Depósito legal ppi 201502ZU4662
Esta publicación científica en formato
digital es continuidad de la revista impresa
Depósito Legal: pp 197402ZU789
● ISSN: 1315-9518 ● ISSN-E: 2477-9431

RCS
Revista de Ciencias Sociales

Universidad del Zulia. Revista de la Facultad de Ciencias Económicas y Sociales
Vol. XXVII, **No. Especial 3, 2021**

Revista de Ciencias Sociales

Esta publicación científica en formato
digital es continuidad de la revista impresa
Depósito Legal: pp 197402ZU789
ISSN: 1315-9518

Revista de Ciencias Sociales (RCS)
Vol. XXVII, Número Especial 3, 2021. 111-127 pp.
FCES - LUZ ● ISSN: 1315-9518 ● ISSN-E: 2477-9431

Como citar APA: Oviedo-Bayas, B., Zhuma-Mera, E., Bowen-Calero, G., y Patiño-Maisanche, Bryan (2021). Voz IP seguras implementadas en Redes Definidas por Software. *Revista de Ciencias Sociales (Ve)*, *XXVII*(Número Especial 3), 111-127.

Voz IP seguras implementadas en Redes Definidas por Software

Oviedo-Bayas, Byron[*]
Zhuma-Mera, Emilio[**]
Bowen-Calero, Génesis[***]
Patiño-Maisanche, Bryan[****]

Resumen

Hablar sobre las redes definidas por software, es un paradigma, son más agiles, escalables y dinámicas, para un mejor aumento de programación y automatización. Una de sus ventajas es la reducción de costos operativos, mejora en el rendimiento de la red, con una arquitectura abierta permitiendo a diferentes proveedores migrar a estas redes. El presente artículo está enfocado a la implementación de una red definida por software que permita brindar servicio de Voz IP seguros, analizar e identificar cuáles son los controladores y protocolos utilizados en las redes y seleccionar el más acorde al estudio. A través de un estudio descriptivo-documental, se realiza el diseño e implementación de una red, en un ambiente físico, utilizando Routerboards y teléfonos IP, permitiendo la interconectividad de los mismos. Se realizan métricas como: Latencia, rendimiento, ancho de banda, y se evalúa la seguridad, que presenta la red mediante la tecnología Voz IP. Como resultado, la prueba realizada en un ambiente controlado, determinó que la red tenía menos pérdidas y retardo en cuestión del intervalo de tiempo de los paquetes. Se concluye, que una buena elección del controlador OpenDayLight y el protocolo OpenFlow, será de gran ayuda para una estable conexión con los equipos.

Palabras clave: Redes definidas por software; flujo abierto; controladores; VoIP; protocolo IP.

[*] PhD. en Tecnologías de la Información y Comunicación. Profesor Titular y Director de Investigación de la Universidad Técnica Estatal de Quevedo, Ecuador. E-mail: boviedo41@uteq.edu.ec ORCID: https://orcid.org/0000-0002-5366-5917

[**] Máster en Conectividad y Redes de Computado (MSc.). Profesor Titular y Coordinador de Carrera en la Universidad Técnica Estatal de Quevedo, Ecuador. E-mail: ezhuma@uteq.edu.ec ORCID: https://orcid.org/0000-0002-3086-1413

[***] Ingeniera en Telemática. Universidad Técnica Estatal de Quevedo, Ecuador. E-mail: genesis.bowen2015@uteq.edu.ec ORCID: https://orcid.org/0000-0002-3702-0203

[****] Ingeniero en Telemática. Universidad Técnica Estatal de Quevedo, Ecuador. Co-Fundador de DOMOBELL-SYSTEMS S.A.S., Ecuador. E-mail: bryan.patino2015@uteq.edu.ec ORCID: https://orcid.org/0000-0003-2723-0204

Recibido: 2021-02-20 · **Aceptado:** 2021-05-10

Secure Voice IP implemented in Software Defined Networks

Abstract

Talking about software-defined networks is a paradigm, they are more agile, scalable and dynamic, for a better increase in programming and automation. One of its advantages is the reduction of operating costs, improvement in network performance, with an open architecture allowing different providers to migrate to these networks. This article is focused on the implementation of a software-defined network that allows to provide secure Voice IP service, analyze and identify which are the controllers and protocols used in the networks and select the one that best suits the study. Through a descriptive-documentary study, the design and implementation of a network is carried out, in a physical environment, using Routerboards and IP phones, allowing their interconnectivity. Metrics such as: Latency, performance, bandwidth are performed, and the security presented by the network is evaluated through Voice IP technology. As a result, the test conducted in a controlled environment, determined that the network had less loss and delay in terms of the time interval of the packets. It is concluded that a good choice of the OpenDayLight controller and the OpenFlow protocol will be of great help for a stable connection with the equipment.

Keywords: Software defined networks; open flow; controllers; VoIP; IP protocol.

Introducción

El área tecnológica ha tenido un gran impacto de desarrollo en los últimos años, puesto que la globalización ha influido en que las personas y las empresas "se vean abocadas a adoptar tecnologías de información para soportar sus actividades diarias de negocio, adaptarse al entorno y basarse en ellas para ser más competitivas" (Briñez, 2021, p.181). Es por ello que, debido a las necesidades de los clientes o usuarios, las grandes empresas tecnológicas se han visto obligadas a implementar equipos acordes a las nuevas tecnologías, tal es el caso de las redes de quinta generación (Orgaz, 2019).

A nivel mundial la comunicación demanda una mejora en la calidad en servicios, debido que estas infraestructuras están quedando obsoletas. Con la llegada de las nuevas tecnologías, estas causan un gran cambio en cuanto a la administración y configuración de los equipos, y muchas veces el medio de comunicación, tales como: Inteligencia artificial, aplicaciones inteligentes, *internet* de las cosas (IoT), telefonía IP, entre otros. Al respecto, Arbeláez-Campillo, Villasmil y Rojas-Bahamón (2021), sostienen que la inteligencia artificial "pudiera en muchos aspectos superar las limitaciones y contradicciones de la inteligencia humana, profundizando su condición de ser una fuerza complementaria de la misma o, por el contrario, terminar resultando en un factor antagonista" (p.504).

En ese sentido, una comparación entre las Redes Definidas por *Software* (RDS) y aquellas con arquitectura (TCP/IP), realizada por Oviedo, et al. (2020), utilizando el protocolo *OpenFlow* en dos escenarios distintos, pero en cada uno de ellos con la misma topología y características, evidenció, por un lado, latencias máximas favorables para RDS en el primer escenario, y, por otro lado, encontraron errores de transmisión para el segundo escenario. Todo lo realizado fue bajo el Sistema Operativo *Ubuntu Server* 18.04.

Por otra parte, de acuerdo con Aguirre y Ortega (2005), la telefonía surgió de la necesidad de tener un medio para que las personas se comuniquen. Hace varias décadas atrás, solamente existía la telefonía fija,

pero con el pasar de los años y el avance tecnológico surgió el desarrollo de las redes de telecomunicaciones y el *Internet*. Todos los servicios analógicos han migrado a ser digitales; es decir la voz se comprime y luego se transmite por medio de la misma red de datos utilizando el protocolo IP; este servicio se lo conoce como telefonía IP (Moyolema, 2015). Con ello, surgieron grandes beneficios para las empresas, puesto que disminuyeron costos en la implementación e instalación de los equipos y llamadas telefónicas a diferentes destinos, solo deben de disponer de una red de datos para servicios de VozIP o Voz sobre protocolo de *internet* (Caldera y Suazo, 2011).

Actualmente, al utilizar más de un servicio, ésta red requiere de un ancho de banda mayor. Así, las Redes Definidas por *Software* (RDS), es un concepto nuevo de arquitectura de red, es escalable, tiene una mejor gestión y toda su inteligencia se encuentra en un controlador central (Parra, Morales y Hernández, 2015). Al respecto, Flores (2018), en su investigación se propuso diseñar una arquitectura para la gestión de redes residenciales, mediante el uso de funciones de virtualización y RDS, en la cual se utilizaron aplicaciones para uso de usuarios, que gestionan servicios residenciales tales como el consumo de *internet* y servicio telefónico. Incluye funcionalidades de administración de tráfico, con el fin de que servicios en línea como videos en alta definición y juegos, tengan un rendimiento óptimo.

En Ecuador, existen varios estudios referentes a redes definidas por *software* y su implicación con servicios de VoIP. Moscoso (2016), indica en su estudio que se desarrolla una aplicación para tener calidad de servicio, priorizando tráfico en una red definida por *software*. La aplicación se desarrolló bajo lenguaje *Java* y se ejecutó junto con el controlador *Floodlight*, las pruebas consistieron en realizar llamadas telefónicas y enviar paquetes que saturen la red, permitiendo que la aplicación implemente calidad de servicio y no se afecte la llamada.

Asimismo, se cuenta con un estudio denominado "Despliegue de una red SDN aplicando el protocolo MPLS y generando políticas de QoS (Calidad de Servicio) para servicios de telefonía IP", realizado por Fernandez y Ulloa (2016), en el cual, los autores plantean un laboratorio de pruebas en una red SDN (RDS, siglas en español) con el controlador *OpenDayLight*, gestionando y administrando todos los aspectos concernientes a equipos que envían paquetes y también se implementa el protocolo de transporte MPLS (Conmutación de etiquetas multiprotocolo). Además, el servicio VoIP es implementado sobre una nube con tecnología de virtualización *OpenStack*, en el cual se aplican pruebas para verificar la calidad de servicio**.**

De igual manera, el proyecto realizado en Berlin, "Etiquetado dinámico de VLAN basado en MAC con *OpenFlow* para redes de acceso WLAN", se compone de dos aplicaciones, la herramienta de ayuda y la función de red (NF). Para realizar el etiquetado dinámico de VLAN basado en MAC, se implementó un NF basado en *Java* utilizando la API interna de *Java* de Floodlight. Este NF, hace uso del protocolo *OpenFlow* utilizando *floodlight* para cubrir el proceso de etiquetado y des-etiquetado de la VLAN directamente detrás del WAP (Koerner y Kao, 2016).

OpenFlow, es actualmente el concepto RDS más implementado, que proporciona comunicación entre el controlador y los conmutadores. Sin embargo, el dinamismo de las redes programables también trae nuevos desafíos de seguridad potenciales relacionados con varios ataques, como escaneo, ataques de suplantación de identidad, denegación de servicio (DoS), y así sucesivamente (Li, Meng y Kwok, 2016).

Por otra parte, un grupo de investigadores realizaron una simulación integral de la carga de trabajo real de la empresa MIXvoip y demostraron que las estrategias propuestas superan a las que se utilizan actualmente. Analizaron las horas de facturación para el aprovisionamiento de VM, el número de llamadas puestas en espera y los criterios de calidad de voz. Demostraron que, las estrategias propuestas con predicción dinámica de RT y

RoC disminuyeron ligeramente la calidad de servicio, reduciendo la hora de facturación hasta un 30%, y aumentando las llamadas en espera de unas 2 a 9 al día (Tchernykh, et al., 2019).

En cuanto al experimento, "Un sistema híbrido de detección de ataques DoS basado en entropía para redes definidas por software (SDN): Un mecanismo de confianza propuesto", desarrollado por AbdelAzim, et al. (2021), el mismo demostró que es robusto, puesto que tiene en cuenta la naturaleza del tráfico de red existente y puede adaptarse en tiempo real a medida que cambia la naturaleza del tráfico en una red. Además, se propuso un marco de confianza para SDN que se puede utilizar para prohibir los nodos que se comportan de forma maliciosa después de darles la oportunidad de rehabilitación.

También, Adil, et al. (2018) en el "Reconocimiento automático de voz para VoIP con ocultación de pérdida de paquetes", adapto una técnica para ocultar la pérdida de paquetes de la Recomendación UIT-G.711, Apéndice I al códec G729 dedicado a la VoIP, cuyo objetivo principal fue la mejora de la ASR en las redes de VoIP y hacer que los sistemas de reconocimiento fueran más robustos ante las pérdidas de paquetes.

Dado todo lo anterior, este artículo tiene como objetivo, la implementación de redes definidas por *software* en un ambiente real, para brindar servicios de Voz IP seguros. Para ello se creó y configuró VLANs mediante el uso de los equipos físicos que contenga el protocolo *OpenFlow*, teniendo como finalidad garantizar la seguridad en la red, confiabilidad en el envío de paquetes, y además evaluar la calidad de servicio en la transmisión de VoIP.

En ese sentido, a través de un estudio descriptivo-documental, se realiza el diseño e implementación de una red, en un ambiente físico, utilizando *Routerboards* y teléfonos IP, permitiendo la interconectividad de los mismos, en los Cuadros 1 y 2, se muestran los recursos de Hardware y Software utilizados. Se realizan métricas como: Latencia, rendimiento, ancho de banda, y se evalúa la seguridad, que presenta la red mediante la tecnología Voz IP. Y, por último, se discuten los resultados obtenidos y se exponen las conclusiones y recomendaciones acerca de los objetivos propuestos.

Cuadro 1
Recursos de *Hardware*

Cantidad	Hardware	Descripción
2	Computadora Portátil	Dell Inspiron ‘15,6 2,90 GHz Intel Core i7-7500U 8 GB RAM 4 GB Tarjeta de video Radeon HP Intel Core i3-4010U 4 GB RAM
1	SSD	Samsung 860PRO 1TB
2	RouterBoard	Microtik RB11UiAS-RM Microtik 951Ui

Fuente: Elaboración propia, 2021.

Cuadro 2
Recursos del *Software*

Software	Descripción
Sistemas Operativos	Windows 10 Ubuntu 18.04
Programas de Oficina	Paquete de Office de Microsoft
Software de trabajo	OpenDaylight VirtualBox Elastix 3CX Wimbox
Otros	Gantt Project Wireshark Advanced IP Scanner Prezzi

Fuente: Elaboración propia, 2021.

1. Etapas para el diseño e implementación de una red

1.1. Etapa 1: Identificación de controladores y protocolos que pueden ser implementados en una RDS

A continuación, en esta etapa se detallará cuáles son los controladores y protocolos y qué función cumplen.

a. Identificación de protocolos RDS

En el Cuadro 3, se indican los protocolos, algunos de ellos ya existentes, con cada uno de sus fabricantes o desarrolladores, y qué funciones realizan cada uno, con el objetivo de determinar cuál es el que mejor se adapta a las necesidades para la implementación de la red RDS. En el mercado, ya se han desarrollado equipos basados en *OpenFlow*, que representa el protocolo predominante para éstas redes, aunque se están diseñando arquitecturas utilizando otros métodos de comunicación, tales como los protocolos: BGP (Protocolo de puerta de enlace de frontera), MPLS-TP (Conutación de etiquetas multiprotocolo: El perfil de transporte), NETCONF (Protocolo de configuración de red), XMPP (Protocolo extensible de mensajería y comunicación de presencia), entre otros.

Cuadro 3
Protocolos de las redes RDS

Protocolo	Desarrollado por	Función
Border Gateway Protocol (BGP)	Estandarizado por la RFC	Permite el intercambio de información entre *hosts* de *Gateway* en la red.
Perfil de Transporte - Multiprotocol Label Switching (MPLS-TP)	Internet Engineering Task Force (IETF)	Diseñado como una tecnología de capa de red en redes de transporte.
NETCONF	Internet Engineering Task Force (IETF)	Resuelve problemas que existen con los protocolos SNMP y la CLI.
Protocolo de Presencia y Mensajería Extensible (XMPP)	Jeremie Miller	Mensajería instantánea, distribución de la información y detección en línea.
OpenFlow	Universidad de Stanford y California	Permite la accesibilidad directa y manipulación entre los dos planos.

Fuente: Elaboración propia, 2021.

De acuerdo a las necesidades de la red RDS, se seleccionó el protocolo *OpenFlow*, puesto que presenta muchas ventajas por lo que sobresale de otros protocolos, esto es gracias a las organizaciones principales como la ONF (OpenNetworking, 2020), que ofrece mejoras en cuanto a sus actualizaciones para un buen desempeño en este tipo de red. Asimismo, permite la accesibilidad directa y manipulación de los planos de datos de los dispositivos y el plano de control, además ofrece un mayor ancho de banda, admite una configuración automatizada de la red, lo cual conlleva a reducir gastos de operación y menor inactividad ante fallos con la red.

b. Identificación de controladores RDS

El controlador, es el núcleo central de la arquitectura RDS (Escobar, 2015). Es por ello, que se han tomado en cuenta seis controladores, de los cuales se analizará cada una de las características, con la finalidad de identificar que controlador es el más acorde al diseño de la red RDS (ver Cuadro 4).

Cuadro 4
Controladores para la RDS

Característica	Controladores					
	NOX	**POX**	**BEACON**	**ONOS**	**Ryu**	**OpenDayLigth**
Soporte	OpenFlow v1.0	OpenFlow v1.0	OpenFlow v1.0	OpenFlow v1.0/v1.3	OpenFlow v1.0/v1.3	OpenFlow v1.0/v1.4
Lenguaje	C++	Python	Java	Java	Python	Java
Plataformas	Linux	Linux, Mac Os, Windows	Linux, Mac Os, Windows	Linux	Linux	Linux, Mac Os, Windows
Código abierto	Si	Si	Si	Si	Si	Si
Interfaz gráfica	Python + QT4	Python + QT4	Web	Web	Web	Web
API	No	No	No	Si	Si	Si
Documentación	Media	Baja	Buena	Media	Media	Media

Fuente: Elaboración propia, 2021.

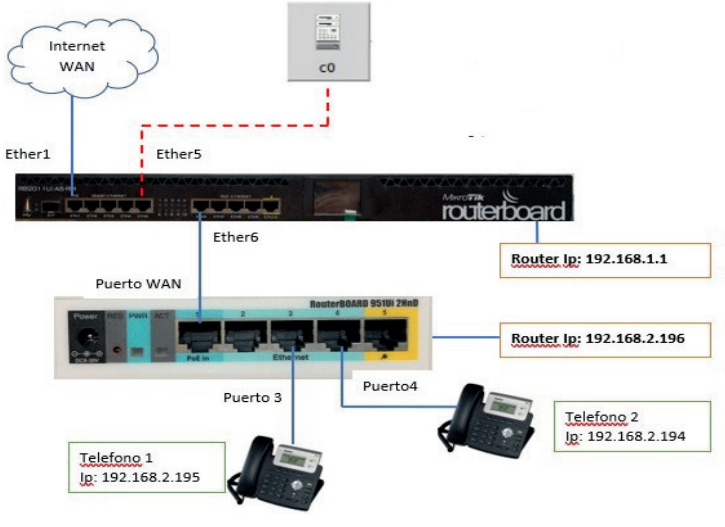
Una vez analizada cada una de las características, se seleccionó el controlador que se adapta mejor al objeto de estudio de la investigación, como lo es el *OpenDayLight*. Este, de acuerdo con Moreno y Zambrano (2018), ofrece una plataforma compatible con

RDS. Además, cuenta con una plataforma de abstracción de servicios, lo cual resulta ventajoso para los desarrolladores, puesto que pueden crear aplicaciones con cualquier protocolo sin ningún inconveniente, de la misma manera en diferentes equipos.

1.2. Etapa 2: Implementación de la red RDS en un escenario físico

Consiste en dos *router MikroTik* RB2011 UiAS-RM y 951Ui, dos teléfonos IP *YEALINK* T20P y una *laptop* (donde estará el controlador). En este escenario, se analiza la red para medir el *jitter*, ancho de banda y latencia para verificar la viabilidad de la red. Para configurar los *routers* se usó el *software Winbox*. Se crearon dos redes, el RB2011 UiAS-RM tendrá la IP 192.168.1.1, para la administración de la VLan; y configuración del puerto 6, para la intercomunicación de los *switch*. El 951Ui, se le asignó la IP 192.168.2.196 para administración de las IP de los teléfonos colocándolo en DHCP. Se colocaron diferentes rangos de direcciones, con la finalidad de evitar conflictos de IP y colapso de la red.

Se descargan los paquetes *Openflow* en la página https://mikrotik.com/download y se los instala en ambos *routers*, arrastrándolo hacia su interfaz, cabe recalcar que, se tiene que verificar primero la versión del *router*, para poder descargar los paquetes correspondientes. Dentro de la opción *Openflow*, se agrega al *router* como *switch*, esto se realiza para ambos *routers*, con el fin de poder implementar la VLan y visualizarlos en el *Opendaylight*. Para interconectar al controlador con el *router* se usó el puerto 5 del MikroTik2011UiAS. En el puerto 6 del mismo, se crea la VLan 200 llamada "Telefonía", que irá conectado al *router* 951Ui al puerto 1, para la intercomunicación entre los "*swtiches*". En el *router* 951Ui se habilitaron los puertos 3 y 4 para los teléfonos y para la VLan el puerto 1. En la Figura I, se divisará la red con sus respectivas conexiones.



Fuente: Elaboración propia, 2021.

Figura I: **Diseño de la red**

En el Cuadro 5, se pueden visualizar las direcciones IP asignadas a los equipos (*routers*, teléfonos, *laptop*), máquinas virtuales y la central telefónica. A continuación, se muestran las direcciones aplicadas en la Figura I. Los teléfonos recibían direcciones DHCP.

Cuadro 5
Asignación de IP

	Dirección IP	3CX, Central telefónica	Observaciones
Router MikroTik2011UiAS	192.168.1.1		Administración de la red para la VLan y asignación del puerto.
Router MikroTik 951Ui	192.168.2.196		
Elastix	192.168.2.249	192.168.2.249:5001	El puerto 5001, es por donde se obtendrá el acceso al navegador. Está en una *VMWare*.
Controlador OpenDayLight (VMWare)	192.168.2.253		Debe estar configurado el *router* como *Switch OpenFlow*. Para visualizarlo por el navegador, se coloca de la siguiente manera, 192.168.2.253:8181/index.html
PC	192.168.2.200		En la cual se está ejecutando las máquinas virtuales y utilizado para ingresar a las configuraciones de los dispositivos.
Teléfono 1	192.168.2.195		
Teléfono 2	192.168.2.194		
VLAN 100			Está editada con el nombre de Telefonía dentro del *router MikroTik* 2011UiAs

Fuente: Elaboración propia, 2021.

De igual manera, en el Cuadro 6 están numerados los puertos a los cuales se hicieron las conexiones de la red, como se muestran en la Figura I.

Cuadro 6
Conexiones de los puertos en los equipos

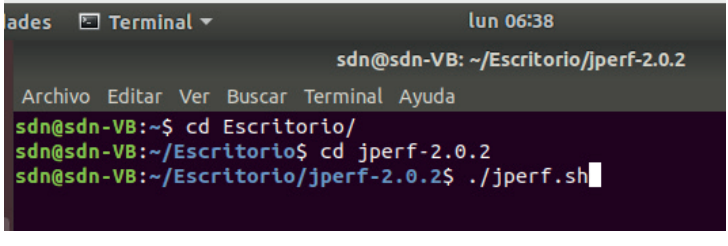
Conexiones / Puertos	MikroTik 2011UiAS	Mikrotik 951Ui	Observaciones
Intercomunicación	6	1	Para la interconexión de los equipos y poder aplicar la VLan, tal como se muestra en la red.
Controlador	5		Se conectará del *MikroTik* hacia el controlador, para este caso, la máquina virtual que se está ejecutando en una laptop.
Teléfonos		3 y 4	Puertos asignados para los teléfonos
WAN	1		

Fuente: Elaboración propia, 2021.

2. Resultados y discusión

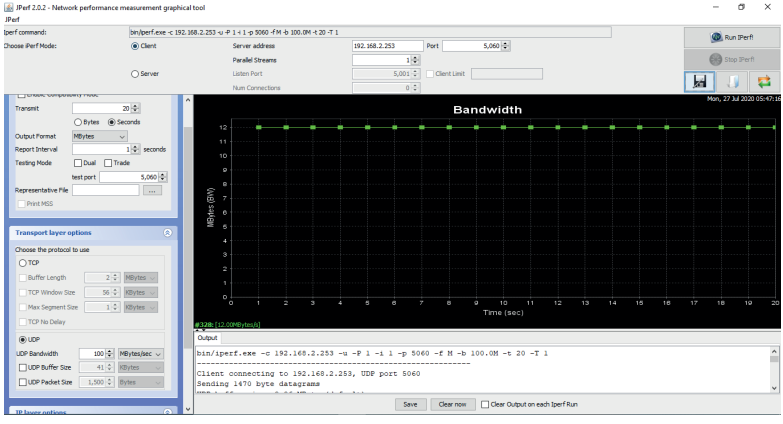
2.1. Jitter y ancho de banda

Utilizando la aplicación *JPerf*, se pudo visualizar el rendimiento de la red evaluado sobre su ancho de banda. Con relación a este escenario, el mismo se ejecutó mediante comandos, como se muestra en la Figura II. En este ejemplo, fue ubicado en el escritorio y se procedió con el ingreso a la carpeta *jperf*-2.0.2, posterior se ejecuta con el comando ./jperf.sh



Fuente: Elaboración propia, 2021.
Figura II: **Comando para ingresar y ejecutar *JPerf***

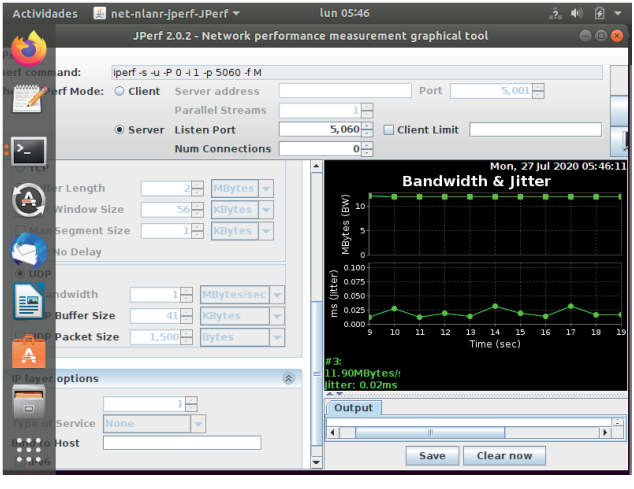
Ejecutada la aplicación, se obtendrán dos modos, cliente y servidor. Asimismo, vendrá un puerto por defecto que es el 5001, para este escenario se colocó 5060, por donde pasa la voz y es admisible con VoIP, y UDP, el cual es el protocolo de transmisión para poder llenar de tráfico la red. Una vez configurado, se ejecuta el programa, evidenciando los resultados como se muestran en la Figura III, obteniendo un ancho de banda (BW) de 12 *Megabytes/sec*, suficiente para evitar las pérdidas y cumplir con la continuidad de los datagramas.



Fuente: Elaboración propia, 2021.

***Figura III: JPerf-Client* con RDS**

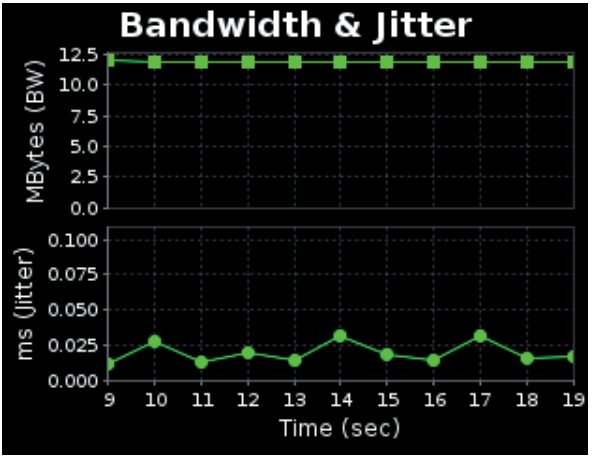
Por el otro extremo, para la configuración del servidor se coloca el puerto 5060 y UDP, para la transmisión y se ejecuta el programa, tal como se aprecia en la Figura IV.



Fuente: Elaboración propia, 2021.

***Figura IV: JPerf-Server* con RDS**

En la Figura V, se puede observar que, durante la transmisión, el *Jitter* alcanzó un valor de 0.030 ms, dando un resultado positivo para las redes RDS y el ancho de banda de 12.5 MgBytes.



Fuente: Elaboración propia, 2021.

Figura V: **Ancho de Banda y *Jitter* con RDS**

2.2. Latencia

Con la finalidad de determinar la latencia, dentro del *Elastix* se ejecutaron 50 repeticiones, con el comando *ping,* entre las direcciones IP de los teléfonos. Para asegurar la interconectividad, se enviaron 500 paquetes desde la central a los teléfonos, estas pruebas se repitieron 50 veces cada una. Primero se comenzó con el teléfono 2, escribiendo el siguiente código dentro de la interfaz de comandos del *Elastix*: Ping 192.168.2.194 -c 500 es el comando utilizado para el teléfono 2; el – c es para enumerar la cantidad de paquetes que se desea enviar, para esta prueba, fueron 500 paquetes.

Obteniendo el 100% de los paquetes llegado a su destino, la latencia que existe (parámetro RTT) es la que se observa en el recuadro verde en la Figura VI, siendo de izquierda a derecha, Latencia: Mínima, media y máxima; y *Jitter*, sus valores, dados en milisegundos, corresponden en el mismo orden después del signo igual.

```
--- 192.168.2.194 ping statistics ---
500 packets transmitted, 500 received, 0% packet loss, time 499675ms
rtt min/avg/max/mdev = 0.915/1.452/6.570/0.546 ms
root@root:~#
```

Fuente: Elaboración propia, 2021.

Figura VI: **Valores de latencia entre la central y el teléfono 2**.

En el Cuadro 7, se aprecian los resultados promedio de las pruebas realizadas, hacia el teléfono 2 cuya IP fue 192.168.2.194, dónde se enviaron y recibieron 500 paquetes, sin pérdida alguna durante todas las pruebas realizadas; además, en cada una de las pruebas se tomó un valor estimado de 499.675 segundos, 64 bytes transmitido en cada paquete.

Cuadro 7
Resultados de latencia y *Jitter* hacia el teléfono 2

Número de paquetes enviados en 50 repeticiones		500
Latencia (ms)	**Mínima**	0.976
	Media	1.633
	Máxima	6.12
Jitter (ms)		0.472

Fuente: Elaboración propia, 2021.

De igual manera, se procedió con los mismos pasos para el teléfono 1, con su correspondiente IP, escribiendo el siguiente código dentro de la interfaz de comandos de *Elastix*: Ping 192.168.2.195 -c 500 es el comando utilizado para el teléfono 1, el – c es para enumerar la cantidad de paquetes que se desea enviar, que, para esta prueba, fueron 500 paquetes. En la Figura VII, se muestra de izquierda a derecha, Latencia (RTT): Mínima, media y máxima; y *Jitter* (ms), sus valores corresponden en el mismo orden después del signo igual.

```
500 packets transmitted, 500 received, 0% packet loss, time 499732ms
rtt min/avg/max/mdev = 0.922/1.434/5.528/0.454 ms
```

Fuente: Elaboración propia, 2021.

Figura VII: **Valores de latencia entre la central y el teléfono 1**

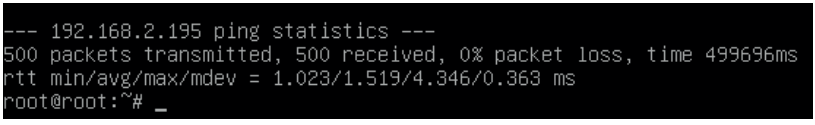
Las pruebas realizadas, hacia el teléfono 1 (192.168.2.195), evidencia los resultados promedios que se detallan en el Cuadro 8, en el cual 500 paquetes fueron enviados y recibidos, sin pérdida alguna durante todas las pruebas realizadas, y en cada prueba se tomó un valor estimado de 499.732 segundos, 64 bytes transmitido en cada paquete.

Cuadro 8
Resultados de latencia y *Jitter* hacia el teléfono 1

Número de paquetes enviados en 50 repeticiones		500
Latencia (ms)	**Mínima**	0.943
	Media	1.567
	Máxima	5.631
Jitter (ms)		0.492

Fuente: Elaboración propia, 2021.

Luego, de comprobada la interconectividad de un teléfono a otro, se procede a la comprobación con ambos teléfonos, para la misma se usó el comando: Ping -c 500 -s 192.168.2.194 192.168.2.195 para la comunicación de los mismos, enviando 500 paquetes cada prueba, siendo el emisor el teléfono 2 y el receptor el 1. En la Figura VIII, se aprecia de izquierda a derecha, Latencia (RTT): Mínima, media y máxima; y *Jitter* (ms), cuyos valores corresponden en el mismo orden después del signo igual.



Fuente: Elaboración propia, 2021.

Figura VIII: **Latencia de la intercomunicación entre el teléfono 2 y el teléfono 1.**

Los resultados de las pruebas, están representados en el Cuadro 9, con valores promedio obtenidos del escenario 2, en el cual se enviaron y recibieron 500 paquetes, sin pérdida alguna durante todas las pruebas realizadas; en cada prueba llevada a cabo se tomó un valor estimado de 499.696 segundos, 200 bytes transmitido en cada paquete.

Cuadro 9
Resultados de latencia y *Jitter* del teléfono 2 al teléfono 1

Número de paquetes enviados en 50 repeticiones		500
Latencia (ms)	Mínima	1.012
	Media	1.678
	Máxima	4.216
Jitter (ms)		0.399

Fuente: Elaboración propia, 2021.

Ahora se realizan las mismas pruebas de forma contraria. Ping -c 500 -s 192.168.2.195 192.168.2.194 para la comunicación de los mismos, enviando 500 paquetes cada prueba, siendo el emisor el teléfono 1 y el receptor el teléfono 2, obteniendo los resultados que se muestran en la Figura IX, de izquierda a derecha, Latencia (RTT): Mínima, media y máxima y *Jitter* (ms), que sus valores corresponden en el mismo orden después del signo igual.

Fuente: Elaboración propia, 2021.

Figura IX: **Latencia de la intercomunicación entre el Teléfono 1 y el Teléfono 2.**

En el Cuadro 10, se muestran los resultados promedios de las pruebas realizadas, dónde 500 paquetes fueron enviados y recibidos, sin pérdida alguna, así como en cada prueba se tomó un valor estimado de 499.668 segundos, 200 bytes transmitido en cada paquete.

Cuadro 10
Resultados de latencia y *Jitter* del teléfono 1 al teléfono 2

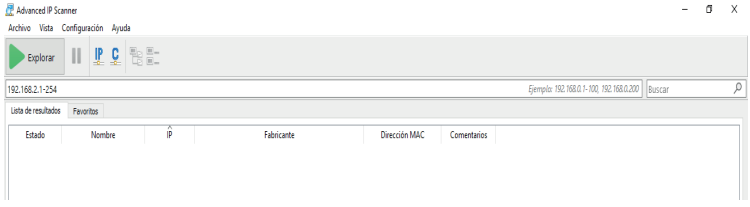
Número de paquetes enviados en 50 repeticiones		500
Latencia (ms)	**Mínima**	1.012
	Media	1.678
	Máxima	4.216
Jitter (ms)		0.399

Fuente: Elaboración propia, 2021.

De igual forma, se realizaron pruebas de conectividad mediante la consola para cada uno de los teléfonos y entre ellos, también, con llamadas telefónicas para comprobar la conexión y paso de voz en las llamadas. Para evidenciar su funcionamiento, se hizo *ping* a cada teléfono por separado, y después de un teléfono a otro. Asimismo, con el fin de comprobar el pase de voz, se hicieron las llamadas en físico, demostrando el funcionamiento de la red.

2.3. Seguridad

Con la finalidad de prevenir ataques a la red, se optó por una medida de seguridad, la cual consiste en tener rangos de direcciones IP distintas, una, para la red de los dispositivos finales; y otra, para la configuración de los mismos (administrador). Dónde el administrador de la red, debe conocer cada una de las direcciones IP, porque a cada dispositivo que se conecte se le asignará una IP dinámica, y se habilitará un puerto. En la Figura X, se procedió hacer uso de la herramienta *Advanced IP*, para confirmar si se arrojaba una dirección IP. Para facilitar los ataques, se escaneó 18 veces la misma, sin resultado alguno. Esto demuestra una ventaja al no facilitar las direcciones establecidas para con los atacantes.

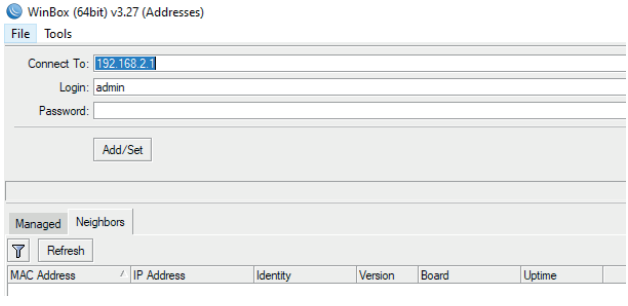


Fuente: Elaboración propia, 2021.

***Figura X: Advanced IP,* para el escaneo de la red**

De igual manera, para comprobar mayor su efectividad, asumiendo que un atacante X sepa o tenga indicios de la IP, se colocó un rango de direcciones IP de la red, 192.168.2.1 hasta la 192.168.2.254 con el fin de detectar algún dispositivo conectado; sin embargo, no arrojó resultados, esta prueba se la repitió 10 veces. De igual forma, se colocó un rango de direcciones 192.168.1.1 hasta la 192.168.1.254 con la finalidad de verificar que no se arroje resultado alguno, en ese sentido, esta prueba se la realizó 8 veces sin resultado alguno.

En otro tipo de configuraciones, se puede dar para cada puerto una dirección IP estática, tal como se muestra en la Figura XI, donde el administrador de la red debe saber a la perfección cuales son, de tal manera que, solo habilitaría los puertos en uso, evitando que otros puertos emitan una dirección IP y el flujo de datos no se daría.



Fuente: Elaboración propia, 2021.

Figura XI: WinBox

Otro método básico, que un atacante podría usar, es conectándose por medio de un puerto disponible con un cable de red. Por lo tanto, se realizó la prueba con direcciones IP estáticas para brindar mayor seguridad, puesto que para poder acceder al *router* por medio de la interfaz de *WinBox* debe tener configurado la dirección IP en su ordenador dentro del rango establecido en las configuraciones. Se conectó una pc al *router* 2011 UiAS, configurada para obtener direcciones DHCP, mostrando la Figura XII, que ninguna dirección o *mac* de algún dispositivo sea reconocida. En el caso que el atacante sepa una dirección IP válida y quiera intentar acceder colocándola en el *WinBox*, tampoco dará resultado alguno. Este

procedimiento se lo realizó a ambos *routers*. Entre más segmentaciones haya en una red y ordenadamente estructurada este, mayor rendimiento tendrá y más difícil será para los intrusos inmiscuirse en los equipos.

Conclusiones

Los resultados evidenciados en la presente investigación, con la finalidad de determinar las herramientas necesarias usadas en las configuraciones, es imprescindible un estudio a fondo sobre los diferentes tipos de controladores y protocolos que se usarán. Una buena elección del controlador *OpenDayLight* y el protocolo *OpenFlow* será de gran ayuda para una estable conexión con los equipos.

Asimismo, la prueba fue realizada en un ambiente controlado, determinando que la red tenía menos pérdidas y retardo en cuestión del intervalo de tiempo de los paquetes, a diferencia de una red telefónica sin VLAN y *OpenFlow*. También, se pudo constatar que la RDS ocupaba casi toda la capacidad máxima de la velocidad de transmisión. Acorde a la seguridad, se pudo percibir varios tipos de seguridad que en un futuro se pueden implementar en las redes RDS con tecnología *Mikrotik*.

Finalmente, dados los resultados obtenidos por el mismo *software* del *router*, se logra observar un mayor fluido del tráfico de voz, minimizando la perdida de paquetes y ocupando la máxima capacidad del canal tanto en transmisión, como en recepción, con minorías de tiempo en el retardo de cada paquete, haciendo que la red sea factible en cuanto a la transmisión Voz IP.

Referencias Bibliográficas

AbdelAzim, N. M., Fahmy, S. F., Sobh, M. A., y Bahaa, A. M. (2021). A hybrid entropy-based DoS attacks detection system for software defined networks (SDN): A proposed trust mechanism. *Egyptian Informatics Journal, 22*(1), 85-90. https://doi.org/10.1016/j.eij.2020.04.005

Adil, B., Abderrahmane, A., Mourad, A., y Lallouani, B. (2018). Automatic speech recognition for VoIP with packet loss concealment. *Procedia Computer Science, 128*, 72-78.

Aguirre, J., y Ortega, E. (2005). *La calidad del servicio como uno de los elementos formadores de imagen. Estudio de caso: Telmex-Maxcom* (Tesis de pregrado). Universidad de las Américas Puebla, Puebla, Mexico.

Arbeláez-Campillo, D. F., Villasmil, J. J., y Rojas-Bahamón, M. J. (2021). Inteligencia artificial y condición humana: ¿Entidades contrapuestas o fuerzas complementarias? *Revista de Ciencias Sociales (Ve), XXVII*(2), 502-513. https://doi.org/10.31876/rcs.v27i2.35937

Briñez, M. (2021). Tecnología de información: ¿Herramienta potenciadora para gestionar el capital intelectual? *Revista de Ciencias Sociales (Ve), XXVII*(1), 180-192. https://doi.org/10.31876/rcs.v27i1.35305

Caldera, J. C., y Suazo, W. E. (2011). *Planeación de un curso especializado en telefonía para profesionales de la industria de telecomunicaciones: Módulo III Telefonía IP.* http://repositoriosiidca.csuca.org/Record/RepoUNI1261

Escobar, J. A. (2015). *Control de flujo de una red definida por software usando sensores térmicos* (Tesis de pregrado). Universidad Técnica de Ambato, Ambato, Ecuador.

Fernandez, M. M., y Ulloa, R. F. (2016). *Despliegue de una red SDN aplicando el protocolo MPLS y generando políticas de QoS para servicios de telefonía IP* (Tesis de pregrado).

Universidad Politécnica Salesiana, Cuenca, Ecuador.

Flores, J. R. (2018). *Contribución a las arquitecturas de virtualización de funciones de red y redes definidas por sotware aplicadas a las redes residenciales con gestión centrada en el usuario* (Tesis doctoral). Universidad Politécnica de Madrid, Madrid, España.

Koerner, M., y Kao, O. (2016). MAC based dynamic VLAN tagging with OpenFlow for WLAN access networks. *Procedia Computer Science, 94*, 497-501. https://doi.org/10.1016/j.procs.2016.08.077

Li, W., Meng, W., y Kwok, L. F. (2016). A survey on OpenFlow-based Software Defined Networks: Security challenges and countermeasures. *Journal of Network and Computer Applications, 68*, 126-139. https://doi.org/10.1016/j.jnca.2016.04.011

Moreno, D. R., y Zambrano, A. M. (2018). *Analisis comparativo del desempeño computacional de una aplicacion distribuida intensiva en datos en redes de dispositivos y redes definidas por software* (Tesis de pregrado). Escuela Politécnica Nacional (EPN), Quito, Ecuador.

Moscoso, E. M. (2016). *Desarrollo de una aplicación para la implementación de calidad de servicio por priorización de tráfico sobre una red definida por software (SDN)* (Tesis de pregrado). Escuela Politécnica Nacional (EPN), Quito, Ecuador.

Moyolema, S. R. (2015). *Diseño de un sistema de VOZ/IP para un call center en el Hospital Docente de la Policía Nacional Guayaquil No. 2* (Tesis de pregrado). Universidad de Guayaquil, Guayaquil, Ecuador.

OpenNetworking (2020). Transforming access and edge networks by collaboratively building next generation mobile and broadband infrastructures, leveraging. *OpenNetworking ONF.* https://opennetworking.org/

Orgaz, C. J. (31 de Mayo de 2019). 3 grandes ventajas que traerá la tecnología 5G y que cambiarán radicalmente nuestra experiencia en internet. *BBC News Mundo*. https://www.bbc.com/mundo/noticias-48477358

Oviedo, B., Zhuma, E., Guzman, D., y Cáceres, C. (2020). Análisis del desempeño de redes definidas por software frente a redes con arquitectura TCP/IP. *RISTI*, *(*E31), 137-150.

Parra, R., Morales, V. M., y Hernández, J. I. (2015). Redes Definidas por Software: Beneficios y riesgos de su implementación en Universidades. *Tecnología Educativa: Revista CONAIC, 2*(3), 48-54. https://doi.org/10.32671/terc.v2i3.153

Tchernykh, A., Cortés-Mendoza, J. M., Bychkov, I., Feoktistov, A., Didelot, L., Bouvry, P., Radchenko, G., y Borodulin, Kirill (2019). Configurable cost-quality optimization of cloud-based VoIP. *Journal of Parallel and Distributed Comnputing, 133,* 319-336 https://doi.org/10.1016/j.jpdc.2018.07.001