Análisis del protocolo Openflow usando mininet para una red SDN

Openflow protocol analysis using mininet for an SDN network (Software Defined Network)

Edith Paola Estupiñán Cuesta[1]

Juan Carlos Martínez Quintero[2]

Michael Steven Torres Chicaguy[3]

Universidad Militar Nueva Granada

Resumen

En este documento se describe el análisis a detalle de los protocolos involucrados en las redes definidas por software (SDN), mediante el diseño y captura de tráfico en una red SDN básica, haciendo énfasis especial en el protocolo Openflow. Por medio de esta simulación se llega a determinar que el comportamiento del controlador SDN es similar al de un router de las redes convencionales, y se identificaron los principales mensajes que intervienen en el protocolo openflow, determinando sus funciones y consecutivo, haciendo uso del analizador de protocolos Wireshark.

Palabras clave:

SDN (redes definidas por software); Openflow; conmutación; enrutamiento; protocolos.

Abstract

This document describes the detailed analysis of the protocols involved in Software Defined Networks (SDN), through the design and capture of traffic in a basic SDN network, with special emphasis on the Openflow protocol. Through this simulation, it is determined that the behavior of the SDN controller is similar to that of a router in conventional networks and the main messages involved in the openflow protocol were

[1] https://orcid.org/0000-0002-4100-4943 / edith.estupinan@unimilitar.edu.co

[2] https://orcid.org/0000-0001-9893-6592 / juan.martinezq@unimilitar.edu.co

[3] Est.michael.torres@unimilitar.edu.co / https://orcid.org/0000-0001-8716-4558

identified, determining their functions and consecutive, using the protocol analyzer Wireshark.

Keywords:

Software Define Network; Openflow; switch; protocolos.

1. Introducción

Las redes de computadoras convencionales han marcado un cambio total en la forma en que las personas acceden a la información, al igual que los métodos de comunicación y de trabajo, como se evidencia en la actualidad. El problema de este modelo convencional es que se convierte en un modelo difícil de gestionar y administrar (IBM, n.d.).

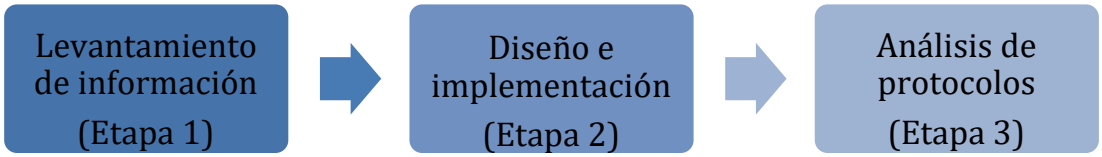
De allí, surgen las redes definidas por software (SDN), un modelo de gestión y administración de redes centralizado, en la cual, todos los nodos o switches de la red se pueden gestionar directamente desde el controlador SDN, eliminando todos los gastos de movilidad y reduciendo en gran medida los tiempos de resolución de problemas técnicos (Kreutz *et al.,* 2015).

El gran cambio radica en la división entre el proceso de toma de decisiones lógicas y el almacenamiento de dichas instrucciones para su rápida ejecución, en situaciones idénticas que lleguen a presentarse. La toma de decisiones lógicas se llevará a cabo en el controlador SDN y el almacenamiento de las instrucciones en los dispositivos de capa 2 o Switches openflow. En este punto, es clave el uso del protocolo Openflow, un protocolo de comunicaciones encargado de establecer la comunicación bidireccional entre el controlador SDN y el switch, para la sincronización y el envío de las instrucciones necesarias entre ambas partes. (Gopakumar *et al.,* 2016)

Este artículo presenta el estudio y análisis de protocolos de una red SDN diseñada en el software Mininet a través de tres etapas: 1) Diseño de la red SDN en el software mininet. 2) Simulación y captura de tráfico de cada uno de los escenarios por medio de Wireshark. 3) Análisis de los mensajes intercambiados y protocolos usados. Este artículo contiene de manera inicial una contextualización teórica del funcionamiento a nivel general de las redes SDN, el procedimiento de creación de las redes y el análisis de resultados.

2. Metodología

Para realizar esta investigación se propone un análisis de los protocolos utilizados en las redes SDN, y se plantea la siguiente metodología a utilizar basada en tres etapas:



1. *Etapa 1. Levantamiento de información:* en esta etapa se realiza la explicación teórica del funcionamiento de las redes convencionales y se realiza la comparativa con el funcionamiento del nuevo modelo de redes SDN.
2. *Etapa 2. Diseño e implementación:* en esta etapa se ilustra el procedimiento de creación de una red básica SDN y la captura de los mensajes, por medio de Wireshark, para la realización del análisis del protocolo openflow.
3. *Etapa 3. Análisis de protocolos:* en esta etapa se realiza el análisis de los mensajes capturados en anteriores etapas para ilustrar el funcionamiento del protocolo openflow y su integración con otros protocolos estandarizados actualmente.

3. Discusión

De acuerdo con la metodología propuesta, a continuación, se muestra el desarrollo de cada una de las etapas:

4. Desarrollo

4.1 Etapa 1. Funcionamiento de las redes SDN

Para entender el funcionamiento de las redes de datos es necesario entender las funciones que cumplen los dos planos que conforman los dispositivos de red: el control plane, encargado de realizar la toma de decisiones incluyendo la elaboración de las tablas de enrutamiento y las tablas ARP (Kreutz *et al.,* 2015). El data plane, encargado de recibir los datos y de elaborar las tablas orientadas específicamente a componentes

físicos, como la tabla FIB, en términos de puertos e interfaces físicas de salida (Kreutz *et al.,* 2015).

En la nueva arquitectura propuesta en el modelo de las redes SDN, los dos planos se separan, asignándole el control Plane al dispositivo conocido como controlador SDN, el cual está a cargo de la toma de decisiones lógicas, mientras que el data plane se almacena en los switches. Para que exista una correcta comunicación entre los dos planos, se desarrolló el protocolo Openflow (Beshley *et al.*, 2019).

En cuanto a la forma de configuración de los controladores SDN, es necesario especificar que se realiza a través de programación en aplicaciones conectadas por medio de APIS. En este caso, los controladores SDN son compatibles con diversos lenguajes de programación como lo son Python, Java, XML o JSON. Toda esta arquitectura la podemos visualizar en la Figura 1 (Kreutz *et al.,* 2015).

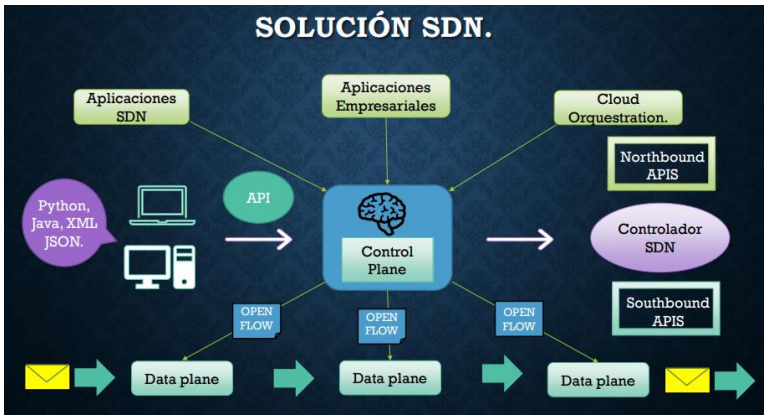


Figura 1. Arquitectura convencional de una red SDN. Fuente: elaboración propia.

4.2 Etapa 2. Diseño e implementación

Para el diseño e implementación de una red SDN, se usó el programa Mininet, un emulador para el despliegue de redes SDN que permite emular el comportamiento del controlador, los switches OpenFlow y los hosts (Mininet, n.d.).

En este programa, la creación de la red se puede realizar a través de comandos o por medio de una interfaz web. Para efectos visuales, la red diseñada para el análisis de protocolos se elaboró en la interfaz gráfica y se ilustra en la Figura 2. Como se puede observar, la red está compuesta por dos hosts, un controlador SDN y un switch de tipo openflow. De igual

forma, se ilustran las configuraciones realizadas a nivel de direccionamiento.

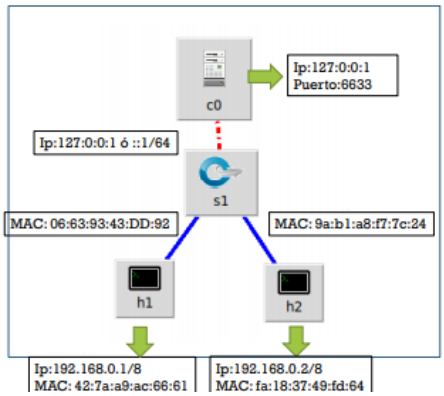


Figura 2. Esquema de direccionamiento. Fuente: elaboración propia.

4.3 Etapa 3. Análisis de protocolos:

Una vez se implementó la red, se generó tráfico y se realizó la captura mediante Wireshark. Teniendo en cuenta su protocolo base de operación Openflow, se realizó un filtrado por "Openflow_V1" teniendo en cuenta que la versión del protocolo que van a utilizar para intercambiar mensajes el switch y el controlador es 1.

Los mensajes involucrados inicialmente se ilustran en la Figura 3.

33	76.576798196	127.0.0.1	127.0.0.1	OpenFl…	74	Type: OFPT_HELLO
35	76.615150342	127.0.0.1	127.0.0.1	OpenFl…	74	Type: OFPT_HELLO
37	76.616281447	127.0.0.1	127.0.0.1	OpenFl…	74	Type: OFPT_FEATURES_REQUEST
39	76.616319546	127.0.0.1	127.0.0.1	OpenFl…	78	Type: OFPT_SET_CONFIG
41	76.618515509	127.0.0.1	127.0.0.1	OpenFl…	130	Type: OFPT_PORT_STATUS
43	76.618608583	127.0.0.1	127.0.0.1	OpenFl…	74	Type: OFPT_FEATURES_REQUEST
44	76.618626359	127.0.0.1	127.0.0.1	OpenFl…	78	Type: OFPT_SET_CONFIG
47	76.619242252	127.0.0.1	127.0.0.1	OpenFl…	242	Type: OFPT_FEATURES_REPLY

Figura 3. Mensajes iniciales. Fuente: elaboración propia.

En este intercambio de mensajes iniciales entre el controlador y el switch, suceden diversas acciones. Inicialmente, se establece la versión del protocolo Openflow a utilizar para la comunicación de acuerdo con las capacidades del switch. Después, se intercambian ciertas características entre el switch y el controlador incluyendo la cantidad de puertos disponibles y los protocolos que se pueden utilizar como la configuración de VLAN.

4.4 Pruebas de conexión

Al realizar conexión entre dos host, se evidencia que de igual forma se involucran una serie de protocolos usualmente utilizados como lo son ARP, para la obtención de la dirección MAC, ICMP, para la realización del PING, entre otros. Todo esto encapsulado en el protocolo Openflow, que establece la comunicación con el controlador SDN para realizar la gestión y la toma de decisiones de manera centralizada, como se observa en la Figura 4.

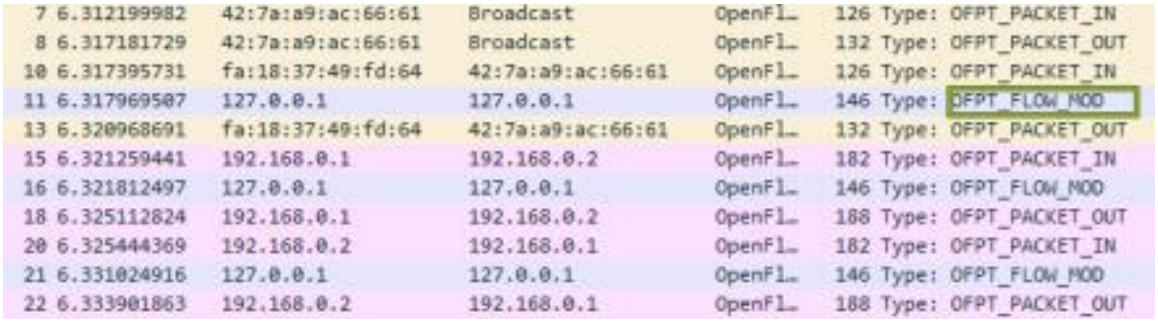


Figura 4. Mensajes de ping. Fuente: elaboración propia.

Como se ilustra en la Figura 4, para la realización del ping entre dos host que no se conocen, primero por medio de mensajes broadcast y con el protocolo ARP se identifica la dirección MAC del host destino. Estos protocolos se encapsulan en el protocolo openflow como se evidencia en la Figura 5.

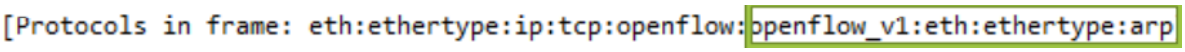


Figura 5. Encapsulación evidenciada. Fuente: elaboración propia.

En seguida se produce un intercambio de información entre el switch y el controlador SDN, que tiene la finalidad de llenar las tablas de flujo del switch con instrucciones acerca de qué acción tomar con respecto al paquete que se ha recibido.

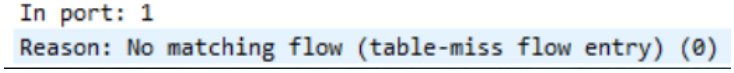


Figura 6. Mensaje del switch. Fuente: elaboración propia.

Por ejemplo, en la Figura 6 se evidencia una notificación que le realiza el switch al controlador, indicando que tras una lectura de sus tablas de flujo no ha encontrado instrucciones claras acerca de qué hacer con el paquete que ha llegado.

Por otro lado, en la Figura 7, se ilustran las instrucciones que le envía el controlador SDN al switch, para que pueda llenar sus tablas de flujo y

determinar qué debe hacer con el paquete. En este caso, el controlador le envía las direcciones de destino de la red a la que debe llegar.

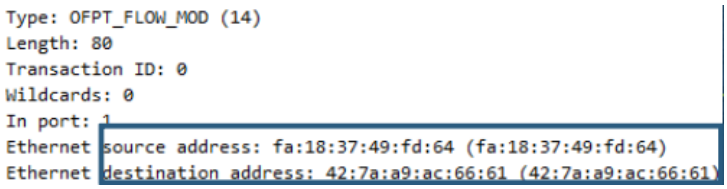


Figura 7. Instrucciones. Fuente: elaboración propia.

Una vez el switch ha recibido las instrucciones enviadas por el controlador a través de los mensajes OFPT_FLOW_MOD, el ping se logra completar de manera exitosa.

En resumen, en la Figura 8 se ilustran cada uno de los mensajes evidenciados en las pruebas realizadas. De igual forma, se incluyen el número que los identifica dentro del tráfico del protocolo Openflow y el origen de acuerdo a si el mensaje es enviado desde el switch o desde el controlador.

Numero	Mensaje	Origen
0	OFPT_HELLO	Switch y Controlador
2	OFPT_ECHO_REQUEST	Switch
3	OFFPT_ECHO_REPLY	Controlador
5	OFFPT_FEATURE_REQUEST	Controlador
6	OFPT_FEATURES_REPLY	Switch
9	OFPT_SET_CONFIG	Controlador
10	OFPT_PACKET_IN	Switch
12	OFPT_PORT_STATUS	Switch
13	OFPT_PACKET_OUT	Controlador
14	OFPT_FLOW_MOD	Controlador

Figura 8. Mensajes evidenciados. Fuente: elaboración propia.

Siguiendo este proceso, el controlador SDN puede tener el rol de control plane para múltiples switches, lo que representa una enorme ventaja en la gestión de la red si lo contrastamos con las redes convencionales que requieren múltiples routers para realizar la gestión de la misma.

5. Conclusiones

El controlador SDN simula el comportamiento de un router en cuanto a la toma de decisiones, con la especial característica de manejar la gestión de toda la red de forma centralizada.

El protocolo Openflow se integra eficazmente con otros protocolos que se utilizan en redes convencionales ya estandarizados, como lo son ARP o ICMP, haciendo que su implementación y sustitución en diferentes entornos actuales de la industria, sea mucho más sencillo.

Debido a la relevancia y a las ventajas que ofrecen las redes SDN en las infraestructuras de red actuales de diversas empresas, se determina la importancia de incluir su estudio en las asignaturas básicas de la carrera de ingeniería de telecomunicaciones.

Referencias

Beshley, M., Pryslupskyi, A., Panchenko, O., & Beshley, H. (2019). SDN/Cloud Solutions for Intent-Based Networking. *2019 3rd International Conference on Advanced Information and Communications Technologies (AICT)*, 22-25. https://www.semanticscholar.org/paper/SDN%2FCloud-Solutions-for-Intent-Based-Networking-Beshley-Pryslupskyi/01ba4a487d193d1e1b282145337b9febd4aa054b

IBM. (n.d.). *SDN Versus Traditional Networking Explained | IBM*. https://www.ibm.com/services/network/sdn-versus-traditional-networking

Kreutz, D., Ramos, F. M. V., Veríssimo, P. E., Rothenberg, C., EAzodolmolky . S. & Uhlig, S. (2015). Software-Defined Networking: A Comprehensive Survey. *Proceedings of the IEEE*, *103*(1), 14-76. https://doi.org/10.1109/JPROC.2014.2371999.

Selvaraj, P., & Nagarajan, V. (2017). Migration from conventional networking to software defined networking. *IEEE International Conference on IoT and Its Applications, ICIOT 2017.* https://doi.org/10.1109/ICIOTA.2017.8073632

Gopakumar, R., Unni, A. M., & Dhipin, V. P. (2016). An adaptive algorithm for searching in flow tables of openflow switches. *Proceedings of the 2015 39th National Systems Conference, NSC 2015, 1–5.* https://doi.org/10.1109/NATSYS.2015.7489115

Van Rossem, S., Tavernier, W., Sonkoly, B., Colle, D., Czentye, J., Pickavet, M., & Demeester, P. (2016). Deploying elastic routing capability in an SDN/NFV-enabled environment. *2015 IEEE Conference on Network Function Virtualization and Software Defined Network, NFV-SDN 2015*, 22–24. https://doi.org/10.1109/NFV-SDN.2015.7387398

harbaoui, M., Contoli, C., Davoli, G., Cuffaro, G., Martini, B., Paganelli, F., Cerroni, W., Cappanera, P., & Castoldi, P. (2018). Demonstration of Latency-Aware and

Self-Adaptive Service Chaining in 5G/SDN/NFV infrastructures. *2018 IEEE Conference on Network Function Virtualization and Software Defined Networks, NFV-SDN 2018*, *i*, 5–6. https://doi.org/10.1109/NFV-SDN.2018.8725645

Tatang, D., Quinkert, F., Frank, J., Röpke, C., & Holz, T. (2017). SDN-GUARD: Protecting SDN controllers against SDN rootkits. *2017 IEEE Conference on Network Function Virtualization and Software Defined Networks, NFV-SDN 2017*, *2017-January*, 297–302. https://doi.org/10.1109/NFV-SDN.2017.8169856

Gopakumar, R., Unni, A. M., & Dhipin, V. P. (2016). An adaptive algorithm for searching in flow tables of openflow switches. *Proceedings of the 2015 39th National Systems Conference, NSC 2015*, 1–5. https://doi.org/10.1109/NATSYS.2015.7489115

Mininet. (n.d.). *Mininet: An Instant Virtual Network on Your Laptop (or Other PC) - Mininet*. Retrieved June 6, 2021, from http://mininet.org/