrollarse a partir de este estudio es la implementación de una SDN física, para ello es indispensable contar con los dispositivos adecuados y el software para el controlador.

Finalmente, se recomienda, comprobar otros protocolos hacia el sur con el mismo controlador, para lo cual se debe tener al alcance los equipos y software necesarios.

Agradecimiento

Este artículo fue desarrollado y financiado por la Universidad Técnica de Machala, por medio de la Gestión de la Maestría en Software mediante el aporte económico y académico para su publicación.

Referencias Bibliográficas

Alcívar, P., & Navia, M. (2020). Comparativa entre red tradicional y red definida por software: Caso de estudio ESPAM MFL. Revista Ibérica de Sistemas e Tecnologias de Informação, E29, 79–90.
Alves, R. C. A., Oliveira, D. A. G., Pereira, G. C. C. F., Alber-

tini, B. C., & Margi, C. B. (2018, agosto 1). WS3N: Wireless Secure SDN-Based Communication for Sensor Networks [Research Article]. Security and Communication Networks; Hindawi. https://doi.org/10.1155/2018/8734389

Ampuño Avilés, A. R., & Chávez Cristóbal, M. M. (2015). Diseño y simulación de una red de Datacenters basada en topología FAT-TREE en un ambiente de redes definidas por software (SDN).

Barrera Pérez, M. Á., Serrato Losada, N. Y., Rojas Sánchez, E., & Mancilla Gaona, G. (2019). State of the art in software defined networking (SDN). Visión Electrónica, 13(1), 178-194. https://doi.org/10.14483/22484728.14424

Barrientos-Avendaño, E., Rico-Bautista, D., Coronel-Rojas, L. A., & Cuesta-Quintero, F. R. (2019). Granja inteligente: Definición de infraestructura basada en internet de las cosas, IpV6 y redes definidas por software. Revista Ibérica de Sistemas e Tecnologias de Informação, E17, 183–197.

Benzekki, K., El Fergougui, A., & Elbelrhiti Elalaoui, A. (2016). Software-defined networking (SDN): A survey. Security and Communication Networks, 9(18), 5803-5833. https://doi.org/10.1002/sec.1737

Carlos, J., Mejia, D., & Bernal, I. (2014, febrero 1). Implementación de un Prototipo de una Red Definida por Software (SDN) Empleando una Solución Basada en Hardware.

Casado, M., Freedman, M. J., Pettit, J., Luo, J., McKeown, N., & Shenker, S. (s. f.). Ethane: Taking Control of the Enterprise. 12.

Célleri-Pacheco, J., Andrade-Garda, J., & Rodríguez-Yáñez, S. (2018). Cloud Computing para PYMEs. Machala: Universidad Técnica de Machala. http://repositorio.utmachala.edu.ec/handle/48000/12507

Centeno, A. G., Vergel, C. M. R., Calderón, C. A., & Bondarenko, F. C. C. (2014). Controladores SDN, elementos para su selección y evaluación. Telemática Magazine, 13(3), 10-20.

Chico, J. C., Mejía, D., & Bernal, I. (2014). Implementación de un Prototipo de una Red Definida por Software (SDN) Empleando una Solución Basada en Hardware. 25, 10.

Ciungu, R. S. (2016). Improving IoT security with software defined networking [Master's Thesis]. Universitat Politècnica de Catalunya.

Fernández Torres, Y., Gutiérrez Fernández, M., & Palomo Zurdo, R. (2019). ¿Cómo percibe la banca cooperativa el impacto de la transformación digital? CIRIEC-España, Revista de Economía Pública, Social y Cooperativa, 95, 11. https://doi.org/10.7203/CIRIEC-E.95.12724

Ghonaim, F. A., Darcie, T. E., & Ganti, S. (2018). Impact of SDN on optical router bypass. IEEE/OSA Journal of Optical Communications and Networking, 10(4), 332-343. https://doi.org/10.1364/JOCN.10.000332

Gilces, C. E. M., & Villamar, R. P. (2019). Aplicación de Balanceo De Carga Dinámico Para Servidores, Basada En Redes Definidas Por Software. RISTI-Revista Ibérica de Sistemas e Tecnologias de Informação, 32, 67–82.

Giraldo, M. R., & Echeverry, A. M. L. (2018). Redes de datos definidas por software—SDN, arquitectura, componentes y funcionamiento. 7.

Gómez, D. F. B. (2013). OPENFLOW: EL PROTOCOLO DEL FUTURO. Revista Académica e Institucional, 93, 61-72.

Gonzalez, C., Flauzac, O., & Nolot, F. (2018). Evolución y Contribución para el Internet de las Cosas por las emergentes Redes Definidas por Software. Memorias de Congresos UTP, 28-33.

Haleplidis, E., Salim, J. H., Halpern, J. M., Hares, S., Pentikousis, K., Ogawa, K., Wang, W., Denazis, S., & Koufopavlou, O. (2015). Network Programmability With ForCES. IEEE Communications Surveys Tutorials, 17(3), 1423-1440. https://doi.org/10.1109/COMST.2015.2439033

Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (2014). Metodología de la investigación.

Hicham, M., Abghour, N., & Ouzzif, M. (2018). 5G mobile networks based on SDN concepts. 7, 2231-2235. https://doi.org/10.14419/ijet.v7i4.12194

Huang, X., Cheng, S., Cao, K., Cong, P., Wei, T., & Hu, S. (2019). A Survey of Deployment Solutions and Optimization Strategies for Hybrid SDN Networks. IEEE Communications Surveys Tutorials, 21(2), 1483-1507. https://doi.org/10.1109/COMST.2018.2871061

Ibáñez Moruno, F., Lévano Lévano, J. A., & Nieto Maldonado, E. S. (2016). Diseño e implementación de una herramienta de visualización para análisis en tiempo real de redes SDN/OpenFlow.

Jiménez Velásquez, Á. L. (2019). Implementación y evaluación de una estrategia para garantizar mantenimiento de QoS en la transmisión de video en tiempo real en redes WLAN bajo el esquema de gestión SDN.

Krishnan, P., Duttagupta, S., & Achuthan, K. (2019). VARMAN: Multi-plane security framework for software defined networks. Comput. Commun. https://doi.org/10.1016/j.comcom.2019.09.014

Kumari, A., & Sairam, A. S. (2019). A Survey of Controller Placement Problem in Software Defined Networks. arXiv:1905.04649 [cs]. http://arxiv.org/abs/1905.04649

Latifis, S. (2011). Análisis al futuro de las arquitecturas de internet. Revista Ingenierías USBMed, 2(1), 18. https://doi.org/10.21500/20275846.244

Manzano, S., Pallo, J., González, P., & Escobar, A. (2017). Gestión de flujo de datos en una red definida por software en relación a variables externas. UTCiencia" Ciencia y Tecnología al servicio del pueblo", 3(2), 73–84.

Ministerio de Telecomunicaciones y de la Sociedad de la Información. (2016). Plan de Telecomunicaciones y TI. https://www.telecomunicaciones.gob.ec/wp-content/uploads/2016/08/Plan-de-Telecomunicaciones-y-TI..pdf

Molina, C., & Fernando, A. (2017). Análisis de servicios Web en redes SDN. http://repositorio.uchile.cl/handle/2250/144287

Muro, Y. A., Alvarez Paliza, F., Carbonell, A., & Dueñas Santos, C. (2017). SDN Application For The Network Access Control. En Conferencia Científica Internacional 8Va Edición.

Muro, Y. A., Alvarez Paliza, F., & Carbonell, A. (2016). Colaboracion De Ims Y Sdn-Openflow. Una Arquitectura Para Mitigar Problemas De Seguridad En Redes Futuras. Revista TONO, 13, 34-39.

Navarro, F. W. S., Bustos, J. G., & Hernández, W. E. C. (2019). Adaptive video transmission over software defined networks. Visión Electrónica, 13(1), 152-161. https://doi.org/10.14483/22484728.14398

Ordonez-Lucena, J., Ameigeiras, P., Lopez, D., Ramos-Munoz, J. J., Lorca, J., & Folgueira, J. (2017). Network Slicing for 5G with SDN/NFV: Concepts, Architectures, and Challenges. IEEE Communications Magazine, 55(5), 80-87. https://doi.org/10.1109/MCOM.2017.1600935

Parashar, M., Poonia, A., & Satish, K. (2019). A Survey of Attacks and their Mitigations in Software Defined Networks. 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 1-8. https://doi.org/10.1109/ICCCNT45670.2019.8944621

Peña Casanova, M., & Anías Calderón, C. (2018). Empleo de modelos de información en arquitectura modificada para gestión de redes y servicios basada en políticas. Ingeniería Electrónica, Automática y Comunicaciones, 39(3), 77–88.

Peña Casanova, M., & Anías Calderón, C. (2019). Sistema para ejecutar políticas sobre infraestructuras de Tecnologías de la Información. Ingeniare. Revista chilena de ingeniería, 27(3), 479–494. https://doi.org/10.4067/S0718-33052019000300479

Peña, M., & Anías Calderón, C. (2018). Empleo de modelos de información en arquitectura modificada para gestión de redes y servicios basada en políticas. 39, 77-88.

Pereira, G., Silva, J., & Sousa, P. (2019). Comparative Study of Software-Defined Networking (SDN) Traffic Controllers. 2019 14th Iberian Conference on Information Systems and Technologies (CISTI), 1-6. https://doi.org/10.23919/CISTI.2019.8760997

Pereira, G. & Gamess, E. (2017). Lineamientos para el Despliegue de Redes SDN/OpenFlow. 4(2), 13.

Pérez, G. C., & Marín, M. F. (2015). Redes definidas por software: Solución para servicios portadores del Ecuador. INVESTIGATIO, 6, 41–63. https://doi.org/10.31095/investigatio.2015.6.2

Rams, J. A., Calderón, C. A., & Fonseca, D. F. (2017). Lenguajes de alto nivel de abstracción para el desarrollo de aplicaciones SDN. Telemática, 16(2), 1-11.

Rodriguez, N., Murazzo, M., & Medel, D. (2016). Consideraciones sobre la arquitectura de Internet del Futuro. Revista Eletrônica Argentina-Brasil de Tecnologias da Informação e da Comunicação, 1(5).

Rodriguez-Natal, A., Portoles-Comeras, M., Ermagan, V., Lewis, D., Farinacci, D., Maino, F., & Cabellos-Aparicio, A. (2015). LISP: A southbound SDN protocol? IEEE Communications Magazine, 53(7), 201-207. https://doi.org/10.1109/MCOM.2015.7158286

Romero Amondaray, L., Artigas Fuentes, F. J., Calderón, C. A., Romero Amondaray, L., Artigas Fuentes, F. J., & Calderón, C. A. (2020). Redes de Sensores Inalámbricos Definidas por Software: Revisión del estado del arte. Ingeniería Electrónica, Automática y Comunicaciones, 41(2), 39-50.

Romero-Gázquez, J. L., & Bueno-Delgado, M. V. (2018, septiembre 27). Software Architecture Solution Based on SDN for an Industrial IoT Scenario [Research Article]. Wireless Communications and Mobile Computing; Hindawi. https://doi.org/10.1155/2018/2946575

Sanabria, F., Bustos, J., & Castellanos, W. (2018). Estudio de la transmisión de video sobre redes definidas por software (Study on Video Streaming over Software Defined Networks). MEMORIAS-CICI.

Saraswat, S., Agarwal, V., Gupta, H. P., Mishra, R., Gupta, A., & Dutta, T. (2019). Challenges and solutions in Software Defined Networking: A survey. Journal of Network and Computer Applications, 141, 23-58. https://doi.org/10.1016/j.jnca.2019.04.020

Trejos, F. D., & Alzate, N. (2016). Cloud gaming: A survey. Entre Ciencia e Ingeniería, 10(20), 82-91.

Tri-Hai Nguyen, & Myungsik Yoo. (2017). Analysis of link discovery service attacks in SDN controller. 2017 International Conference on Information Networking (ICOIN), 259-261. https://doi.org/10.1109/ICOIN.2017.7899515

Trois, C., Del Fabro, M. D., de Bona, L. C. E., & Martinello, M. (2016). A Survey on SDN Programming Languages: Toward a Taxonomy. IEEE Communications Surveys Tutorials, 18(4), 2687-2712. https://doi.org/10.1109/COMST.2016.2553778

Valencia, B., Santacruz, S., Becerra, L. Y., & Padilla, J. J. (2015). Mininet: Una herramienta versátil para emulación y prototipado de Redes Definidas por Software. Entre ciencia e ingenieria, 9(17), 62–70.

Valencia, J. C. C., Muñoz, W. Y. C., & Golondrino, G. C. (2020). Análisis de QoS para IPTV en un Entorno de Redes Definidas por Software. Revista de Ingenierías: Universidad de Medellín, 19(36), 29-51. https://doi.org/10.22395/rium.v19n36a2

Yang, L., Anderson, T., Dantu, R., & Gopal, R. (2004). Forwarding and Control Element Separation (ForCES) Framework. https://doi.org/10.17487/rfc3746