ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS

INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA ELECTRÓNICA EN REDES Y COMUNICACIÓN DE DATOS

TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DE TÍTULO DE INGENIERO EN: ELECTRÓNICA EN REDES Y COMUNICACIÓN DE DATOS

TEMA: IMPLEMENTACIÓN DE LA TECNOLOGÍA SDN PARA CONTROL DE ACCESO Y CALIDAD DE SERVICIO EN REDES DOMÉSTICAS

AUTOR: PAZMIÑO CHICAIZA, JUAN EDUARDO

TUTOR: DR. CARRERA ERAZO, ENRIQUE VINICIO

SANGOLQUÍ - ECUADOR

2018

ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA ELECTRÓNICA EN REDES Y COMUNICACIÓN DE DATOS

CERTIFICACIÓN

Certifico que el trabajo de titulación, **"IMPLEMENTACIÓN DE LA TECNOLOGÍA SDN PARA CONTROL DE ACCESO Y CALIDAD DE SERVICIO EN REDES DOMÉSTICAS"** realizado por el señor ***JUAN EDUARDO PAZMIÑO CHICAIZA***, ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar al Señor ***JUAN EDUARDO PAZMIÑO CHICAIZA*** para que lo sustente públicamente.

Sangolquí, 08 de enero del 2018.

Ing. Enrique V. Carrera

ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

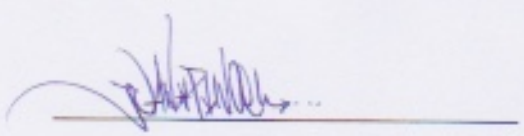
CARRERA DE INGENIERÍA ELECTRÓNICA EN REDES Y COMUNICACIÓN DE DATOS

AUTORÍA DE RESPONSABILIDAD

Yo, ***JUAN EDUARDO PAZMIÑO CHICAIZA***, con cédula de identidad N° 1718385154, declaro que este trabajo de titulación **"IMPLEMENTACIÓN DE LA TECNOLOGÍA SDN PARA CONTROL DE ACCESO Y CALIDAD DE SERVICIO EN REDES DOMÉSTICAS"** ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance de la investigación mencionada.

Sangolquí, 08 de enero del 2018.



Juan Eduardo Pazmiño Chicaiza

CI: 1718385154

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA ELECTRÓNICA EN REDES Y COMUNICACIÓN DE DATOS

AUTORIZACIÓN

Yo, ***JUAN EDUARDO PAZMIÑO CHICAIZA***, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca Virtual de la institución el presente trabajo de titulación **"IMPLEMENTACIÓN DE LA TECNOLOGÍA SDN PARA CONTROL DE ACCESO Y CALIDAD DE SERVICIO EN REDES DOMÉSTICAS"** cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Sangolquí, 08 de enero del 2018.

Juan Eduardo Pazmiño Chicaiza

CI: 1718385154

DEDICATORIA

A mi mamá por ser mi apoyo incondicional, amiga, y ejemplo de constancia, dedicación y superación.

A mi hermano que ha sido el mejor y más sincero amigo que he tenido.

AGRADECIMIENTO

Agradezco a mi mamá, por todo el sacrificio, amor y dedicación que ha sabido entregarme desde el comienzo de mi vida.

Sin ella estas páginas no existirían, gracias mamá.

Gracias a los amigos por brindarme su ayuda cuando más lo necesite y sobre todo por aguantarme.

Gracias a aquellos profesores que supieron transmitir sus conocimientos y experiencia.

ÍNDICE DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

En la actualidad, las redes domésticas han crecido tanto en cantidad de dispositivos como en aplicaciones que compiten por un ancho de banda sobre un único canal, degradando así su rendimiento. Este fenómeno visto desde el lado del usuario, se traduce en una red deficiente y lenta. Una solución es la implementación de la calidad de servicio y control de acceso; el primero permite asignar un determinado ancho de banda a cierto tipo de tráfico, mientras que el segundo se encarga de limitar de acuerdo a ciertas reglas la circulación de tráfico de aplicaciones no deseadas. Con dicho enfoque y el uso de la prometedora tecnología SDN, se propone la implementación de un sistema que clasifica el tráfico de acuerdo al tipo de aplicación y lo envía por un determinado enlace, además de bloquear la circulación de tráfico de aplicaciones no deseadas mediante el bloqueo de determinados puertos que estas utilizan para salir a Internet. Para la comprobación del funcionamiento y rendimiento del sistema propuesto dentro de una red doméstica se utilizaron dos topologías, una con la implementación SDN que se propone en este trabajo y otra sin ella; en cada una se realizaron dos análisis que comprenden la medición de parámetros como lo son el *jitter*, pérdida de paquetes, latencia y *throughput*. Con los resultados obtenidos se realizó una comparación entre ambas topologías, concluyendo que la implementación propuesta en este trabajo tiene una disminución del *jitter*, latencia y pérdida de paquetes en un 87,10%, 72,57% y 92,7% respectivamente, frente a una red doméstica tradicional.

Palabras Clave

- **SDN**
- **QoS**
- **CONTROL DE ACCESO**
- **OPENFLOW**

ABSTRACT

Nowadays, home networks have grown both in number of devices and applications that compete for bandwidth over a single channel, degrading their performance. This phenomenom viewed from the user side, this means a poor and slow network. One solution is the implementation of quality of service and access control, the first allows assigning a bandwidth to a certain type of traffic, while the second is responsible for limiting the traffic circulation of certain applications through network rules. With this approach and the use of promising SDN technology, we propose the implementation of a system that classifies traffic according to the type of application and sends it through a specific link, in addition to blocking traffic of unwanted applications by blocking certain ports that are used to go through Internet. To verify the functionality and performance of the proposed system within a home network, two topologies were used, one with the SDN implementation proposed in this work and the other without it; in each network two analysis were performed out, that include the measurement of parameters such as jitter, packet loss, latency and throughput. With the results obtained, a comparison was made between both topologies, concluding that the proposed implementation in this work has a decrease jitter, latency and packet loss by 87.10%, 72.57% and 92.7% respectively, compared to a traditional home network.

Keywords

- **SDN**
- **QoS**
- **ACCESS CONTROL**
- **OPENFLOW**

CAPÍTULO 1

INTRODUCCIÓN

1.1.ANTECEDENTES

Las redes definidas por software (SDN) son un potente paradigma de redes computacionales, cuyo objetivo es facilitar el diseño e implementación de redes altamente adaptables, escalables y fáciles de manejar, por tanto la idea central es separar el plano de control (software) del plano de datos (hardware). El plano de control está manejado mediante un controlador central que posee una vista global de toda la red, dejando de esta manera a los *switch* con la única funcionalidad de reenviar los paquetes (Boucadair, 2014) (Bianco, Giaccone, Mahmood, Ullio, & Vercellone, 2015).

Cuando se tiene una arquitectura programable, adaptable y económica como SDN es posible crear aplicaciones que extraigan toda la funcionalidad de la infraestructura que se encuentra debajo. El tener un control total sobre la red permite ajustar dinámicamente el flujo de tráfico para así satisfacer las cambiantes necesidades. Su uso con grandes anchos de banda lo hace ideal (Open Networking Foundation, 2012).

La gestión en una red en la que se usa la tecnología SDN es centralizada, el administrador y programador de la red tiene una vista global de la red, lo cual le permite implementar cualquier regla de tráfico de manera rápida y eficaz, todo esto desde un controlador SDN (Bianco, Giaccone, Mahmood, Ullio, & Vercellone, 2015).

En la actualidad debido a la reducción del impacto en la innovación de las redes ya sea por la gran cantidad de equipos instalados o la reticencia de experimentar con el tráfico de producción, las nuevas ideas simplemente quedan en fundamento teórico ya que no son probadas en un ambiente suficientemente realista como para llevarlo a una escala real de tráfico. Uno de los motivos por los cuales la virtualización de las

redes es tan importante, es que estas al permitir el ingreso de nuevas ideas incrementan la tasa de innovación en la infraestructura de red. La implementación de dichas innovaciones llegan a ser de bajo costo y alto rendimiento (McKeown, Anderson, Balakrishnan, Parulkar, Peterson, & Rexford, 2008).

Gracias a su naturaleza centralizada y el respaldo de muchos protocolos y arquitecturas que han sido propuestas para respaldar el paradigma SDN su implementación puede darse en redes tan pequeñas como lo es una red doméstica, considerándose así una alternativa a la implementación tradicional, todo esto con el fin de simplificar el manejo de la red. (Bianco, Giaccone, Mahmood, Ullio, & Vercellone, 2015).

La programabilidad que ofrecen las redes definidas por software hacen que sea posible el desarrollo de diversos tipos de aplicaciones, mediante las cuales es posible aplicar diferentes reglas o políticas de tráfico a toda la infraestructura de red, además de facilitar su manejo mediante interfaces graficas, evitando de este modo su configuración a bajo nivel (Hong, Ma, Banerjee, & Mao, 2016).

1.2.JUSTIFICACIÓN E IMPORTANCIA

En la actualidad las redes se han convertido en parte fundamental de la infraestructura de negocios, universidades así como de los hogares (McKeown, Anderson, Balakrishnan, Parulkar, Peterson, & Rexford, 2008).

Actualmente las redes domésticas poseen una mayor complejidad, así como la cantidad de dispositivos y servicios que prestan crece día a día, una red doméstica típica posee: computadores, consolas de video juegos, teléfonos y televisores inteligentes, todos ellos haciendo uso del mismo ancho de banda que la red proporciona para aplicaciones que han crecido en complejidad y por tanto son más exigentes (Gharakheili, Bass, Exton, & Sivaraman, 2014).

Las redes domésticas al crecer en complejidad, requieren mayores anchos de banda, ya que lo que más se consume dentro de los hogares es contenido multimedia tales como video, música en *streaming* así como juegos de video en línea, actualizaciones del PC, descargas varias, video chat, navegación web entre otros. Al tener tantos servicios disponibles y en funcionamiento se pueden crear cuellos de

botella, los cuales generaran lentitud o fallo de los diferentes servicios que se estén consumiendo, en general de cualquier actividad que requiera un medio o alto consumo de ancho de banda.

Es importante comprender el funcionamiento de la arquitectura SDN ya que su programabilidad permite crear diversos tipos de aplicaciones, las cuales podrían implementar políticas de red automáticamente o durante periodos de tiempo específicos, facilitando así el trabajo del administrador de red, además no solo ayudaría a mejorar la circulación de tráfico en redes domésticas también se lo podría hacer en redes de mayor escala y complejidad.

Actualmente la calidad de servicio en las redes abarca varios factores tales como confiabilidad, disponibilidad, retraso, escalabilidad entre otras. Estos factores junto al hecho de que no solo las redes corporativas sino las domésticas se han vuelto más complejas desempeñando un papel importante en como son manejados los paquetes dentro de la red, la tecnología SDN permite manejar todo el tráfico sin necesidad de equipos específicos que cuenten con capacidades de QoS (Wallner, 2013).

Históricamente la calidad de servicio ha sido una cuestión de gastos operativos y requerimientos diferenciados para cada tipo de usuario, muchas de los servicios actuales tales como *Netfix, Spotify* o juegos en línea así como tecnologías tales como VoIP, necesitan calidad de servicio y eficacia garantizada para su buen funcionamiento (Wallner, 2013).

1.3.ALCANCE DEL PROYECTO

El presente proyecto tiene como finalidad comprobar el funcionamiento de la tecnología SDN dentro de redes domésticas, para esto se definirá una red LAN con salida a Internet en la cual se implementará el control de acceso y calidad de servicio.

Para cumplir con los objetivos planteados en este proyecto se necesita un escenario real de pruebas para lo cual se hará uso de un router *Mikrotik* en el que se habilitará el funcionamiento del protocolo OpenFlow, además se hará uso de un controlador SDN y varios hosts.

La implementación SDN se llevará a cabo mediante un *script* que será ejecutado sobre el controlador SDN. Dentro de este *script* se definirán todos los parámetros necesarios tanto para el control de acceso como para la calidad de servicio (QoS).

Con los parámetros definidos dentro del script se pretende asegurar una tasa de datos en la red, un retardo, una variación de retardo (*jitter*) y un ancho de banda mínimo de acuerdo al que la red tenga acceso. Se definirán tres tipos de tráfico a los cuales el usuario podrá asignar un determinada tasa de datos dentro de la red.

Se evaluarán las ventajas y desventajas así como el desempeño de esta implementación SDN dentro de un entorno real en una red doméstica que posea una velocidad de carga y descarga de 0.95 Mbps y 8Mbps, respectivamente; para lo cual se realizarán pruebas en las que se verificarán parámetros como el throughput (velocidad real de transporte de datos), latencia (retardos temporales dentro de una red), jitter (retrasos de la señal), para esto se hará uso de herramientas como *IPERF* y *D-ITG* dentro de los hosts disponibles.

Para finalizar se realizará una comparación entre la red propuesta y una sin la tecnología SDN, los resultados obtenidos en esta comparación nos permitirán saber si el uso de dicha tecnología es conveniente o no en redes pequeñas.

1.4.OBJETIVOS

1.4.1. Objetivo general

Evaluar el uso la tecnología SDN para el control de acceso y calidad de servicio en redes domésticas.

1.4.2. Objetivos específicos

- Investigar y entender el funcionamiento de la tecnología SDN en las redes.
- Diseñar e implementar el prototipo de red LAN de prueba con la tecnología SDN.

- Diseñar y realizar pruebas de rendimiento, desempeño entre las diferentes topologías haciendo uso de herramientas como *IPERF* y *D-ITG*.
- Realizar una comparación del desempeño de la implementación del control de acceso y QoS en una red con la tecnología SDN y en otra que no la posea.
- Analizar los resultados obtenidos y verificar la fiabilidad del uso de la tecnología SDN en redes domésticas.

CAPÍTULO 2

MARCO TEÓRICO

2.1. REDES DEFINIDAS POR SOFTWARE (SDN)

2.1.1. Introducción

Las redes definidas por software son un nuevo paradigma utilizado para el manejo de redes de computadora. El objetivo es facilitar el desarrollo, implementación y manejo de la red, evitando así la gestión a bajo nivel (Open Networking Foundation, 2012).

En las redes Ethernet tradicionales tanto el plano de control como el plano de datos se encuentran en un mismo dispositivo de red (*e.g.*, Router), en una red SDN se separan ambos planos, dejando al equipo de red con la única funcionalidad de reenviar los paquetes, el plano de control es trasladado a un equipo diferente el cual toma las decisiones de reenvío de todos y cada uno de los paquetes que están en la red (Robertson, 2017).

La flexibilidad que entregan las redes definidas por software permiten que un dispositivo de red funcione como un hub, router o firewall, pudiendo así implementar diferentes servicios, protocolos y funcionalidades que solamente se tiene en equipos diferentes, además de permitir una rápida escalabilidad dentro de la red, consiguiendo un control centralizado (Kim & Feamster, 2013).

2.1.2. Características

Entre las principales características de las redes definidas por software se pueden destacar las siguientes:

- **Directamente programables**: gracias a que el controlador se encuentra separado de las funciones de reenvío es posible programar en la red diferentes políticas y protocolos.

- **Manejo centralizado:** el controlador SDN tiene una vista global de toda la red lo que le permite implementar cambios o actualizaciones de forma rápida y segura.
- **Estándares abiertos:** al implementar una red con la tecnología SDN se simplifica su manejo y operación ya que el controlador es el único que proporciona las instrucciones de manejo en lugar de tener varios dispositivos de diferentes marcas con comandos y protocolos propietarios (Open Networking Foundation, 2012).

2.1.3. Arquitectura

La arquitectura que se propone en una Red Definida por Software esta compuesta por tres planos que son: aplicación, control y datos; como se muestra en la Figura 1, los cuales permiten simplificar y optimizar el reenvío de paquetes, al mismo tiempo que se tiene la oportunidad de tomar ventaja de un determinado lenguaje de programación para entregar la funcionalidad deseada a toda la infraestructura de red (Open Networking Foundation, 2012) (OpenFlow Switch Specification, 2012).

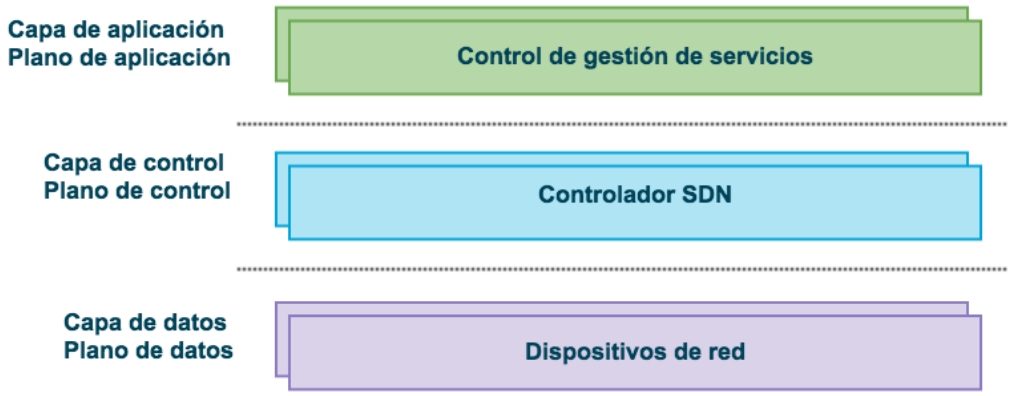


Figura 1. Arquitectura SDN.

A. Plano de aplicación

El plano de aplicación provee un método de soporte, mantenimiento y administración de toda red, es capaz de interactuar con cada regla y manejar como es enviado cada paquete, consiguiendo de esta forma un alto nivel de abstracción de la red, lo cual permite simplificar el proceso de configuración de la red.

B. Plano de control

Un controlador SDN es una entidad lógica centralizada que trabaja entre la aplicación y el plano de datos, se encarga de traducir las políticas de red que se predefinieron en la capa de aplicación a reglas de envió de paquetes y trasmitirlas a la infraestructura de red.

El controlador está basado en una serie de protocolos (*e.g.*, OpenFlow) que permiten manejar la red de manera centralizada de acuerdo a las necesidades y requerimientos del administrador. Desde el controlador se envían una serie de instrucciones y reglas que son instaladas en cada *switch* que se encuentra conectado a él, cada una de estas reglas o instrucciones determinan el funcionamiento general de los flujos que se envían y reciben dentro de la red controlada (Kim & Feamster, 2013) (Robertson, 2017) (Pinilla, 2015).

C. Plano de datos

El plano de datos representa toda la infraestructura de red agrupada en una o más sub redes que contienen un grupo específico de tráfico circulando entre sus interfaces.

Los dispositivos que comprenden una red SDN pueden ser físicos o virtuales, estos a su vez tienen la capacidad de mirar dentro de la trama Ethernet y usar esa información para determinar que regla o acción asociada usar, convirtiéndolos de este modo en equipos multicapa ya que pueden operar a través de la capa 1 a la capa 4 de la trama Ethernet como se muestra en la Figura 2 (Robertson, 2017).

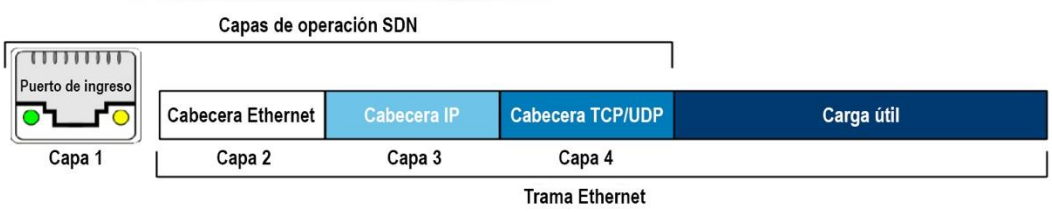


Figura 2. Trama Ethernet

Capa 1: es la llamada capa física en la que se definen los puertos físicos por donde los datos son recibidos y enviados.

Capa 2: Conocida como enlace de datos, en la que se establece y determina las conexiones entre los dispositivos físicos o virtuales, utiliza las direcciones MAC (media Access control) para identificar a cada dispositivo.

Capa 3: Llamada capa de red, es la encargada de proveer un mecanismo para transmitir datos de un nodo a otro, es responsable del reenvío de paquetes entre enrutadores que usan la información de direcciones del protocolo de Internet (IP).

Capa 4: Conocida como capa de transporte, es la responsable de establecer conexiones entre aplicaciones que corren en los dispositivos del host. Utiliza sockets para definir puntos finales dentro de la red, la dirección de un socket es la combinación de la dirección IP y el número del puerto. El número del puerto define el protocolo de comunicación y la aplicación asociada, los protocolos más usados son TCP y UDP

2.1.4. Protocolo OpenFlow

OpenFlow es un protocolo que permite a un servidor enviar instrucciones, reglas y políticas de red que serán instalados en la infraestructura de red. A diferencia de una red convencional en donde cada marca tiene sus propias configuraciones y protocolos de comunicación, OpenFlow permite centralizar el manejo y configuración de la red de modo que esta se pueda configurar independientemente de la marca o fabricante de los equipos de comunicación, de tal manera que la funcionalidad de los mencionados equipos se ve limitada por las reglas de reenvío de paquetes que el administrador de red implemente.

OpenFlow ofrece reglas de flujo con las que gestiona las tablas de ruta y la configuración de los dispositivos que forman parte de la infraestructura de red, las entradas que pueden ser gestionadas se muestran en la Figura 3 (OpenFlow Switch Specification, 2012), dichas entradas junto a las reglas de flujo permiten el reenvío de paquetes a puertos o host específicos que se encuentran en una determinada VLAN, además de la encapsulación y reenvío de paquetes al controlador (Siebertz, 2014) (McKeown, Anderson, Balakrishnan, Parulkar, Peterson, & Rexford, 2008).

Figura 3. Ejemplos de las entradas de las reglas de flujo en un switch OpenFlow.

Gracias al protocolo OpenFlow, el controlador SDN puede agregar, actualizar y eliminar entradas de flujo, tanto de manera reactiva (en respuesta a los paquetes) como de manera proactiva (Siebertz, 2014).

2.2. CONTROLADORES SDN

En una red definida por software los controladores SDN son usados para manejar las interfaces físicas y virtuales de la infraestructura de red, de acuerdo a los parámetros o instrucciones definidas en la aplicación.

El controlador SDN es el cerebro y parte central de la red ya que es el encargado de instalar, actualizar e implementar las reglas y políticas sobre el reenvío de los paquetes en los diferentes dispositivos que forman parte de la red, dejando de esta forma a la infraestructura de red con la única funcionalidad de reenviar paquetes.

En la actualidad existe un variado número de controladores SDN tanto de código abierto como comerciales los cuales se listan en las Tablas 1 y 2 respectivamente, cada uno de ellos esta escrito en diferentes lenguajes de programación y está diseñado para permitir la participación de aplicaciones de terceros así como su modificación y desarrollo (Centeno, Vergel, & Calderón, 2014). Dichas aplicaciones desarrolladas para un controlador SDN en específico no pueden ser ejecutadas junto a un controlador diferente, ya que para su desarrollo se utilizan módulos específicos así como un diferente lenguaje de programación (Centeno, Vergel, & Calderón, 2014) (Pinilla, 2015).

Tabla 1.
Ejemplos de controladores SDN de código abierto.

CÓDIGO ABIERTO		
Controlador	**Lenguaje**	**Descripción**
OpenDayLight	Java	Es un proyecto de código abierto promovido por la fundación Linux.
Floodlight	Java, Python	Proyecto de código abierto con soporte de Big Switch Networks, permite implementar aplicaciones directamente sobre el controlador y manejarlas desde la interfaz gráfica de este.
RYU	Python	Controlador que proporciona una API bien definida para un rápido desarrollo, no posee una interfaz gráfica que facilite su uso.
POX	Python	Es un controlador multiplataforma ya que su único requisito para funcionar es tener instalado una de las versiones python.

Tabla 2.
Ejemplos de controladores SDN comerciales

COMERCIALES		
Controlador	**Desarrollador**	**Descripción**
APIC	CISCO	Su objetivo principal es proveer políticas y mecanismos de resolución de políticas a los dispositivos CISCO.
VAN	HP	Proporciona un punto de control para la administración y orquestación de la red, se adapta a las versiones 1.0 y 1.3 del protocolo OpenFlow, además de dar soporte para más de 50 modelos de *switchs* con OpenFlow habilitado.
NSX	VMware	Integra seguridad, administración, funcionalidad, control de máquinas virtuales, directamente desde su hipervisor, en el cual se pueden crear complejas arquitecturas de red, que incluye los servicios de red de la capa 2 a la capa 7.

Algunas de las características más relevantes de los controladores SDN son las que se listan a continuación.

- Poseen un modelo de datos de alto nivel que captura las relaciones entre los recursos gestionados, las políticas y otros servicios prestados por el controlador.
- Mecanismo de descubrimiento de dispositivos, topología y servicio además de sistemas de cálculo de ruta.
- Sesión de control segura sobre el Protocolo de Control de Trasmisión (TCP) entre el controlador y los agentes asociados en los elementos de la red.

- Entornos de desarrollo robustos que permiten la expansión de las capacidades básicas del núcleo y la posterior publicación de las APIs para los nuevos módulos.

2.2.1. Controlador POX

POX es un software de código abierto escrito en Python para redes definidas por software, este controlador requiere para su funcionamiento Python 2.7 o Python 2.6 y puede funcionar bajo Linux, Mac OS, Windows o cualquier sistema operativo que cuente con alguna de las versiones de Python antes mencionadas (Dandekar, Kharade, & Tathare, 2015).

El controlador POX utiliza la versión 1.0 del protocolo OpenFlow para comunicarse con la infraestructura de red, además incluye soporte para Open vSwitch, no cuenta con una interfaz gráfica donde se puedan correr las aplicaciones así que el manejo se lo realiza mediante líneas de comando, ejecutando una varias aplicaciones relacionadas a la vez (Siebertz, 2014).

2.3. SWITCH OPENFLOW

Un Switch OpenFlow el cual se muestra en la Figura 4, consiste en una o más tablas de flujos y un grupo de tablas que realizan búsquedas de paquetes y reenvío a través del protocolo OpenFlow a un controlador externo, el switch OpenFlow se comunica con el controlador y el controlador lo gestiona a través del protocolo OpenFlow (OpenFlow Switch Specification, 2012).

- **Canal Seguro:** Permite conectar el controlador al *switch*, permitiendo que comandos y paquetes se puedan enviar entre estos dos mediante el protocolo OpenFlow.
- **Tablas de flujos:** Contiene las entradas de flujo que proporcionan adaptación, reenvío y modificación de paquetes en el switch OpenFlow. Cada tabla de flujo en el Open vSwitch contiene un conjunto de entradas de flujo, cada entrada de flujo consiste en campos de coincidencia, contadores y un conjunto de instrucciones para aplicar a paquetes coincidentes.

- **Tabla de grupo:** Consta de entradas de grupo. La capacidad de una entrada de flujo para apuntar a un grupo permite al protocolo OpenFlow representar métodos adicionales de reenvío.

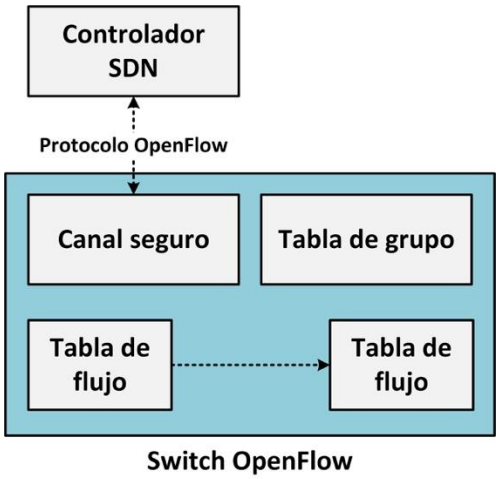


Figura 4. Componentes principales de un switch OpenFlow.

Un Switch OpenFlow puede ser de dos tipos: basado en software o basado en hardware; los *switchs* basados en software pueden ser: NetFPGA switch, Open vSwitch, OpenWrt, entre otros.

2.3.1. OpenWrt

OpenWrt es un firmware basado en una distribución de Linux para dispositivos usados en redes domésticas, el cual proporciona un sistema de archivos con escritura completa y administración de paquetes dándole una amplia flexibilidad de configuración (Seddiki, Muhammad, & Donovan, 2014) (OpenWRT, 2017).

Inicialmente toda la configuración se la puede realizar mediante la interfaz de línea de comandos utilizada en Linux, modificando y editando sus archivos de configuración, aunque también existe una interfaz WEB para usuarios menos avanzados, pudiendo desde ella agregar o eliminar paquetes, así como detener procesos y ver gráficos en tiempo real del tráfico que circula en las interfaces del dispositivo (OpenWRT, 2017).

OpenWrt utiliza fuentes del kernel GNU/Linux oficiales y solamente agrega parches relacionados a los SoC (System on Chip) y controladores de las interfaces de

red, todo el código propietario es re-implementado dentro de los archivos *tar* suministrados por los diferentes fabricantes (OpenWRT, 2017).

Los componentes de OpenWrt han sido optimizados para tener un tamaño suficientemente pequeños como para caber en el limitado espacio de memoria que poseen los dispositivos de red domésticos.

Originalmente el soporte estuvo limitado al modelo *Linksys WRT54G*, pero gracias a su rápida expansión en la actualidad existe una amplia lista de dispositivos así como de fabricantes que lo soportan, entre los más conocidos están: *MikroTik*, *TP-Link*, *D-Link*, *3Com*, *Netgear*, entre otros (OpenWRT, 2017).

2.3.2. Open vSwitch

Es una implementación de código abierto de un switch virtual, multicapa, programable que tiene como objetivo principal proveer un stack de switching para la virtualización de hardware (OpenFlow Switch Specification, 2012).

Open vSwitch está en capacidad de reenviar el tráfico entre diferentes máquinas físicas o virtuales dentro del mismo host físico donde se encuentra instalado como se muestra en la Figura 5, haciendo uso tanto de las interfaces de red físicas como virtuales.

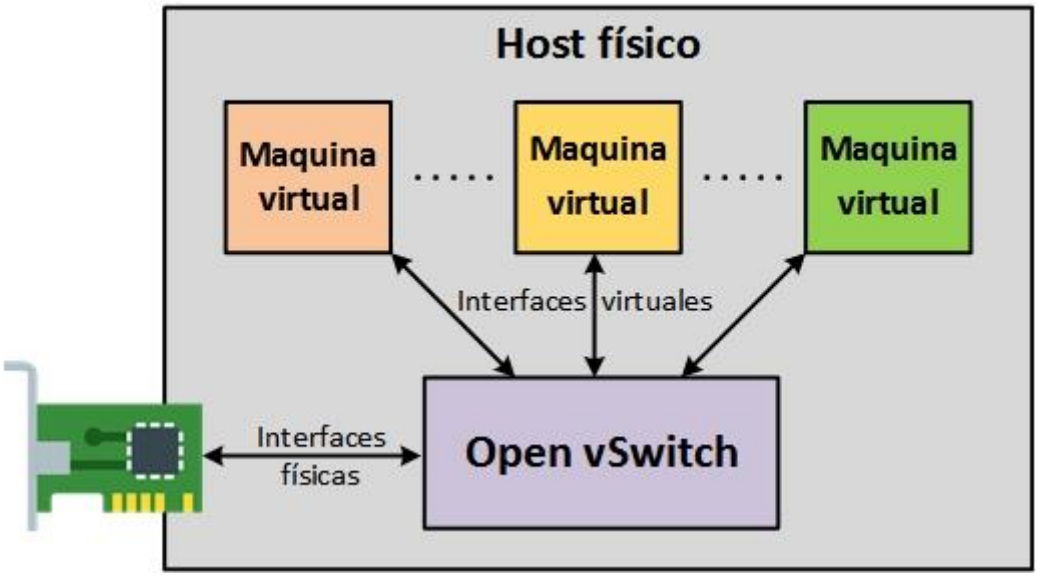


Figura 5. Open vSwitch dentro de un host físico.

Su diseño esta pensado para permitir la automatización masiva de la red a través de una extensión programable, al mismo tiempo que sigue soportando interfaces y protocolos estándar de administración.

Entre sus características principales se encuentran:

- Visibilidad entre equipos virtualizados a través de: NetFlow, túneles GRE, IPFIX, SPAN, RSPAN y sFlow.
- Soporta el estándar 802.1Q para vlans.
- Políticas de tráfico entre host virtualizados y físicos.
- Soporte para IPv6
- Múltiples protocolos de *tunneling* (GRE, VXLAN, STT)
- Configuración remota a través de lenguajes de programación como C, Python o Java.

2.4. CALIDAD DE SERVICIO

Las redes operan en base al mejor esfuerzo, esto quiere decir que lo único que se trata es enviar los datos sin preocuparse si llegan o no. En el mejor esfuerzo o *best effort* todo el tráfico que circula en la red tiene la misma prioridad, por tanto si llegan a existir cuellos de botella o congestión todo ese tráfico corre el riesgo de ser desechado.

Al existir la calidad de servicio o QoS se diferencia cada flujo de tráfico dando prioridad a uno o a otro de acuerdo a los requerimientos del usuario, de este modo se logra tener un rendimiento promedio sin afectar la experiencia del usuario.

En al actualidad la calidad de servicio es muy importante ya que varias tecnologías han sido desarrolladas e implementadas sobre las redes de computadoras, un ejemplo es la telefonía IP, video y música en *streaming*, los cuales no deben tener retrasos por lo que se debe asegurar un ancho de banda.

En los últimos años el incremento del uso de Internet ya sea con propósitos comerciales o privados y la cantidad de datos que se transmite ha aumentado exponencialmente es por esto que la calidad de servicio juega un papel fundamental asegurando los recursos necesarios de red para cierto tipo de aplicaciones, esto se

traduce en una adecuada tasa de datos, retardo y variación de retardo. Al existir diferentes tipos de tráfico que hacen un uso intensivo del ancho de banda en necesario dar prioridad a los que según las necesidades del usuario final es más importantes que el resto y debe tener un trato preferencial dentro de la LAN.

En una red doméstica tradicional las aplicaciones que pueden ver comprometido su rendimiento son aquellas que tienen envió y recepción de datos en tiempo real de baja latencia y alto consumo de ancho de banda, como juegos de video en línea o servicios de *streaming* de música y video, demás de aplicaciones de VoIP.

2.4.1. Control de tráfico

El control de tráfico es un conjunto de mecanismos utilizado por los enrutadores, los cuales deciden que hacer con los paquetes que se reciben y transmiten a través de las interfaces, esto incluye decidir que paquetes son aceptados o no, la velocidad y orden de entrada o salida de mismos (Brown, 2006).

Una de las ventajas del control de tráfico es el uso predecible de los recursos de la red, esto quiere decir que simultáneamente se puede asignar de acuerdo a la prioridad del tráfico una determinada cantidad del ancho de banda, sin afectar su rendimiento.

El control de tráfico se encuentra implementado en el *kernel* de los sistemas Linux como una utilidad, algunos de sus componentes son los siguientes:

- **Shaping**

El Shaping es un mecanismo de control de tráfico por el cual los paquetes se retrasan antes de su transmisión con el objetivo de evitar la sobrecarga de la red, esta es una las formas más utilizadas cuando se busca soluciones de control de ancho de banda, ya que limita o raciona el flujo tráfico para no superar una cantidad establecida, consiguiendo de esta forma controlar la latencia y garantizar un ancho de banda para la transferencia de datos (Brown, 2006) (Seddiki, Muhammad, & Donovan, 2014).

Este método se utiliza comúnmente en los bordes de la red para controlar el tráfico que entra así como también es una práctica muy usada por lo proveedores de Internet (Brown, 2006).

- **Scheduling**

Scheduling es un mecanismo mediante el cual los paquetes se organizan o reordenan entre la entrada o salida de una cola en particular, el más común es el llamado FIFO que quiere decir, el primero en entrar es el primero en salir, este reordenamiento también se denomina prioridad y ocurre a salida del tráfico, de este modo se intenta evitar que una sola aplicación o cliente acapare todo el uso de la red.

- **Clasificación**

El clasificador es el encargado de separar el tráfico en colas para posteriormente darles un tratamiento diferente, este proceso de clasificación se lo puede realizar durante la aceptación, enrutamiento y transmisión de un paquete en el dispositivo que recibe el tráfico, de este modo se realiza un efectivo control de tráfico con cada paquete que se recibe y envía.

- **Politicas**

Las políticas son un elemento de control de tráfico, por el cual se limita su flujo, es muy utilizado en el borde de la red con el fin de garantizar que una determinada cola no consuma más ancho de banda del que tiene asignado, este mecanismo aceptara el tráfico que cumpla con los parámetros establecidos, con el resto se realizará otro tipo acción el cual puede ser desechar los paquetes.

- **Droping**

Es un mecanismo por el cual se descarta por completo un flujo que no cumple con los parámetros establecidos, ya sea a la entrada o salida.

2.5.CONTROL DE ACCESO A LA RED

El control de acceso en la red es un conjunto de protocolos que tienen como finalidad implementar políticas de seguridad sobre la red y recursos que esta ofrece,

antes de permitir el acceso a los hosts. Para conseguir los niveles de seguridad deseados se implementan reglas o políticas que incluyen pre-admisión, autenticación, autorización de dispositivos conectados a la red y controles de admisión sobre los diferentes recursos con los que esta cuenta.

Cuando un equipo se conecta a una red no se le permite el acceso a los recursos, a menos que cumpla con una política establecida por el administrador de dicha red, esto es usado a nivel empresarial en donde se dan diferentes niveles de acceso a cada tipo de usuario.

Las redes de computadora domésticas es su mayoría se implementan sin un mecanismo que restrinjan o controlen el acceso a los recursos informáticos, con el control de acceso se pueden corregir estos errores y de este forma evitar el acceso de host no deseados a la red.

El proceso de autenticación se lo hace a través del estándar IEEE 802.1X, el cual es un control de acceso basado en puertos, permite la autenticación de dispositivos conectados a la red LAN tanto el redes alámbricas como inalámbricas, para esto establece una conexión punto a punto o impidiendo la conexión a ese puerto si la autenticación falla.

CAPÍTULO 3

DISEÑO E IMPLEMENTACIÓN DE LA RED

En esta sección se detalla tanto el software como el hardware necesarios para el funcionamiento de la red con la tecnología SDN, además de las herramientas para la inyección de tráfico, test de penetración y una explicación general del funcionamiento y ejecución de los módulos que comprenden el sistema implementado. Posteriormente se describe la configuración necesaria del ruteador *Mikrotik* y el controlador SDN, la cual fue realizada en su totalidad de forma remota, utilizando una conexión via *ssh* para ingresar a cada equipo y editar los archivos de configuración necesarios.

3.1. CONTROLADOR SDN

3.1.1. Hardware

La plataforma escogida para el funcionamiento del controlador SDN fue una Raspberry Pi modelo B, la cual cuenta con una velocidad de procesamiento de 700MHz y 512MB de RAM, lo cual es suficiente para ejecutar con eficiencia las aplicaciones que realizarán el control de acceso y calidad de servicio sobre la red.

3.1.2. Software

El controlador SDN escogido fue POX el cual funciona sobre el sistema operativo Raspbian Jessie – Lite, los requerimientos de hardware necesarios para el correcto funcionamiento del controlador SDN son mínimos y no sobrepasan la capacidad de procesamiento de la Raspberry Pi, pudiendo ser ejecutado en cualquier sistema operativo que cuente con Python 2.6 o superior. Al ser un proyecto open source cuenta con una amplia comunidad de desarrollo que brinda soporte y actualizaciones.

3.1.3. Configuración

El controlador SDN fue instalado y ejecutado sobre una Raspberry Pi, en la

cual se configuró una IP estática, se instaló PHP5 y se cargaron los módulos de calidad de servicio y control de acceso para su posterior uso.

3.2. RUTEADOR

Se utilizó el ruteador Mikrotik 750GL al cual previamente se reemplazó el firmware que viene de fábrica por OpenWRT, en el cual se habilitó el protocolo OpenFlow e instaló el paquete Open vSwitch. Gracias a que el ruteador cuenta con 64MB de memoria y una velocidad de procesamiento de 400MHz, fue posible instalar y ejecutar una gran cantidad de aplicativos sin que su desempeño y funcionamiento se vea afectado.

En el Anexo 1 se detallan todos los comandos y pasos necesarios para reemplazar el firmware de fabrica e instalar OpenWRT.

3.2.1. Configuración básica

3.2.2. VLANS

Se crearon tres VLANs, WAN, LAN y CTRL, las dos primeras serán las redes a controlar, la tercera será la VLAN controladora en donde únicamente funciona el controlador SDN. En la Tabla 3 se listan las VLANs y sus correspondientes puertos junto a una descripción.

Tabla 3.
Descripción de puertos y vlans.

VLAN		PUERTOS	DESCRIPCIÓN
1	WAN	1	Este puerto esta configurado como cliente DHCP, se lo conecta directamente al equipo de cable modem entregado por el proveedor de servicios de Internet.
2	LAN	2	A esta red se conectan todos los host que harán uso de los recursos proporcionados por la red.
		3	
3	CTRL	4	VLAN del controlador SDN
-	-	5	Se dejo un puerto libre para futuras configuraciones.

3.2.3. Firewall

Se configura el firewall con el propósito de dar a cada una de las VLANs los permisos necesarios para comunicarse entre ellas, además de permitirles el acceso a Internet, para esto se debe configurar la opción de *forwarding* en la cual se ingresan las VLANs de origen y destino como se muestra en la Figura 6.

```
config forwarding
        option src              lan
        option dest             wan
```

Figura 6. Captura de configuración.

3.2.4. Open vSwitch

Los *switch* virtuales o también llamados Open vSwitch serán los encargados de realizar el reenvío de flujos de acuerdo a las reglas instaladas en ellos mediante el controlador SDN. Estas reglas incluyen el bloqueo o reenvío de flujos a través de puertos determinados, lo cual corresponde al control de acceso, así como asignar un determinado ancho de banda para cada enlace por el cual circulará un determinado tipo de tráfico.

Antes de su configuración en necesario verificar si se tiene instalada una de las versiones de Open vSwitch así como los paquetes *veth.ko* y *sch_tbf.ko*, estos últimos serán necesarios para la creación de los enlaces virtuales utilizadas en la calidad de servicio.

Mediante una conexión vía *SSH* se ingresó al ruteador, y se configuraron dos Open vSwitch y seis interfaces virtuales, tres serán para el Open vSwitch 1 y las restantes para el segundo, cada interfaz virtual será conectada con su correspondiente par, consiguiendo así los tres enlaces virtuales, adicionalmente se agregaron los puertos físicos de la LAN y la WAN a cada Open vSwitch respectivamente.

Finalmente en cada Open vSwitch se configuró la dirección IP y puerto en el cual funciona el controlador SDN, consiguiendo así su correcta conexión.

3.3. ESCENARIOS DE PRUEBA

3.3.1. Red sin la tecnología SDN

La red sin la tecnología SDN que se muestra en la Figura 7 cuenta con un ruteador TP-Link WR740N, al cual se le configuró el control de puertos, usado para las pruebas de calidad de servicio y acceso remoto, adicionalmente se configuró el control de acceso disponible en el router a fin de bloquear el tráfico proveniente de un host específico.

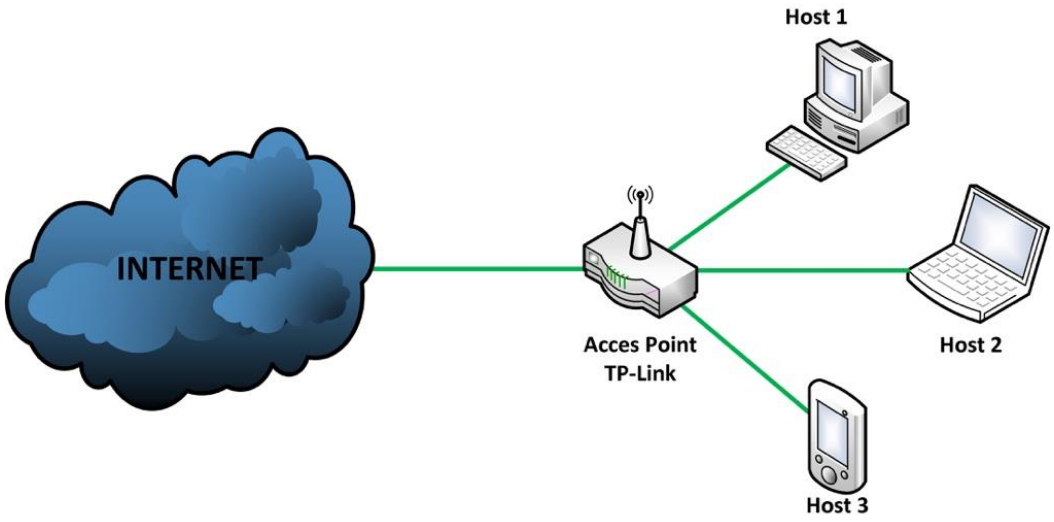


Figura 7. Topología sin la tecnología SDN.

3.3.2. Red con la tecnología SDN

La implementación de la red con la tecnología SDN consta de una topología como la que se muestra en la Figura 8, en ella se crearon 3 VLANs, las 2 primeras (LAN y WAN) serán las redes a controlar y la última será la red controladora, donde únicamente funciona el controlador SDN.

Junto al controlador SDN se ejecutará el script necesario para realizar tanto el control de acceso como la calidad de servicio (QoS), adicionalmente se configuraron en el ruteador todos los parámetros necesarios para la conexión y funcionamiento del Open vSwitch con el controlador SDN.

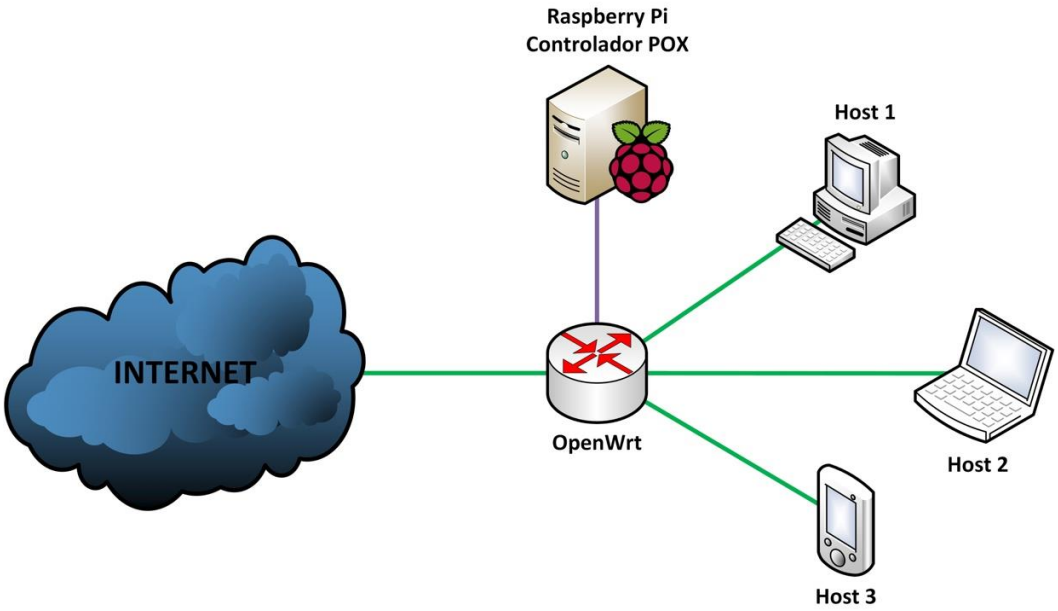


Figura 8. Topología con la tecnología SDN.

3.4.GENERADORES DE TRÁFICO

Un generador de tráfico es un programa informático que sirve para inyectar un determinado o determinados tipos de tráfico dentro de una o varias redes con la finalidad de medir su rendimiento y calidad de la misma.

3.4.1. IPERF

Esta herramienta de código abierto escrita en C++, fue utilizada para medir el rendimiento de la comunicación entre dos extremos, a través de la creación de flujos de datos TCP y UDP.

Gracias a que *IPERF* puede ejecutarse en varias plataformas, y funcionar como cliente o servidor, devolviendo medidas de tiempo estandarizadas con la cantidad de datos transmitidos y el rendimiento medio, fue posible verificar el aumento o disminución del rendimiento de la red.

3.4.2. D-ITG

Para la generación de tráfico correspondiente a VoIP se utilizó la herramienta *D-ITG*, ya que imita el comportamiento de aplicaciones de nivel superior, siendo capaz de obtener métricas del rendimiento, retardo, *jitter* y pérdida de paquetes con gran exactitud.

Gracias a que *D-ITG* cuenta con soporte para varios sistemas operativos fue posible realizar pruebas de rendimiento utilizando tanto Windows como Linux.

3.5. HERRAMIENTA DE PENETRACIÓN NMAP

Nmap es una herramienta de penetración que sirve para realizar un escaneo profundo de todos puertos abiertos, cerrados o filtrados en la red, se la utilizó para realizar las pruebas de control de acceso, identificando el estado de los diferentes puertos.

Adicionalmente Nmap brinda toda la información posible sobre los host conectados a la red como lo es su dirección mac, dirección IP, además de identificar el tipo, marca del equipo y sistema operativo, pudiendo ser este un servidor, PC, Celular, Router, o cualquier dispositivo que haga uso de los recursos de la red.

3.6.DESCRIPCIÓN GENERAL DEL SISTEMA

En la Figura 9 se muestra la arquitectura general del sistema implementado, el cual consta de dos módulos, el primero es usado para el control de acceso y el segundo para la calidad de servicio.

El sistema implementado permite a los usuarios configurar una determinado porcentaje del ancho de banda disponible en la red para los tres tipos de aplicación que se tomaron en cuenta (Web, VoIP y video), además de bloquear puertos TCP/UDP usados por dichas aplicaciones dentro de la red.

El funcionamiento del sistema parte de la conexión entre el controlador SDN y el ruteador a través del protocolo OpenFlow, mediante la ejecución los módulos de

calidad de servicio y control de acceso, los cuales instalan reglas de reenvío de flujos sobre el Open vSwitch, dando la funcionalidad deseada a la red.

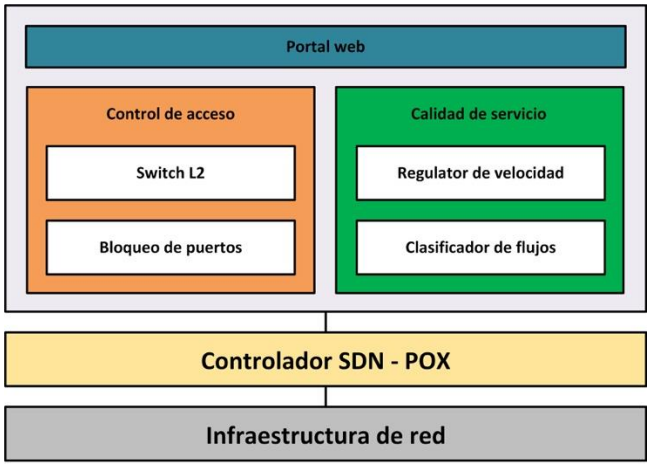


Figura 9. Arquitectura implementada.

3.6.1. CALIDAD DE SERVICIO

La implementación de la calidad de servicio permite al usuario definir un ancho de banda para cada tipo de aplicación, especificando un porcentaje del ancho de banda, según sus necesidades. La configuración se la realiza a través de un portal web que genera un archivo de configuración, este archivo es utilizado por el regulador de velocidad para implementar los respectivos anchos de banda a cada conexión.

3.6.2. Clasificador de flujos

El clasificador de flujos mantiene una tabla con la dirección ip de origen, dirección ip de destino, protocolo, puerto de origen y puerto de destino. Cuando el ruteador recibe un flujo envía una copia del primer paquete del flujo al controlador, donde el módulo de calidad de servicio determina el tipo de aplicación al que corresponde, pudiendo ser web, video o VoIP.

La verificación del tipo de aplicación se la hace a través de la librería *Libprotoident*, esta realiza una identificación del protocolo de capa de aplicación para los flujos, haciendo uso de los cuatro primeros bytes de carga útil.

3.6.3. Libprotoident

Libprotoidet es una librería que realiza una identificación del protocolo de capa de aplicación, a diferencia de otras técnicas de identificación que requieren capturar toda la carga útil del paquete. Libprotoident utiliza los cuatro bytes de carga útil enviados, el tamaño del primer paquete y los números de puerto TCP/UDP para dicho flujo (Alcock & Nelson, 2012).

- **Carga útil:** los cuatro bytes de carga útil registrados para una dirección se comparan con un patrón de carga útil conocido para el protocolo.
- **Tamaño de carga útil:** Este tipo de regla requiere que la cantidad de carga identificada en el primer paquete portador de carga corresponda a un tamaño determinado, un ejemplo son las reglas de Skype, las cuales requieren que la carga útil inicial en una dirección sea exactamente de 11 bytes.
- **Número de puerto:** Los números de puerto se utilizan para eliminar la ambigüedad de los casos en los que la carga útil podría coincidir con varias reglas o para fortalecer las reglas que de otro modo serian débiles y tendrían posibilidad de falsos positivos.

3.6.4. Regulador de velocidad

El regulador de velocidad es el encargado de asignar a cada flujo una apropiada tasa de velocidad, para conseguirlo se configuran dos Open vSwitchs dentro del ruteador los cuales se muestran en la Figura 10, cada uno de ellos esta conectado a través de los tres enlaces o conexiones virtuales configurados previamente con un ancho de de banda específico.

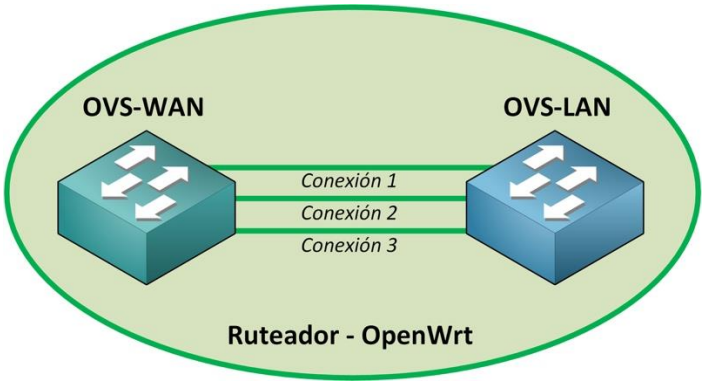


Figura 10. Conexiones virtuales.

Cada conexión entre los Open vSwitch corresponde a un grupo de aplicaciones diferente, cuando un flujo llega al Open vSwitch este lo reenvía al controlador SDN, en donde es clasificado, una vez hecho esto, el controlador instala las reglas de reenvío en el Open vSwitch de tal forma que cada flujo que llega es reenviado a través de su correspondiente conexión, las cuales ya han sido previamente configuradas con una tasa de velocidad para cada tipo de aplicación.

3.7. CONTROL DE ACCESO

El control de acceso consiste en bloquear los puertos TCP/UDP a través de los cuales las aplicaciones envían o reciben tráfico, de esta forma se consigue bloquear su funcionamiento y uso de recursos de la red, para esto el controlador SDN instala en el Open vSwitch reglas de reenvío de paquetes. Se parte de dos módulos escritos en Python: *l2_learning.py* y *blocker.py*, el primero funciona como un *switch* de capa 2, el cual es el encargado del reenvío de flujos a través de la red, el segundo módulo analiza los flujos, si estos coinciden con los puertos que el usuario definió para ser bloqueados detiene el proceso de reenvío de flujos del primer módulo, de este modo se evita que dichos flujos lleguen a su destino.

CAPÍTULO 4

RESULTADOS Y EVALUACIÓN

En el presente capítulo fueron realizados dos análisis, para cada análisis fueron utilizados dos escenarios, el primer escenario es una red con la tecnología SDN y el segundo una red sin dicha tecnología, los parámetros tomados para medir el rendimiento fueron throughput, latencia, jitter y pérdida de paquetes. Una vez finalizadas las pruebas junto a su análisis y respectiva comparación se espera obtener una mejora en el rendimiento de la red, ya sea por la disminución o aumento de los parámetros mencionados anteriormente.

A cada prueba ejecutada se realizó un análisis diferente; el primero al cual se lo denomino *Escaneo de puertos en la red*, tiene como objetivo evaluar mediante el uso de las herramientas NMAP y PAPING el correcto funcionamiento del control de acceso, para esto se cierran determinados puertos TCP con el objetivo de impedir el paso de tráfico a través de ellos, posteriormente se determinará si realmente los puertos están cerrados o no mediantes el uso de las herramientas ya mencionadas.

En el segundo análisis denominado *Rendimiento de la red*, se evalúa la calidad de servicio en referencia a voz sobre ip y descarga de contenido desde la web, para ello se utilizaran dos redes geográficamente separadas, entre las cuales se inyectará tráfico haciendo uso de las herramientas IPERF y D-ITG, todo esto con el objetivo de verificar si la implementación propuesta en este trabajo mejora el rendimiento de la red.

Finalmente de los resultados obtenidos se realizará una comparación entre el escenario con la tecnología SND y sin ella, de este modo se verificará el rendimiento, ventajas y desventajas que aporta la implementación propuesta en este trabajo.

4.1. ESCANEO DE PUERTOS EN LA RED

4.1.1. Escenario 1: Red con la implementación SDN

En el sistema propuesto se evaluó la capacidad de bloquear el paso de tráfico a través de determinados puertos TCP o UDP con el propósito de impedir el funcionamiento de aplicaciones que hacen uso de dichos puertos, para esto se tiene la topología mostrada en la Figura 11, en ella se tienen dos computadores portátiles, el primero con sistema operativo *Debian*, será en donde se realizará un escaneo profundo de la red mediante el uso de la herramienta NMAP, en el segundo computador con sistema operativo Windows 7 se ejecutara la herramienta de pruebas de conectividad PAPING.

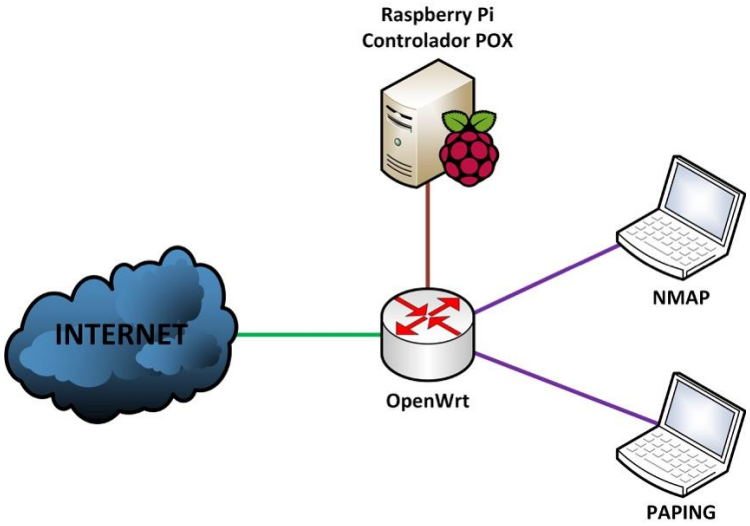


Figura 11. Red con la tecnología SDN.

Los puertos a ser bloqueados se muestran en la Tabla 4, cada uno de ellos corresponde a un servicio proporcionado por la red.

Tabla 4.
Puertos bloqueados.

PUERTOS	SERVICIO
23	Telnet
22	SSH
80	Conexiones http
443	Conexiones HTTPS

Previamente dichos puertos fueron ingresados en la aplicación, en conjunto con el controlador SDN fueron los encargados de instalar las reglas de reenvío de tráfico en los Open vSwitch.

Mediante el uso de la herramienta Nmap en Linux se realizó un escaneo profundo de los puertos, obteniendo como resultado los puertos abiertos, cerrados y filtrados, los cuales se puede apreciar en la Figura 12.

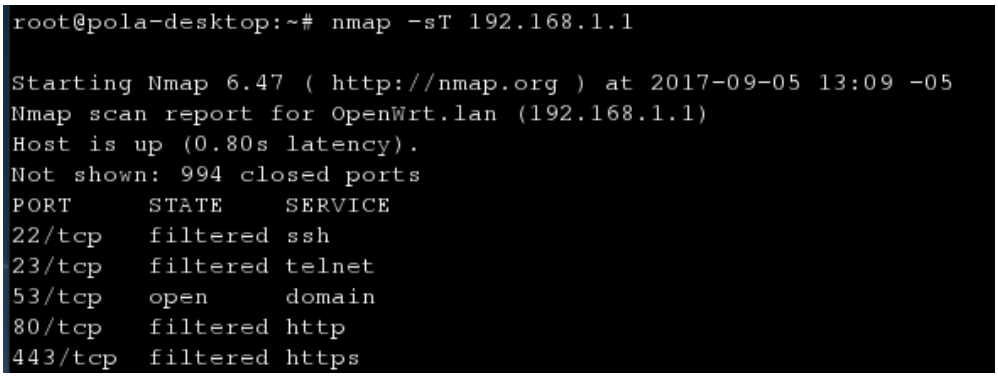


Figura 12. Puertos filtrados y abiertos.

Los puertos abiertos son aquellos que aceptan conexiones y paquetes TCP y UDP, además indican que servicios están disponibles en la red para ser utilizados, los puertos que aparecen como cerrados siguen siendo accesibles, pero no tienen una aplicación escuchando en ellos.

Finalmente los puertos con el estado filtrado son los que se ingresaron en la aplicación para bloquear el paso de tráfico a través de ellos, el filtrado proviene de las reglas instaladas por el controlador SDN en el Open vSwitch, cuando se intenta establecer una conexión a través de ellos, responden con mensajes de error.

Mediante el uso de la herramienta *PAPING* se realizaron pruebas de conectividad a través de los puertos filtrados, se pudo observar en la Figura 13 que se tuvo una pérdida del 100%, demostrando así que el sistema implementado funciona correctamente al momento de impedir el paso de tráfico a través de dichos puertos.

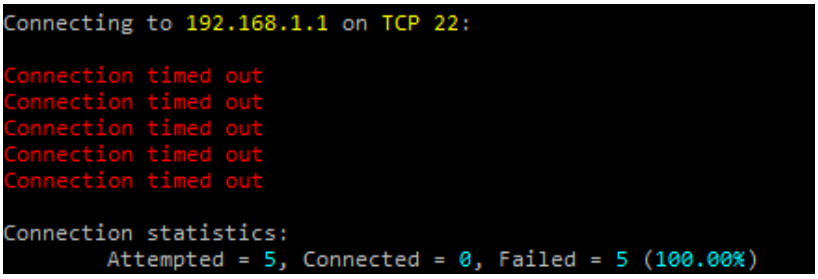


Figura 13. Conectividad fallida a través del puerto 22.

4.1.2. Escenario 2: Red sin la implementación SDN

En la red sin la tecnología SDN se realizó un escaneo de puertos, de los resultados obtenidos se puede verificar que no existen puertos filtrados como se muestra en Figura 14, que impidan el paso de tráfico TCP o UDP, esto se comprobó realizando una prueba de conectividad a través de los puertos que fueron bloqueados en el escenario 1, en la Figura 15 se puede apreciar que los resultados obtenidos muestran un 100% de éxito en la transmisión, para esto se utilizó la herramienta PAPING.

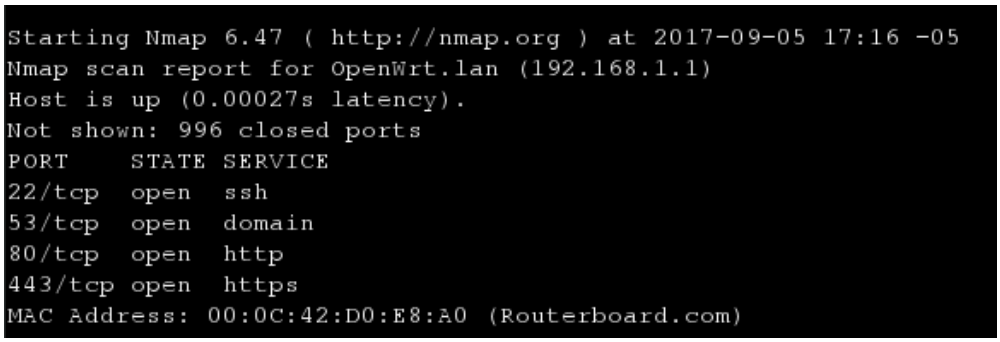


Figura 14. Puertos abiertos.

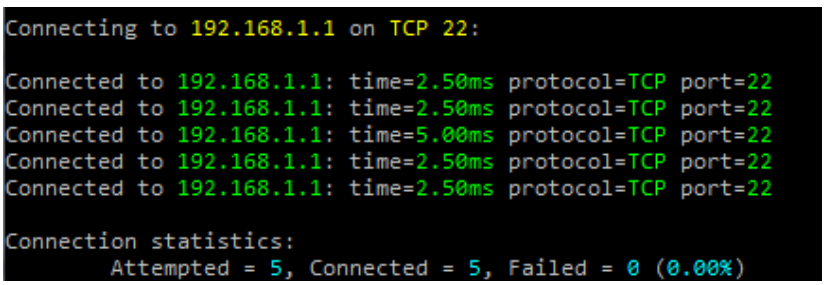


Figura 15. Éxito en la conectividad a través del puerto 22.

El éxito del envió se debe a que en la red sin la implementación propuesta en este trabajo, no existe ningún elemento que impida el paso normal del tráfico a través de los diferentes puertos.

4.2. RENDIMIENTO DE LA RED

Para la realización del análisis se cuenta con dos escenarios como se los muestra en la Figura 16, en los cuales se realizó la misma prueba durante el mismo período de tiempo, el primero cuenta con la tecnología SDN, mientras que el segundo es una red doméstica tradicional, en cada uno de ellos se tienen dos redes separadas geográficamente, la primera denominada Localidad 1, y la segunda denominada Localidad 2.

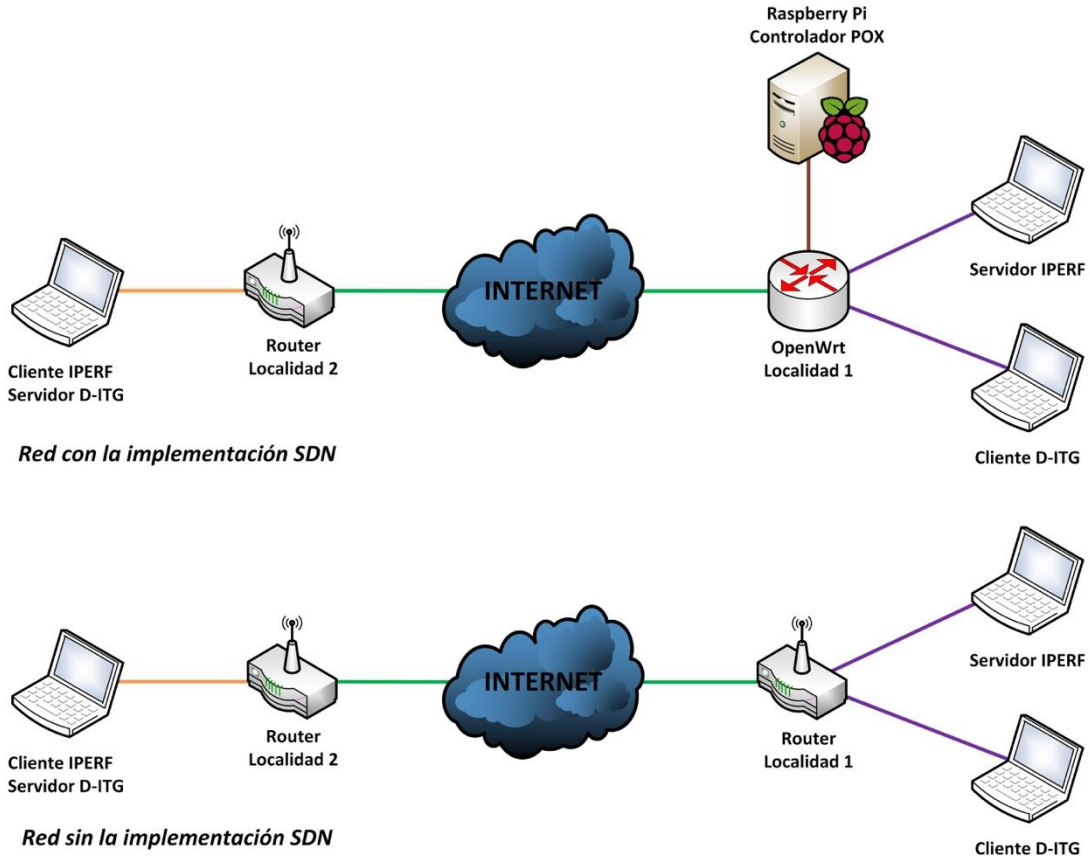


Figura 16. Escenarios con la implementación SDN y sin ella.

En la Localidad 1 correspondiente al escenario uno, es donde se tiene la red con la implementación SDN, en esta se configuran dentro del router dos Open vSwitch

con tres enlaces virtuales entre ellos, a cada uno le corresponde un tipo de tráfico: VoIP, video y web junto al resto del tráfico como se puede observar en la Figura 17, con un ancho de banda de 3 Mbps, 3 Mbps y 2 Mbps, respectivamente. La suma de las porciones de ancho de banda de cada enlace virtual corresponden al ancho de banda total proporcionado por el ISP, en este caso la suma da como resultado 8Mbps.

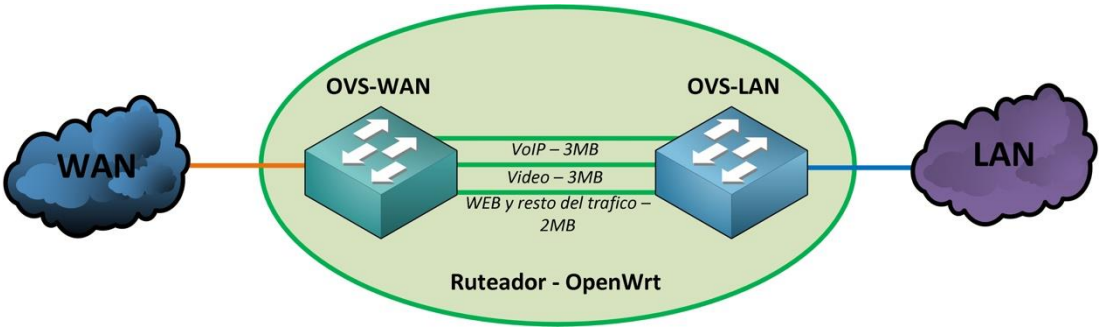


Figura 17. Router configurado con Open vSwitch con enlaces virtuales.

En el escenario 2 el cual corresponde la red sin la implementación SDN, no es necesario realizar ningún tipo de configuración adicional, todo quedará con la configuración por defecto.

En la localidad 2 tanto para el escenario uno como para el escenario dos, se cuenta con un router doméstico en el cual se abrieron todos los puertos necesarios para permitir el paso de tráfico a la PC en la cual fueron ejecutadas las herramientas *D-ITG* e *IPERF*.

Para el presente análisis fue necesario realizar una prueba en ambos escenarios con una duración de 10 minutos (600 segundos), para lo cual se utilizaron tres computadores, dos en la localidad 1 y una en la Localidad 2, en cada uno de ellos se utilizaron las herramientas *IPERF* y *D-ITG* para enviar simultáneamente flujos de datos de los cuales se determinará parámetros como lo son el *jitter*, *throughtput* y latencia, adicionalmente se monitoreará la pérdida de paquetes mediante la utilidad Ping.

En la Localidad 1 se utilizó el primer computador para generar tráfico de VoIP mediante el uso de la herramienta *D-ITG*, simultáneamente durante el mismo periodo

de tiempo se tiene el segundo host que simulará la descarga de un archivo a través de la herramienta *IPERF*.

En la prueba realizada se analiza el *jitter* y latencia en referencia al tráfico de VoIP ya que es propenso a tener variaciones o retrasos en el arribo de paquetes, es por esto que para tener una adecuada comunicación la latencia no debería sobrepasar los 150ms, a lo mencionado se suma el *jitter* que entre más bajo sea mejor será la calidad de la comunicación.

Los valores del *jitter*, *througput*, latencia y pérdida de paquetes se los obtuvo mediante las herramientas: *D-ITG* para el primero, *IPERF* para el segundo y PING para los dos restantes.

4.2.1. Tráfico de voz sobre IP

En la Tabla 5 se muestran los valores de *jitter* obtenidos en ambos escenarios tras enviar tráfico de VoIP entre las dos localidades, los valores expuestos tanto en la tabla como en los gráficos demuestran que la implementación SDN es capaz de reducir los valores de *jitter* y latencia para este tipo de tráfico aun cuando se tiene otra aplicación compitiendo por el ancho de banda.

Tabla 5.
Resultados obtenidos tras realizar la misma prueba en ambos escenarios.

Resultados		
Parámetros	**Con SDN**	**Sin SDN**
Número de flujos	1	1
Tiempo	600 s	600 s
Jitter promedio	1,219951 ms	9,456475 ms
Latencia promedio	69,72064 ms	254,235738 ms
Paquetes perdidos	6.2%	85%

En la Figura 18 se tiene los resultados correspondientes al *jitter*, en ellas se puede ver la ventaja que tiene la implementación propuesta en este trabajo versus la red doméstica tradicional, en la primera los valores del *jitter* se mantienen constantes sin presentar picos que sobrepasen los 4 ms, a diferencia de la segunda en la que ocurre

todo lo contrario, los valores son completamente aleatorios, pasan de estar en un nivel bajo a uno muy alto, en donde se llega a sobrepasar los 20 ms.

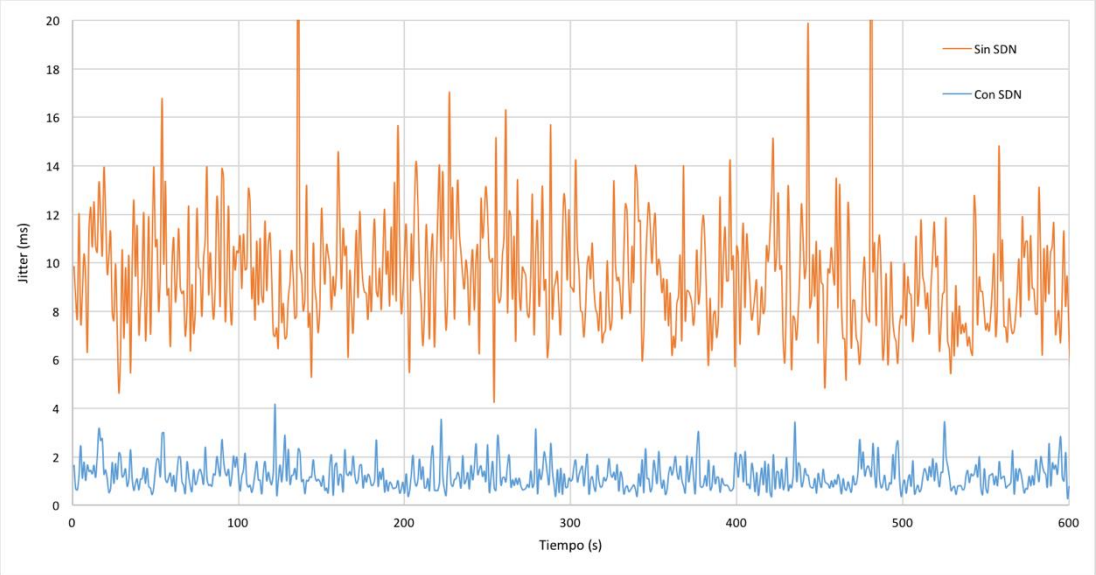


Figura 18. Jitter obtenido con y sin la tecnología SDN y sin ella.

La Figura 19 correspondiente tanto a la red con la implementación SDN y a la red sin dicha implementación, se muestran el tiempo que tardan los paquetes en llegar a su destino, en el primer caso se puede apreciar que los valores de la latencia no sobrepasan los 200 ms, a diferencia de la red en la que no se tiene dicha implementación en donde se tiene tiempos que sobrepasan los 500 ms.

Los valores de latencia recomendados para que la comunicación sea de buena calidad no deberían sobrepasar los 200 ms, lo cual demuestra que la implementación SDN es capaz de mejorar el rendimiento de la red.

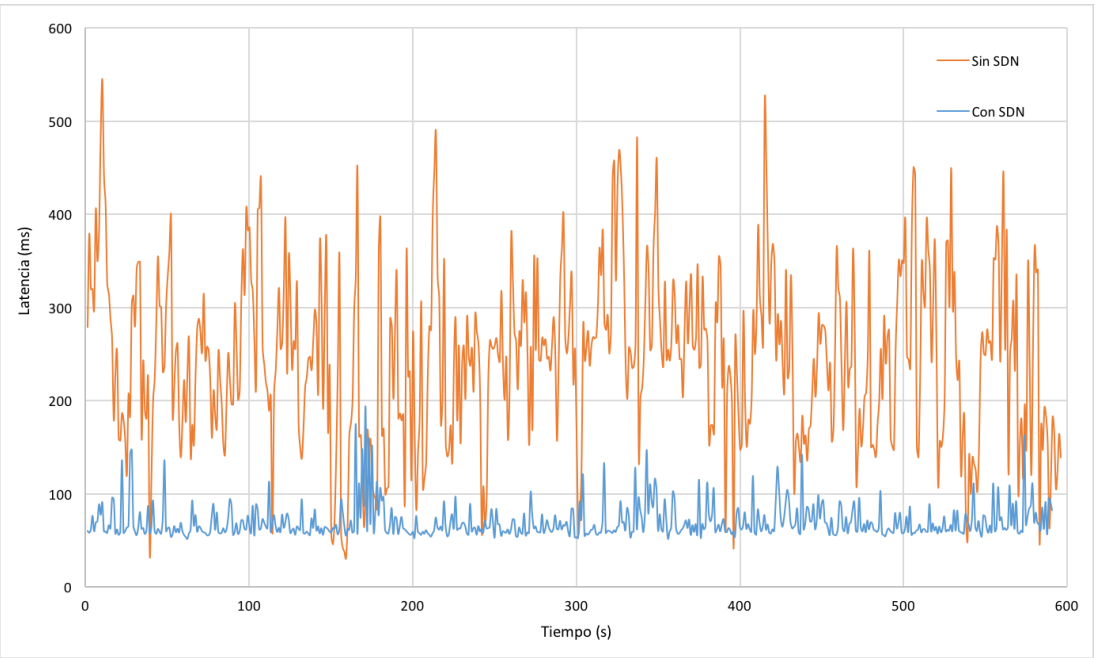


Figura 19. Latencia obtenida con la tecnología SDN y sin ella.

Los resultados obtenidos en la red con la tecnología SDN demuestran que tanto el jitter como la latencia se mantienen estables, aun cuando se tiene una aplicación más compitiendo por el ancho de banda, los parámetros mencionados no aumentan a niveles en los que el uso de aplicaciones de VoIP podrían verse afectados, esto se consigue con la clasificación de tráfico que se realiza en el controlador SDN.

4.2.1. Descarga de contenido

La calidad de servicio fue evaluada en el contexto de navegación web, en donde se determino la velocidad real a la que circula este tipo de tráfico, en la Figura 20 se puede ver el througput de la red con la tecnología SDN y la red sin dicha tecnología. En la Tabla 6 se tiene un resumen de los resultados obtenidos tras finalizar la prueba en ambos escenarios.

Tabla 6.
Resultados obtenidos tras realizar la misma prueba en ambos escenarios.

Resultados		
Parámetros	**Con SDN**	**Sin SDN**
Número de flujos	1	1
Tiempo	600 s	600 s
Througput promedio	1,99261 Mbits	0,97494 Mbits

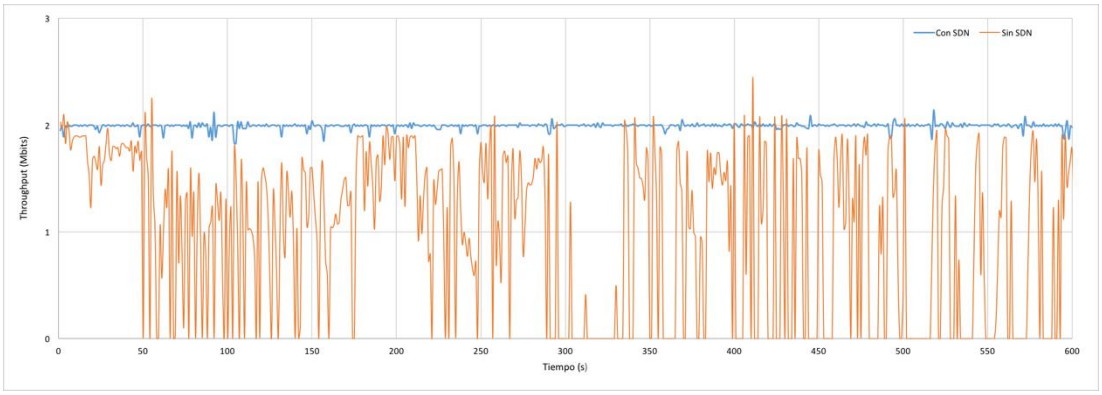


Figura 20. Throughput obtenido con la tecnología SDN y sin ella.

En la Figura 20, al comparar las graficas resultantes se puede observar que bajo la implementación SDN el ancho de banda asignado al tráfico de navegación web se mantiene constante, sin sobrepasar o disminuir significativamente de los 2 Mbps que inicialmente fueron configurados. El tener un tráfico constante como el que se obtiene con al implementación SDN garantiza que el rendimiento del resto el tráfico incluyendo al de VoIP que paralelamente esta circulando en la red no se vea afectado o interrumpido por perdida de paquetes.

La red sin la implementación SDN al no poseer un limitador de velocidad, todo el tráfico de la red es enviado por el único canal existente degradando tanto su rendimiento como el del tráfico de VoIP, es por esta razón que durante los 600 segundos de duración de la prueba se tiene momentos en el que el throughput es de 0 Mbps, esto se traduce en una red lenta y deficiente.

4.3. ANÁLISIS DE RESULTADOS

En la actualidad el control de acceso en los routers domésticos se consigue bloqueando en su totalidad el tráfico proveniente de un determinado dispositivo dentro de la red mediante su dirección MAC, con la implementación propuesta en este trabajo ya no es necesario bloquear completamente al dispositivo o host, lo que se hace es bloquear el o los puertos TCP/UDP que utilizan las aplicaciones para enviar o recibir tráfico de Internet, permitiendo así que otras aplicaciones que también hacen uso del ancho banda puedan ser utilizadas sin ningún problema.

El uso del control de acceso propuesto en este trabajo permite que los cambios que se realizan tengan un impacto sobre toda la red y no solo sobre un host en específico.

Como se pudo observar en los resultados obtenidos tras realizar la misma prueba en ambos escenarios (escenario uno y escenario dos), la implementación propuesta en este trabajo impide el paso de tráfico por determinados puertos, permitiendo así el uso de otras aplicaciones que hagan uso de los recursos de la red, a diferencia de la red en la que no se tiene la implementación mencionada, en donde no se puede bloquear determinados puertos, teniendo como opción el bloqueo total del host.

La diferencia de una red doméstica tradicional con la red propuesta en este trabajo, radica en que en la primera todas las aplicaciones compiten por el ancho de banda de la red, degradando su rendimiento, esto se ve agravado por las limitaciones de hardware y software que tienen los tuteadores domésticos, haciendo que muchas veces no sea posible realizar ningún tipo de configuración referente a QoS.

En la implementación que se propone en este trabajo se logra impedir que las aplicaciones compitan por el ancho de banda disponible en la red, consiguiendo que el rendimiento correspondiente en este caso a tráfico de VoIP y WEB se mantengan estables y sin pérdidas.

A diferencia de una red doméstica tradicional, en la implementación propuesta en este trabajo el tráfico generado por las herramientas *D-ITG* e *IPERF* es enviado al controlador SDN para ser clasificado de acuerdo al tipo de aplicación, posterior a dicha clasificación sigue la instalación de reglas de reenvío en cada Open vSwitch, finalmente dichos *switchs* virtuales envían tráfico por cada uno de los enlaces que corresponden a cada tipo de aplicación, en este caso puntual todo el tráfico correspondiente a VoIP es enviado por el enlace virtual de 3 Mbps.

Como se pudo observar en los resultados obtenidos en la prueba *Rendimiento de la red*, tanto el *Tráfico de voz sobre IP (VoIP)* y *Descarga de contenido*, la red con la implementación SDN tiene una clara ventaja sobre la red doméstica tradicional, los parámetros que se tomaron en consideración para determinar el rendimiento demostraron mantenerse contantes, en el caso del jitter los valores no sobrepasaron

los 4 ms, algo similar ocurrió con la latencia la cual no sobrepaso los 200 ms, a diferencia de la red sin la implementación SDN donde los valores del jitter y latencia alcanzaron valores de 20ms y 500ms respectivamente.

En la red propuesta en este trabajo se tiene una clara ventaja frente a una red tradicional ya que con la implementación SDN se consigue una disminución del 87,10%, 72,57% y 92,7% del *jitter*, latencia y pérdida de paquetes respectivamente además el throughput tiene una mejora del 51,07%, esto se traduce en una red que hace un uso mas eficiente del ancho de banda que posee.

4.3.1. VENTAJAS

En la actualidad muchos ruteadores domésticos cuentan con un control de acceso que les permite bloquear completamente el acceso a los recursos de la red a un determinado host mediante su dirección mac, la ventaja que se presenta en este trabajo es el bloqueo de aplicaciones mediante el cierre de los puertos TCP/UDP a través de los cuales envía tráfico, esto permite implementar políticas de navegación a nivel de toda la red, y no simplemente a determinados hosts (Seddiki, Muhammad, & Donovan, 2014).

Los ruteadores que actualmente entregan los ISP a los hogares no cuentan con una opción de calidad de servicio entre su configuración, lo que hace que para el usuario final sea imposible mejorar el rendimiento de la red, es por esto que el sistema implementado en este trabajo trae funcionalidades muy poco comunes a las redes domésticas, como lo es la configuración del ancho de banda de determinados tipos de aplicaciones desde una interfaz web.

Una de las principales ventajas con las que cuenta la tecnología SDN es la facilidad y rapidez con la cual se pueden aplicar cambios dinámicos en el tiempo de las reglas de tráfico o reglas de control de acceso, esto quiere decir que el usuario final puede ordenar mediante la aplicación SDN el bloqueo o priorización de un determinado tipo de tráfico a una determinada hora del día.

Las ventajas que brinda la programabilidad de la tecnología SDN a las redes hace que cualquier actualización a las políticas o reglas de control de tráfico que se tengan en la red puedan ser ejecutadas sobre toda la infraestructura de red de forma rápida y sin comprometer su funcionalidad.

4.3.2. DESVENTAJAS

El desarrollo de aplicaciones con la tecnología SDN en gran medida va a estar limitado por el controlador SDN que se utilice, ya que cada uno cuenta con librerías o módulos propietarios que hacen que las aplicaciones desarrolladas junto a dichos controladores sean incompatibles con el resto, esto limita la portabilidad, soporte y escalabilidad de las aplicaciones desarrolladas.

La tecnología SDN propone un modelo de red centralizado en la cual un único dispositivo es el encargado de controlar el tráfico que circula a través de todas y cada una de las interfaces de los diferentes equipos de red, esto genera un gran problema ya que si este dispositivo llegase a fallar toda la red seria vulnerable a sufrir perdidas de paquetes, desconexiones o inoperatividad total, dejando así a todos los usuarios sin acceso a los diferentes servicios que ofrece la red.

La seguridad dentro de una red SDN es un punto muy importante, ya que el controlador SDN al ser parte central de la red es propenso en mayor medida a ataques informáticos pues es el dispositivo encargado de recibir y reenviar todo el tráfico que circula dentro de la red. Un atacante al conocer este tipo de arquitectura dedicaría todo su esfuerzo en vulnerar la seguridad del controlador SDN mas que de un host en particular.

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1.CONCLUSIONES

La implementación propuesta en este trabajo permite agregar funcionalidades que antes no se tenían dentro de una red doméstica, como lo son el control de acceso mediante el bloqueo de puertos y la calidad de servicio, estas funcionalidades y todas las que se puedan agregar a futuro pueden ser probadas y monitoreadas de forma inmediata ya que la tecnología SDN centraliza la red en un solo dispositivo evitando así configurar el resto de equipos que se podrían tener.

Phython al ser un lenguaje de programación orientado a objetos, multiplataforma y con requerimientos mínimos de hardware permitió desarrollar e implementar tanto el control de acceso como la calidad de servicio directamente sobre una Raspberry Pi, siendo esta ultima un dispositivo con especificaciones de hardware básicas.

OpenWRT al ser una distribución de *Linux Open Source* puede ser instalado en una gran cantidad de dispositivos que cuentan con requerimientos mínimos de hardware, además de permitir un detallado control y configuración a bajo nivel del dispositivo donde se lo instaló. Pudiendo habilitar el soporte a diferentes funcionalidades como en el caso de este trabajo, que se agrego el modulo Open vSwitch para la creación de los *switchs* virtuales.

La implementación de la tecnología SDN demostró que el rendimiento de la red es superior en comparación a una red tradicional, aun cuando se tienen varias aplicaciones generando tráfico simultáneamente, los valores que se obtuvieron tanto del jitter como de la latencia se mantuvieron bajo los 20 ms y 200 ms respectivamente, lo cual permite una comunicación óptima y fluida.

La tecnología SDN permite diseñar e implementar aplicaciones de red que se adaptan a las necesidades del administrador, brindándole un control centralizado de toda la red, gracias a esta característica en el presente trabajo se tuvo completo

control sobre el tráfico que circula en la red, pudiendo así implementar el control de acceso y calidad de servicio dentro de una red doméstica.

La programabilidad en las redes definidas por software da la flexibilidad de automatizar la red mediante la ejecución de aplicaciones o scripts que permiten implementar reglas de tráfico capaces de habilitarse o deshabilitarse automáticamente a una hora determinada del día o durante periodos de tiempo específicos.

5.2.RECOMENDACIONES

Antes de realizar cualquier tipo de implementación que involucre redes definidas por software se recomienda realizar un estudio de sus componentes, funcionamiento y protocolos así como equipos que soportan la instalación del software necesarios para su funcionamiento.

Es recomendable escoger el controlador SDN que mejor se adapte al tipo de implementación que se desea realizar así como de la capacidad computacional de los equipos que se tiene a disposición, ya que existen controladores necesitan más recursos que otros.

Se recomienda utilizar ruteadores que permitan la instalación de cualquiera de las versiones de OpenWRT, pues al ser un sistema operativo de código abierto permite agregar gran cantidad de aplicativos que podrían servir para realizar diferentes tipos de pruebas directamente desde el ruteador, adicionalmente tanto su configuración como operación se facilitan pues es un sistema operativo basado en Linux.

Se propone utilizar el presente trabajo como base para futuras investigaciones donde el controlador SDN pueda funcionar directamente en la infraestructura del ISP, sin necesidad de tener un equipo adicional en la red doméstica con fue el caso de la presente investigación.

BIBLIOGRAFÍA

Open Networking Foundation. (2012). *Software-Defined Networking (SDN) Definition*. Recuperado el 12 de Enero de 2017, de Open Networking Foundation: https://www.opennetworking.org/sdn-resources/sdn-definition

OpenWRT. (12 de Agosto de 2017). *Open WRT Wireless Freedom*. Recuperado el 20 de Septiembre de 2017, de Open WRT Wireless Freedom: https://wiki.openwrt.org/doc/start

Feamster, N. (2010). Outsourcing Home Network Security. *Proceedings of the 2010 ACM SIGCOMM workshop on Home networks (HomeNets '10)*, (págs. 37-42). New York.

Bianco, A., Giaccone, P., Mahmood, A., Ullio, M., & Vercellone, V. (2015). Evaluating the SDN control traffic in large ISP networks. *En IEEE International Conference on Communications (ICC)*, (págs. 5248-5253). Londres.

Pfaff, B., Pettit, J., Koponen, T., & Jackson, E. (2015). The Design and Implementation of Open vSwitch. *En 12th USENIX Conference on Networked Systems Design and Implementation (NSDI'15)*, (págs. 117-130). California.

Gharakheili, H., Bass, J., Exton, L., & Sivaraman, V. (2014). Personalizing the Home Network Experience using Cloud-Based SDN. *En IEEE International Symposium on a World of Wireless* (págs. 1-6). Sydney, NSW, Australia: IEEE.

Pinilla, R. A. (2015). *Estudio de las redes definidas por software mediante el desarrollo de escenarios virtuales basados en el controlador OpenDayLight*. Tesis, Universidad Politécnica de Madrid, Escuela Técnica Superior de Ingenieros de Telecomunicación, Madrid, España.

Bozkurt, I. N., Zhou, Y., & Benson, T. (2015). Dynamic Prioritization of Traffic in Home Networks. *CoNEXT Student Workshop'15*, (págs. 1-3). Heidelberg, Germany.

Kim, H., & Feamster, N. (2013). Improving Network Management with Software Defined Networking. *IEEE Communications Magazine , 51* (2), 114-119.

Seddiki, S., Muhammad, S., & Donovan, S. (2014). FlowQoS: Per-Flow Quality of Service for Broadband Access Networks. *Third workshop on Hot topics in software defined networking (HotSDN '14)* (págs. 1-8). Chicago, Illinois, USA: SIGCOMM.

McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., & Rexford. (2008). OpenFlow: Enabling Innovation in Campus Networks. *ACM SIGCOMM Computer Communication Review , 38* (2), 69-74.

Hong, D. K., Ma, Y., Banerjee, S., & Mao, M. (2016). Incremental Deployment of SDN in Hybrid Enterprise and ISP Networks. *Symposium on SDN Research SOSR '16*, *1*, pág. 7. Santa Clara, CA, USA.

Alcock, S., & Nelson, R. (2012). *Libprotoident: Traffic Classification Using Lightweight Packet Inspection*. Informe técnico, University of Waikato, WAND Network Research Group, Waikato, Nueva Zelanda.

Kim, W., Sharma, P., Lee, J., & Banerjee, S. (2009). Automated and Scalable QoS Control for Network Convergence. *Network management conference on Research on enterprise networking (INM/WREN'10)*, (págs. 1-1). Berkeley, CA, USA.

Ma, Y.-W., Chen, J.-L., Chang, C.-C., Chiang, C.-M., & Xie, Y.-L. (2015). SDN Test Cases Development and Implementation. *17th International Conference on Advanced Communication Technology (ICACT)*, (págs. 618-621). Seoul, South Korea.

Dandekar, Kharade, R., & Tathare, S. (2015). Implementing a low cost SDN ecosystem in LAN. *Annual IEEE India Conference (INDICON)* (págs. 1-5). New Delhi, India: IEEE.

Siebertz, F. (2014). *Software Defined Networking*. Tesis, Hochschule Bonn-Rhein-Sieg University of Applied Sciences, Department of Computer Science, Sankt Augustin, Alemania.

Centeno, A. G., Vergel, C. M., & Calderón, C. A. (2014). Controladores SDN, elementos para su selección y evaluación. *Revista Telem@tica , 13* (3), 10-20.

OpenFlow Switch Specification. (25 de Junio de 2012). *Open Networking Foundation*. Recuperado el 1 de Junio de 2017, de Open Networking Foundation: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.0.noipr.pdf

Brown, M. A. (28 de 10 de 2006). *Traffic Control HOWTO*. Recuperado el 22 de Julio de 2017, de Traffic Control HOWTO: http://tldp.org/HOWTO/Traffic-Control-HOWTO/index.html

Boucadair, J. (Marzo de 2014). *Software-Defined Networking: A Perspective from within a Service Provider Environment*. Recuperado el 15 de Junio de 2017, de RFC 7149: https://www.ietf.org/mail-archive/web/ietf-announce/current/msg12553.html

Robertson, P. (2017). *Software-Defined Networking Changes the Paradigm for Mission-Critical Operational Technology Networks*. Recuperado el 3 de Mayo de 2017, de Schweitzer Engineering Laboratories: https://cdn.selinc.com/assets/Literature/Publications/White%20Papers/0016_SDNChanges_PR_20170214.pdf?v=20170619-053020

Wallner, R. C. (2013). An SDN Approach: Quality of Service using Big Switch's Floodlight Open-source Controller. *Asia-Pacific Advanced Network , 35*, 14-19.