



Análisis de herramientas que permitan el modelado de tráfico en redes SDN

Analysis of traffic modeling tools in SDN networks

Centro Sur.
Social Science Journal
Julio – Diciembre Vol 4 No 2
<http://centrosureditorial.com/index.php/revista>
eISSN: 2600-5743
revistacentrosur@gmail.com
Recepción: 4 de marzo 2018
Aprobación 12 noviembre 2019
Doi: 10.37955/cs.v4i2.57
Pag 1-15
Atribución/Reconocimiento-
NoComercial-CompartirIgual 4.0
Licencia Pública Internacional —
CC BY-NC-SA 4.0
<https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode.es>

Ángel Iván Torres Quijije¹
Andrea Raquel Zuñiga Paredes²

Resumen

El presente trabajo consiste en el estudio de herramientas que permiten configurar un entorno de simulación o emulación de redes dentro de un mismo sistema facilitando de cierta forma plantear un escenario de red que cuenta con topologías y flujos de tráfico. Específicamente se enfocó en el modelado de tráfico en redes SDN (Redes Definidas por Software) que permite que un solo equipo administre una red en su totalidad, haciéndolo capaz de tolerar cambios en toda la estructura de la red sin necesidad de detener los servicios. Se realiza una investigación bibliográfica de fuentes de carácter documental sobre simuladores que permiten el modelado de redes para determinar la más adecuada; los instrumentos analizadas fueron: NS-3, EstiNet y MiniNet. Posteriormente de acuerdo a la información obtenida y el estudio del

mismo, se realizó una tabla comparativa con los parámetros que distinguen a cada una, permitiendo de esta forma concluir de manera idónea la mejor opción para emulación de topologías en ambientes SDN.

Palabras clave: emulación, SDN, EstiNet, MiniNet.

Abstract

The present research work consists of the analysis of the tools that allow configuring a simulation or network emulation environment within a single

¹ Máster en Conectividad y Redes de Ordenadores. Universidad Técnica Estatal de Quevedo. Quevedo - Ecuador. Email: atorres@uteq.edu.ec ORCID: 0000-0002-7037-7191 Google académico: <https://scholar.google.com.ec/citations?hl=es&user=JmhrffMAAAAJ>

² Master en Matemáticas y ciencias de la Computación. Universidad Autónoma Regional de los Andes. Quevedo-Ecuador. Email: uq.andreazuniga@uniandes.edu.ec ORCID: 0000-0002-6231-262X. Google académico: <https://scholar.google.com.ec/citations?hl=es&user=5wg27JgAAAAJ>

system, facilitating in some way a network scenario that has topologies and traffic flows. Specifically, it focused on the traffic modeling in SDN networks (Software Defined Networks), which allows a single team to manage a network in its entirety, making it able to tolerate changes in the entire network structure without stopping services. A bibliographic investigation of documentary sources on simulators that allow the modeling of networks to determine the most appropriate one is

carried out; The instruments analyzed were: NS-3, EstiNet and MiniNet. Subsequently, according to the information obtained and the study of the same, a comparative table was made with the parameters that distinguish each one, thus allowing to ideally conclude the best option for emulation of topologies in SDN environments.

Key words: emulation, SDN, EstiNet, MiniNet

Introducción

Las Redes Definidas por Software (SDN) están siendo consideradas como un nuevo paradigma en redes de datos, debido a las diferentes ventajas que tiene con respecto al funcionamiento de las redes convencionales. Las SDN separan el plano de control de un dispositivo de red, del plano de datos, permitiendo controlar, monitorizar y gestionar una red desde un nodo centralizado o controlador, lo cual promete simplificar la gestión de red e incluir innovación a través de su programación.

A lo largo de la historia las redes de telecomunicaciones han ido evolucionando eventualmente, debido el crecimiento del volumen de información que se ha dado en estos años. Según datos de la ITU (ITU, ITU- ICT STATISTICS, 2017), hasta el 2017 el uso de internet ha seguido creciendo constantemente a nivel mundial, con lo las redes han tenido que irse adaptando constantemente a las nuevas necesidades de los usuarios. Estas nuevas redes se las conoce como redes de nueva generación (NGN), la cual define un concepto de red basada en IP que soporta los requerimientos de todos los servicios que se pueden ofrecer (REYES & CORTÉS, 2010).

El nacimiento de esta nueva generación de redes ha tenido muchos aspectos positivos dentro de las tics, pero también han sido un reto constante en su administración debido a la incursión de nuevos conceptos dentro de éstas, los administradores de redes tienen que adaptarse a los nuevos requerimientos. En 1996 (ITU, ITU, s.f.), algunos organismos de estandarización dentro de las tics ya tenían una idea del problema que iba a ser la administración de grandes redes en los siguientes años, con lo que incursionó el concepto de redes definidas por software (SDN).

SDN consiste en proporcionar redes más autónomas con la finalidad de poder administrar cada nodo de manera inteligente gracias a un software de control de la misma. Actualmente esta nueva perspectiva de administración de las redes a estado

en constante desarrollo por parte de grandes desarrolladores como CISCO, pero han salido proyectos a nivel mundial de investigadores y desarrolladores, con lo que empezaron a nacer entornos de emulación para SDN.

Estos entornos de simulación llegaron como solución a los nuevos desarrolladores e investigadores en la implementación de laboratorios enfocados en estas redes, pero sin invertir tanto. En la actualidad existen algunas herramientas que permiten emular redes SDN cada una con sus características y enfoques, en este artículo se analizarán NS-3, EstiNet y Mininet.

SDN

Las SDN se enfocan en la programación de redes que consiste en la capacidad de inicializar, controlar, cambiar y gestionar el comportamiento de reenvío del tráfico de una red, esto se lo logra mediante una interfaz de programación de aplicaciones o también conocidas como APIs estas interfaces suelen ser de código abierto. En una red SDN se apartan los planos de control y datos de los dispositivos de red y se desplaza el plano de control a una unidad central conocida como controlador SDN. Un controlador SDN es el que se encarga de definir y comunicar las reglas de reenvío de tráfico a los dispositivos SDN y abstraer la infraestructura de red y su topología a las aplicaciones (E. Haleplidis, 2015).

Al separar los planos de datos y control, los switches de la red se convierten en dispositivos de reenvío simples y la lógica de control se implementa en un sistema operativo de red centralizado o NOS (Network Operating System) (H. Kim, 2013).

Arquitectura de Red SDN

Una red SDN está conformada por tres capas que son: Capa de Aplicación, Capa de Plano de Control y Capa de Plano de Datos, en la ilustración 1 se puede ver la descripción gráfica de estas capas.

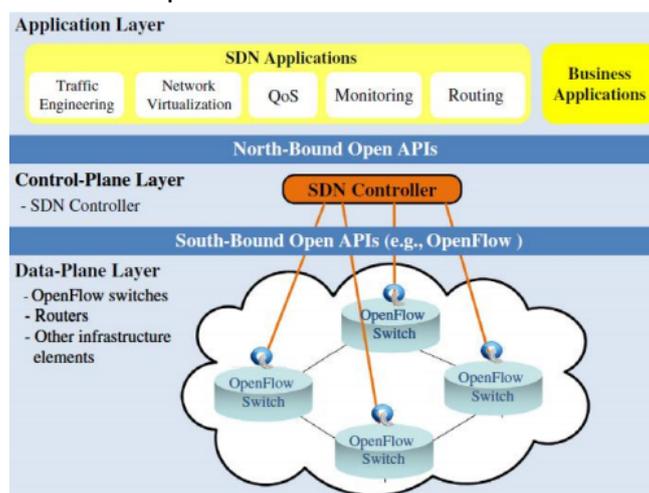


Ilustración 1: Representación Lógica de una Arquitectura de Red SDN.

Fuente: https://upcommons.upc.edu/bitstream/handle/2099.1/24951/Thesis_Axifra.pdf?sequence=1&isAllowed=y

Como primer punto se tiene la capa de aplicación que está conformada por las aplicaciones de negocio y de servicios de red (Porxas, 2014). Luego pasamos a la capa del Plano de Control que Contiene los controladores SDN encargados de gobernar y dirigir la manera en que se transportan los datos en los dispositivos SDN. Esta capa se encuentra a cargo de todas las funciones complejas de enrutamiento, manejo de políticas, monitoreo y chequeos de seguridad de la red (Porxas, 2014). Por último se tiene a la capa del Plano de Datos que está conformada por dispositivos SDN físicos y virtuales encargados de transportar datos en base a instrucciones recibidas por los controladores SDN de la red (Porxas, 2014).

Modelos de Despliegue SDN

En la práctica se utilizan tres modelos de despliegue SDN: Modelo SDN basado en Dispositivos (Foundation O. N., 2012), Modelo SDN Overlay (Foundation O. O., 2014), y Modelo SDN Híbrido (Masayoshi Kobayashi, 2014).

El Modelo SDN como se lo puede visualizar en la ilustración 2, está basado en Dispositivos se refiere a una red de switches físicos SDN que operan solo bajo las instrucciones de un controlador SDN. Se implementa con rapidez en despliegues nuevos como un nuevo complejo de oficinas dentro de un campus (Foundation O. N., 2012).

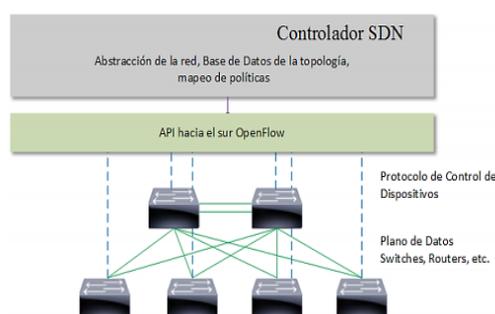


Ilustración 2: Ejemplo del Modelo SDN basado en Dispositivos.

Fuente: <http://concisa.net.ve/memorias/CoNCISa2017/CoNCISa2017-p060-072.pdf>

Otro de los modelos que se suele utilizar es SDN Overlay que como se puede visualizar en la ilustración 3, está basado en la superposición de redes sobre una infraestructura de red física subyacente. En este modelo los nodos finales SDN son dispositivos virtuales que forman parte de hipervisores en un ambiente de virtualización de servidores (Foundation O. O., 2014).

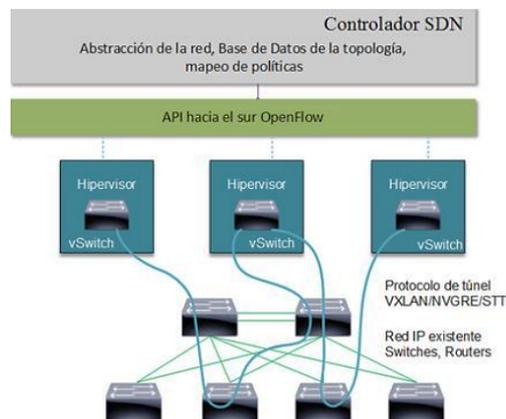


Ilustración 3: Modelo SDN Overlay.

Fuente: <http://concisa.net.ve/memorias/CoNCISa2017/CoNCISa2017-p060-072.pdf>.

Por último está el modelo SDN Híbrido que está basado en la convivencia de tecnologías de redes tradicionales con tecnologías de red SDN en un mismo entorno (Masayoshi Kobayashi, 2014). En este caso un gateway SDN corre tanto el modelo SDN overlay, como el modelo SDN basado en dispositivos para esquemas híbridos de modelos SDN basado en dispositivos con modelos SDN overlay (Masayoshi Kobayashi, 2014). También existe la opción de disponer de un gateway SDN que corre protocolos legados tradicionales y protocolos SDN/OpenFlow (Masayoshi Kobayashi, 2014). El gateway en un esquema híbrido se comunica con el controlador SDN OpenFlow y con los dispositivos de red Ethernet tradicionales y corre bajo ambos esquemas de red y protocolos (Masayoshi Kobayashi, 2014).

OPENFLOW

El protocolo de conmutador OpenFlow proporciona una interfaz abierta para controlar la conectividad y los flujos dentro de esa conectividad en una Red definida por software (SDN). Este protocolo es extensible, es decir, que proporciona mecanismos para que los programadores de SDN definan elementos de protocolo adicionales (por ej., Nuevos campos de coincidencia, acciones, propiedades de puertos, etc.) para abordar las nuevas tecnologías y comportamientos de red (ONF).

EMULACIÓN DE REDES DEFINIDAS POR SOFTWARE

La evaluación y la realización de pruebas de protocolos de red pueden realizarse a través de diferentes pruebas experimentales, simuladores y emuladores (Tseng, 2014). Cuando se utiliza la prueba de emulación, el sistema a evaluar se representa con algunos elementos modificados, pero otros se tratan exactamente igual que en un caso real, debido a esto, en la emulación todo se ejecuta en tiempo real (Saldaña, 2014).

HERRAMIENTAS DE SIMULACIÓN Y EMULACIÓN PARA SDN

Los emuladores y/o simuladores más utilizados actualmente para la evaluación de Redes Definidas por Software son: ns-3 (ns-3, s.f.), EstiNet (EstiNet, EstiNet Technologies, 2015) y Mininet (Mininet, s.f.).

NS-3

Según su historia, Network Simulator (ns) tiene sus orígenes alrededor del año 1989 como una extensión de otro simulador, llamado REAL. Sin embargo, la primera versión oficial e independiente nació en 1995 con la ayuda de DARPA (agencia encargada de desarrollar tecnologías para el uso militar), la Universidad de Berkeley y el centro de investigación XEROX PARC de California, entre otros (Palacio, 2016). El núcleo de ns-1 estaba fundamentado en un lenguaje básico de programación de objetos, como es el C++, además de incluir también en sus bases el lenguaje Tool Command Language (Tcl) (Palacio, 2016).

Años después, nació la segunda versión del simulador, conocida como ns-2. En esta contribuyeron Sun Microsystems y UCB - Daelus, contribuyendo con módulos para simular topologías inalámbricas. ns-2 siguió basando su arquitectura en el lenguaje C++ pero evolucionó ante el lenguaje de comandos Tcl, pasando a utilizar su extensión orientada a objetos, llamada Object Tool Command Language (OTcl) (Palacio, 2016).

Finalmente, en 2005 de la mano de Tom Henderson se propuso el núcleo de la tercera y actual versión del simulador, totalmente incompatible con la anterior. Desarrolladores de la Universidad de Washington, el Instituto Tecnológico de Georgia y el centro de investigación francés INRIA en Junio de 2008 dieron apertura al primer lanzamiento, llamado ns-3.1 (Palacio, 2016).

Actualmente, esta versión sigue en desarrollo, encontrándose ahora mismo en la ns-3.25, lanzada en Marzo del 2016. El núcleo sigue centrado en el uso del lenguaje C++, sin embargo, también tiene disponibles sus características para el lenguaje Phyton (Palacio, 2016).

NS-3 en su página oficial menciona que es un simulador de red para eventos discretos, destinados principalmente para el uso de investigación y educación. Ns-3 es un software gratuito usado principalmente entornos Linux, licenciado bajo la licencia GNU GPLv2, y está disponible públicamente para investigación, desarrollo, uso, modificación y extensión de su código (ns-3, 2011-17) (Palacio, 2016).

La finalidad, es evitar la utilización de sistemas reales que incrementen el coste y la dificultad de la investigación (Palacio, 2016).

En la actualidad la herramienta de software libre ns-3 es el simulador de red más utilizado en el mundo (S.-Y. Wang, C.-L. Chou and C.-M. Yang, 2013). Ns-3 permite

simular el funcionamiento de un controlador OpenFlow real, implementando de C++, que compila y vincula con su código de motor de simulación (S.-Y. Wang, C.-L. Chou and C.-M. Yang, 2013). De hecho, todos los dispositivos y objetos simulados en ns-3 se implementan como módulos C++ compilados y enlazados con su código de motor de simulación para formar un programa ejecutable a nivel de usuario, es decir, ns-3 (S.-Y. Wang, C.-L. Chou and C.-M. Yang, 2013).

Direct Code Execution (DCE) es una infraestructura para NS3 que provee facilidades para ejecutar, dentro de NS3, implementaciones existentes de protocolos de red sea en el espacio de usuario o del kernel del sistema operativo, incluso puede ejecutarse aplicaciones; en ambos casos sin la necesidad de realizar cambios en código fuente (Christian Alexander Valdivieso Pinzón, 2017/06/30).

Abstracciones empleadas en NS3

Para trabajar en NS3 se consideran algunas abstracciones que usa el simulador como:

Nodos: en NS3 es una computadora o host a la cual se le pueden añadir ciertas funcionalidades como el stack de protocolos, aplicaciones o interfaces de red. Esta abstracción se emplea mediante la clase Node, la cual provee funciones para el manejo de los hosts en la simulación. Otra clase relacionada es NodeContainer, la cual provee la abstracción de un arreglo de nodos (nsman).

Aplicación: En NS3, una aplicación es un programa de software que corre sobre los nodos para realizar ciertas tareas a nivel de usuario. Esta abstracción es implementada por la clase Application, la cual provee funciones para el manejo de aplicaciones a nivel de usuario (nsman).

Canal: Un canal en NS3 permite la conexión entre nodos. La clase que realiza esta abstracción es Channel, la cual provee funciones para el manejo de la conexión entre los nodos. Una clase específica derivada de Channel es PointToPointChannel (nsman).

Dispositivos de Red: Para NS3, un dispositivo de red cubre el software y el hardware de una tarjeta de red. Cuando un dispositivo de red se asocia a un nodo, el nodo está en la capacidad de comunicarse con otros mediante un canal. La clase que maneja esta abstracción se la denomina NetDevice, ésta maneja las funciones para comunicar los nodos a través de un canal (nsman). La clase NetDeviceContainer permite crear un arreglo de dispositivos de red (nsman).

TopologyHelper: Esta abstracción en NS3 facilita la asociación entre los distintos nodos, dispositivos de red y canales. Clases específicas son, por ejemplo: DceManagerHelper, InternetStackHelper (nsman).

Tecnologías clave en ns-3

Ns-3 es una biblioteca basada en C++ que proporciona un conjunto de modelos de simulación de red implementados como objetos C++ y envueltos a través de Python (1808). En la ilustración 4 se puede visualizar como están compuestas estas bibliotecas internamente.

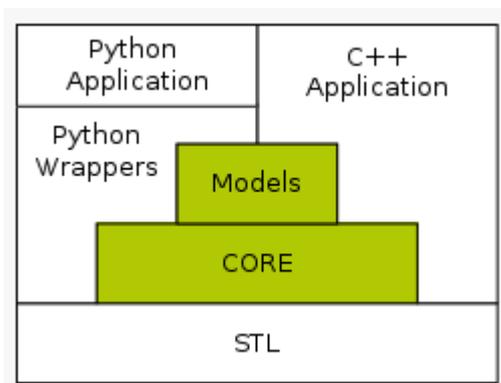


Ilustración 4: Biblioteca ns-3 Fuente: ns-3

Estructura de ns-3

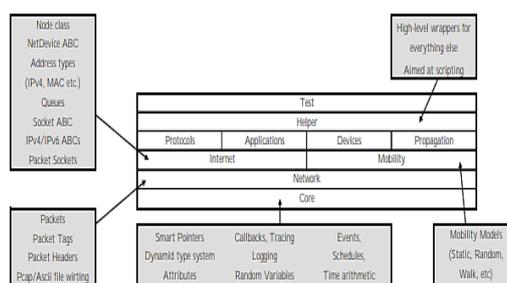


Ilustración 5: Estructura del simulador ns-3.
Fuente: Alejandro Pérez Palacio

Ns-3 en su estructura descrita en la ilustración 5, en general sigue un sistema de directorios, con lo que logra abarcar una alta gama de protocolos, desde la capa física, hasta el nivel de aplicación. La estructura del simulador está dividida en capas, la capa inferior incluye todo tipo de parámetros relacionados con el núcleo, (tiempos, eventos y atributos generales). Mientras asciende a capas superiores, entran en proceso los módulos más importantes como Internet, Network y Applications, etc (Palacio, 2016).

EstiNet

EstiNet se fundó como una empresa profesional de software de simulación de redes en 2011. Su primer producto simulador de red "EstiNet 9" recibió el "Premio a la Aplicación y Producto Destacado" en Taiwán en 2012. EstiNet se expandió como una empresa profesional de soluciones de red dedicada a productos basados en SDN y

soluciones para Enterprise, Service Provider y Private Cloud, desde 2013. Integra altamente la plataforma de software SDN de fuente abierta y los socios upstream/downstream para crear una plataforma de arquitectura abierta que ofrece automatizaciones más simples, inteligentes y eficientes redes para los clientes [22].

EstiNet Technologies Inc. es una compañía profesional de TI dedicada al desarrollo de productos y soluciones para redes definidas por software (SDN). Los principales financiadores de EstiNet incluyen a las grandes empresas de diseño de circuitos integrados de Taiwán. El producto simulador y emulador de red EstiNet SDN es compatible con el protocolo OpenFlow (EstiNet, EstiNet, s.f.).

Con esta herramienta puede simularse o emularse una red OpenFlow. Una buena propiedad de EstiNet es que utiliza la metodología del “kernel re-entering” (S. Wang and H. Kung, 1999), para permitirle a los programas de aplicación real no modificados funcionar en máquinas simuladas. Debido a esta capacidad, los resultados de la simulación de EstiNet son tan precisos como los resultados obtenidos de un emulador. EstiNet utiliza su propio reloj de simulación para controlar el orden de ejecución de eventos de simulación, pero generando resultados muy precisos.

EstiNet es una plataforma de red de simulación compatible con Linux Kernel (EstiNet, s.f.).

En una red OpenFlow simulada por EstiNet, programas de controlador OpenFlow reales como NOX/POX (NOX, 2015, s.f.), Ryu (RYU, 2014), y Floodlight (Floodlight, s.f.) se pueden ejecutar directamente en un host simulado para controlar switches OpenFlow simulados sin ninguna modificación. En cuanto a EstiNet como emulador, como cualquier emulador debe hacer, éste necesita llevar a cabo la emulación en tiempo real y permite que el programa de aplicación del controlador se ejecute en una máquina externa para controlar switches OpenFlow emulados. Debido a la utilización de la metodología del kernel re-entering, EstiNet también permite al programa del controlador y a los switches OpenFlow emulados ejecutarse en la misma máquina.

MiniNet

Mininet consiste en una herramienta de prueba y desarrollo para SDN, aprovechando los requisitos de los ordenadores de una manera óptima para lograr emular diferentes proyectos en un ambiente virtual. Utiliza una virtualización liviana para hacer que un solo sistema se vea como una red completa, ejecutando el mismo kernel, sistema y código de usuario logrando que un host Mininet se comporte como una máquina real (Software-Definet Networking: The New Norm for Networks, s.f.).

Características de Mininet

Mininet cuenta con características muy útiles para la simulación de redes algunas de ellas se mencionan a continuación:

Rápido: para comenzar con una red simple no lleva más que unos segundos con lo que el ciclo de correr, editar y depurar puede hacer bastante rápido (GitHub Introduction to Mininet, s.f.).

Permite la creación de topologías personalizadas: Lo que da la facilidad a la creatividad del usuario (GitHub Introduction to Mininet, s.f.).

Ejecución de programas reales: Permite utilizar cualquier herramienta del sistema operativo Linux (GitHub Introduction to Mininet, s.f.).

Permite personalizar el envío de paquetes: Esto se logra gracias el protocolo OpenFlow y los conmutadores Mininet, que son programables para poder personalizar la velocidad de los paquetes hacia los nodos (GitHub Introduction to Mininet, s.f.).

Facilidad de ejecución: Mininet tiene una facilidad de ejecución en varios entornos ya sea en un computador portátil, un servidor, una máquina virtual o en la nube (GitHub Introduction to Mininet, s.f.).

Permite compartir y replicar resultados: Una vez que el código esté empaquetado, cualquier persona puede ejecutar (GitHub Introduction to Mininet, s.f.).

Facilidad de uso: Mininet está basado en ejecución de scripts de Python lo que hace su fácil aprendizaje (GitHub Introduction to Mininet, s.f.).

Es un proyecto de código abierto: Al ser un proyecto de código abierto da la pauta para acudir a la comunidad Mininet en caso de una duda con los códigos desarrollados para su ejecución (GitHub Introduction to Mininet, s.f.).

Desarrollo activo: A través de actualizaciones constantes para la corrección de errores y mejoras para una mejor experiencia en el desarrollo de las redes (GitHub Introduction to Mininet, s.f.).

Componentes de una red con Mininet

Una red basada en Mininet está conformada básicamente por tres componentes, los cuales al trabajar en conjunto permitiendo que la red trabaje en una completa armonía. El primer componente de una red con mininet, son los hosts aislados que serán catalogados como los clientes de la red, cada uno de estos hosts suelen agruparse en un conjunto de procesos para obtener un espacio y adquirir de forma exclusiva interfaces, puertos y tablas de enrutamiento. Para lograr diferenciar estos grupos de hosts se suele dar un nombre a cada grupo y poder identificarlos (GitHub Introduction to Mininet, s.f.).

El segundo componente para la red son los enlaces emulados no son más que el medio que permitirá la conexión entre los interruptores y los hosts. Para poder asignar una tasa de datos a cada enlace se utiliza Linux Traffic Control (GitHub Introduction to Mininet, s.f.), el cual está formado por un conjunto de programadores de paquetes para dar forma al tráfico a una velocidad configurada previamente (Software-Defined Networking: The New Norm for Networks, s.f.).

Por último tenemos los switch emulados que cumplen la función de intercambiar los paquetes a través de las diferentes interfaces conectadas a los hosts aislados mediante los enlaces emulados. En esta etapa mininet utiliza Linux Brige u Open vSwitch ejecutados en modo kernel (modo privilegiado) o el modo usuario (programas de usuario) (GitHub Introduction to Mininet, s.f.).

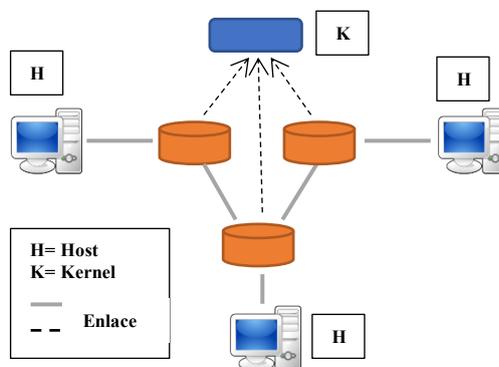


Ilustración 6. Estructura de una red mininet.

Fuente: Autores

Limitaciones de Mininet

Mininet es un entorno muy completo para inmiscuirse en las SDN pero también tiene sus limitaciones una de esta es que solo es posible ejecutarlo en entornos Linux lo que en caso de querer utilizar otras herramientas exclusivas de otros sistemas no será posible (GitHub Introduction to Mininet, s.f.). Otra limitación es que Mininet aunque esté basado en OpenFlow, el administrador debe configurar desde cero este controlador lo que puede ser una limitación para nuevos usuarios (GitHub Introduction to Mininet, s.f.).

Materiales y métodos

El presente trabajo presenta una investigación cualitativa, con la cual se pretende revelar un estudio del estado actual de la herramienta empleada para la configuración y simulación de entornos de trabajo en un entorno de red definida por software, con el planteamiento que permitan el modelamiento de tráfico que es uno de las métricas a considerar para garantizar calidad de servicio. Esto tiene como propósito de analizar las principales herramientas de uso que permitan obtener un esquema de red óptimo analizando las fortalezas y debilidades de las herramientas, así contribuir con la dinámica de impartición y transferencia de conocimientos.

Empleando este tipo de investigación se logró obtener características de la redes de ordenadores como soporte, virtualización de red, funcionalidad de la red, escalabilidad, rendimiento, entre otros que permita la mejor selección de herramientas para la simulación, tema que es de gran interés ante el auge de las redes SDN y que se volverán más predominante ante la implementación de redes de quinta generación, por su separación entre el plano de control y el plano de datos que facilita su gestión, y provocando un incremento de herramientas de simulación para este tipo de redes. El método utilizado para analizar de forma general el empleo de reglas, normas y protocolos de acción en las redes definidas por software fue el Analítico- Sintético. También se utilizó el método inductivo-deductivo en la investigación para obtener las conclusiones del fenómeno estudiado, con él se pudo extraer argumentos efectivos.

Resultados

Mediante la información recopilada, se realizó una tabla comparativa entre las herramientas mencionadas anteriormente, clasificando sus características de la siguiente manera:

Tabla 1. Tabla comparativa entre NS-3, EstiNet y Mininet

PARÁMETRO	NS-3	EstiNet	MiniNet
Modo de simulación	Si	Si	No
Modo de emulación	Si	No	Si
Compatible con controladores reales	Si	No	Si
Resultado repetible	Si	No	Si
Escalabilidad	Alta (un proceso)	Alta (un proceso)	Media (múltiples procesos)
Exactitud en los resultados de rendimiento	Si	No admite protocolo de árbol y ningún controlador real	Sin fidelidad de rendimiento
Soporte de GUI	Para configuración y observación	Sólo para observación	Sólo para observación
Lenguaje	C++ y Python	Sin información	Python
Sistema Operativo	Linux	Fedora	Linux
Openflow versión	Si tiene soporte	1.0, 1.3	1.0, 1.1, 1.2, 1.3
Licencia	Código abierto	Propietaria	Código abierto

Basándose en los dos primeros parámetros de la tabla 1 se puede observar que NS-3 tiene una ventaja sobre el resto en varias características analizadas, dado que

contiene los modos de emulación y simulación. En el tercer punto se habla de la compatibilidad con controladores reales en donde EstiNet no cuenta con esta característica, mientras Ns-3 y MiniNet sí. Uno de los puntos fuertes en estos entornos es la escalabilidad en los cuales NS-3 y EstiNet presentan un nivel alto, pero solo al ejecutar un proceso y Mininet muestra escalabilidad media pero puede ejecutar múltiples procesos a la vez.

En soporte a través una interfaz gráfica o GUI funciona de diferente manera en todos los entornos. En NS-3 se obtiene una interfaz gráfica para la configuración y observación. EstiNet y Mininet es usada solo para observación de la red y la parte de configuración se la hace a través de comandos.

En cuanto a la familiarización con los lenguajes donde el predominante entre los 3 entornos es C++ también se integra Python pero solo en Mininet y NS-3. Cabe acotar que en cuestión de rendimiento también existen diferencias pero eso depende del desarrollo sobre las redes SND que se quiera integrar y evaluar.

El simulador de redes NS3 es una versión reescrita de NS2, permite el desarrollo de modelos de simulación de alto desempeño, lo que habilita el uso de la emulación. En NS3 se tiene soporte para OpenFlow y NS2 tiene una funcionalidad más sencilla, sin embargo, no tiene soporte para OpenFlow, por lo cual es necesario escribir todo el script de código de la topología, flujos de información, monitoreo, protocolos y el comportamiento de la red para trabajar en un esquema SDN.

En EstiNet se requiere licencia aunque cuenta con un periodo de prueba con recursos limitados, a pesar de que tiene detalles de funcionamiento con algunos controladores como Floodlight, cuenta con herramientas de monitoreo y configuración de la red.

Mininet es la plataforma de emulación de código abierto y es una de las plataformas más utilizadas para trabajar con OpenFlow. Dada su naturaleza de código abierto, el soporte técnico y asesoramiento es proporcionado por los mismos usuarios y la comunidad de desarrollo.

Conclusiones

En la actualidad las redes se han tornado importante para el trato de la información y así mismo con la integración de cada vez más servicios de red para satisfacer las necesidades de muchos usuarios. Los administradores de red se enfrentan a un reto que con las redes SDN podríamos solucionar pero al ser un área que todavía está en desarrollo el uso de herramientas que logren emular cada diferente entorno de configuración. Las herramientas que se han presentado en este estudio logran en buena parte ayudar a comprender el estudio de estas redes. Como conclusión las herramientas presentadas van orientadas a diferentes tipos de usuarios en si se desea tener un inicio algo rápido en redes SDN se podría recomendar Mininet ya que su entorno lo hace más amigable con nuevos desarrolladores, pero si se desea para la

implementación de redes más complejas y robustas NS-3 sería una buena opción. Cabe acotar que este estudio se hizo en base las características de cada entorno, pero en el sentido de rendimiento y efectividad sería un punto que los usuarios deben tomar en cuenta.

Referencia

- (s.f.). Recuperado el 24 de 08 de 2018, de <https://www.nsnam.org/overview/key-technologies/>
- Christian Alexander Valdivieso Pinzón, C. D. (30 de 06 de 2017/06/30). SIMULACIÓN EN NS3 DEL PROBLEMA DENOMINADO CUELLO DE BOTELLA COMPARTIDO (SHARED BOTTLENECK) QUE SE PRESENTA EN EL PROTOCOLO MP-TCP. *RITI Journal*, 5(68), 10.
- E. Haleplidis, K. P. (Enero de 2015). *RFC 7426 Software-Defined Networking (SDN): Layers and Architecture Terminology*. Recuperado el 24 de 08 de 2018, de <https://www.rfc-editor.org/info/rfc7426>
- EstiNet*. (s.f.). Recuperado el 23 de 08 de 2018, de http://www.estinet.com/ns/?page_id=21169
- EstiNet*. (2015). *EstiNet Technologies*. Recuperado el 23 de 08 de 2018, de <http://www.estinet.com/es/>
- EstiNet*. (s.f.). *EstiNet*. (©2018 EstiNet Technologies Inc.) Recuperado el 24 de 08 de 2018, de http://www.estinet.com/es/?page_id=20620
- Floodlight. (s.f.). *Floodlight OpenFlow Controller*. Recuperado el 15 de 8 de 2018, de <http://www.projectfloodlight.org/floodlight/>
- Foundation, O. N. (Abril de 2012). *BibSonomy*. Recuperado el 23 de 08 de 2018, de <https://www.bibsonomy.org/bibtex/229da0c7fde89c4ce2405166a57d16009/chesteve>
- Foundation, O. O. (11 de 02 de 2014). *Migration Use Cases and Methods*. Recuperado el 23 de 08 de 2018, de <https://www.opennetworking.org/news-and-events/press-releases/open-networking-foundation-publishes-open-sdn-migration-use-cases-and-methods/>
- GitHub Introduction to Mininet*. (s.f.). Recuperado el 16 de 08 de 2018, de <https://github.com/mininet/mininet/wiki/Introduction-to-Mininet>
- H. Kim, N. F. (2013). Improving Network Management with Software. *IEEE Communications Magazine*, 51(2), 114-119.
- ITU. (2017). *ITU- ICT STATISTICS*. Recuperado el 15 de 08 de 2018, de <https://www.itu.int/en/ITU-D/Statistics/Pages/facts/default.aspx>
- ITU. (s.f.). *ITU*. Recuperado el 15 de 08 de 2018, de <https://www.itu.int/es/Pages/default.aspx>
- Masayoshi Kobayashi, N. M. (2014). Maduración de OpenFlow y redes definidas por software a través de implementaciones. *Diario Redes de computadoras: Revista Internacional de Redes de Computadores y Telecomunicaciones*, 61(C), 151-175.

-
- Mininet. (s.f.). *MiniNet: An Instant Virtual Network on your Laptop (or other PC) 2015*. Recuperado el 24 de 08 de 2018, de www.mininet.org.
- NOX, 2015. (s.f.). Recuperado el 15 de 8 de 2018, de <http://www.noxrepo.org/>
- ns-3. (s.f.). Recuperado el 20 de 08 de 2018, de <https://www.nsnam.org/>
- ns-3. (2011-17). (Copyright ©) Recuperado el 23 de 08 de 2018, de <https://www.nsnam.org/overview/what-is-ns-3/>
- nsman. (s.f.). *ns-3 Tutorial 3.14 version*. Recuperado el 17 de 08 de 2018, de <https://www.nsnam.org/docs/release/3.14/tutorial/ns-3>
- ONF. (s.f.). *Onf*. Recuperado el 24 de 08 de 2018, de <https://www.opennetworking.org/technical-communities/areas/specification/open-datapath/>
- Palacio, A. P. (Septiembre de 2016). *UCrea*. Recuperado el 24 de 08 de 2018, de <https://repositorio.unican.es/xmlui/bitstream/handle/10902/9052/386645.pdf?sequence=1&isAllowed=y>
- Porxas, A. X. (Diciembre de 2014). *Virtualization-enabled Adaptive Routing for QoS-aware Software-Defined (Master thesis)*. Recuperado el 23 de 08 de 2018, de <https://upcommons.upc.edu/handle/2099.1/24951>
- REYES, M. C., & CORTÉS, O. J. (2010). TENDENCIAS EN LA INTERCONEXIÓN DE REDES DE NUEVA GENERACIÓN. *Gerencia Tecnológica Informática*, 9(23).
- RYU. (2014). *Component-based software defined networking framework. Build SDN Agilely*. Recuperado el 15 de 8 de 2018, de <http://osrg.github.io/ryu/>
- S. Wang and H. Kung. (1999). A Simple Methodology for Constructing extensible and high-fidelity TCP/IP network simulators. *INFOCOM Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings.*, 3, 1134 - 1143.
- S.-Y. Wang, C.-L. Chou and C.-M. Yang. (2013). EstiNet openflow network simulator and emulator. *IEEE Communications Magazine*, 51(9), 110-117. Obtenido de <http://www.estinet.com/es/>
- Saldaña, J. M.-M. (2014). *Emulación de escenarios de red mediante un testbed*. Recuperado el 24 de 08 de 2018, de http://diec.unizar.es/~jsaldana/personal/testbed_URSI_2010_in_proc.pdf
- Software-Defined Networking: The New Norm for Networks*. (s.f.). (Open Networking Foundation, White Paper) Recuperado el 16 de 08 de 2018, de <https://www.opennetworking.org>
- Tseng, C. C. (2014). IPv6 operations and deployment scenarios over SDN of Asia-Pacific. Asia: Network Operation and Management Symposium (APNOMS) 2014.