



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

**ESCUELA DE INGENIERÍA ELECTRÓNICA EN TELECOMUNICACIONES
Y REDES**

**“EVALUACIÓN DE RENDIMIENTO E INTEROPERABILIDAD DE
UNA PLATAFORMA HW/SW DE ROUTING OPENSOURCE”**

TRABAJO DE TITULACIÓN: DISPOSITIVO TECNOLÓGICO

Para optar al Grado Académico de:

**INGENIERO EN ELECTRÓNICA, TELECOMUNICACIONES Y
REDES**

AUTORES: ANDRÉS SEBASTIÁN LAGUA GAVILANES

JUAN CARLOS LUGMANIA BALLADAREZ

TUTOR: ING. ALBERTO ARELLANO AUCANCELA

Riobamba – Ecuador

2017

@2017, Andres Sebastian Laguna Gavilanes y Juan Carlos Lugmania Balladarez

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA ELECTRÓNICA EN TELECOMUNICACIONES Y

REDES

El tribunal del trabajo de titulación certifica que: El trabajo de titulación: EVALUACIÓN DE RENDIMIENTO E INTEROPERABILIDAD DE UNA PLATAFORMA HW/SW DE ROUTING OPENSOURCE”, de responsabilidad de los señores ANDRÉS SEBASTIÁN LAGUA GAVILANES Y JUAN CARLOS LUGMANIA BALLADAREZ, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Titulación, quedando autorizada su presentación.

NOMBRE	FIRMA	FECHA
Ing. Washington Luna.		
DECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
Ing. Franklin Moreno		
DIRECTOR DE LA ESCUELA DE INGENIERÍA ELECTRÓNICA EN TELECOMUNICACIONES Y REDES
Ing. Alberto Arellano Aucancela.		
DIRECTOR DEL TRABAJO DE TITULACIÓN
Ing. Oswaldo Martínez		
MIEMBRO DEL TRIBUNAL

Nosotros, Andrés Sebastián Laguna Gavilanes, con cédula de identidad 180382398-6, Juan Carlos Lugmania Balladarez, con cédula de identidad 050288165-9, declaramos que el presente trabajo de titulación es de nuestra autoría y que los resultados del mismo son auténticos y originales. Los textos constantes en el documento que provienen de otra fuente están debidamente citados y referenciados.

Como autores, asumimos la responsabilidad legal y académica de los contenidos de este trabajo de titulación.

Riobamba, 01 de noviembre de 2017

Andrés Sebastián Laguna Gavilanes

ESTUDIANTE

Juan Carlos Lugmania Balladarez

ESTUDIANTE

Ing. Alberto Arellano Aucancela

DIRECTOR TRABAJO DE TITULACIÓN

Ing. Oswaldo Martínez Guashima

MIEMBRO TRIBUNAL

Nosotros, **ANDRÉS SEBASTIÁN LAGUA GAVILANES** y **JUAN CARLOS LUGMANIA BALLADAREZ**, somos responsables de las ideas, doctrinas y resultados, expuestos en este trabajo, y el patrimonio intelectual del Trabajo de Titulación pertenece a la **ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**.

Andrés Sebastián Laguna Gavilanes

Juan Carlos Lugmania Balladarez

DEDICATORIA

Quiero dedicar con mucho cariño este trabajo de titulación, a mi Padre Ángel Laguna, quien siempre con sus consejos y su apoyo me ha enseñado a ser una mejor persona cada día, a mi amada madre Silvia Gavilanes quien se convirtió en el pilar fundamental de mi vida y que siempre ha buscado mi superación, gran parte de este logro es gracias a ellos dos.

A mis hermanos María y Luis, quienes siempre han sido un apoyo fundamental en mi vida, a mi sobrina Pamela Pérez, ella con el tiempo ha sabido convertirse en una hermana más, esperando con esto poder inculcar un buen ejemplo en ella.

A una persona muy especial Estefanía Merino, y a todos mis amigos sobre todo a Patricio Lizano y Javier Sacán, quienes ha compartido grandes momentos de alegrías y tristezas conmigo, brindándome soporte en los momentos y las circunstancias a lo largo de la carrera que más lo necesite.

A mí querida mascota Lucky, que a pesar de ya no estar conmigo se convirtió en una parte muy especial en mi vida y siempre estará conmigo en todo momento

Andrés Sebastián Laguna Gavilanes

Esta tesis se la dedico a mi madre Georgina Balladarez por ser el pilar más importante de mi vida, ya que con su fortaleza, enseñanza, consejo, afecto y con su sacrificio hicieron posible la finalización de esta etapa de mi vida. A mi abuelita Piedad Balladarez, con su humildad, carisma, apoyo incondicional y sobre todo amor he podido alcanzar mi meta. A mi padre Washington Lugmania, con sus palabras y recursos han sabido apoyarme. A mis hermanos Santiago y Andrés por estar siempre a mi lado en las buenas y en las malas para poder seguir adelante a pesar de las adversidades.

A mi mejor amiga Cristina Barragán que ha dejado un vacío enorme en mi vida desde su partida que nadie jamás podrá llenar, fue la única persona que estuvo junto a mí en los peores momentos brindándome su apoyo, muchas veces compartiendo circunstancias que nos unieron mucho más. Los recuerdos que me has dejado siempre me acompañarán y te recordare por el resto de mi vida.

A mis amigos de la infancia Cristian Chiluisa (CH) y Santiago Chariguaman (Chory), gracias a sus ocurrencias hicieron de esta experiencia una de las más especiales y a mi familia en general.

Juan Carlos Lugmania Balladarez

AGRADECIMIENTO

Agradecemos a Dios por habernos permitido culminar esta etapa importante de nuestras vidas, a nuestros queridos padres y hermanos porque han sido un pilar fundamental y nos han brindado un apoyo incondicional a lo largo de nuestra carrera. A nuestros amigos, quienes sin esperar nada a cambio compartieron con nosotras grandes momentos, supieron estar a nuestro lado e hicieron de esta experiencia una de las más especiales.

De la misma manera, a la Escuela Superior Politécnica de Chimborazo, así como también a la Escuela de Ingeniería Electrónica en Telecomunicaciones y Redes ya que durante esta etapa se convirtió en nuestro segundo hogar.

Con una expresión de gratitud a nuestros profesores quienes nos incentivaron en muchos sentidos a seguir adelante, guiándonos con sus conocimientos para alcanzar esta meta. Y sobre todo a nuestro amigo y estudiante Javier Sacán y al Ingeniero Alberto Arellano quienes supieron guiarnos, con su apoyo incondicional, conocimientos y su ayuda han hecho posible la culminación con éxito del presente trabajo de titulación.

Andrés Laguna y Juan Lugmania

TABLA DE CONTENIDOS

ÍNDICE DE TABLAS.....	x
ÍNDICE DE FIGURAS.....	xi
ÍNDICE DE GRÁFICOS.....	xiii
ÍNDICE DE FOTOGRAFÍAS.....	xiv
RESUMEN.....	xv
SUMARY	xvi
INTRODUCCIÓN	1
CAPÍTULO 1	
1. MARCO TEÓRICO REFERENCIAL	6
1.1. Redes de Computadoras	6
1.1.1. <i>Introducción</i>	6
1.2. Protocolos de enrutamiento.	6
1.2.1. <i>Enrutamiento estático</i>	7
1.2.2. <i>Enrutamiento dinámico</i>	8
1.3. Sistemas Autónomos.....	9
1.3.1. <i>BGP</i>	10
1.4. Plataformas de enrutamiento.....	12
1.4.1. <i>Plataformas de enrutamiento OpenSource</i>	13
1.4.1.1. <i>Router modular click (click modular router)</i>	14
1.4.1.2. <i>Vyatta System</i>	14
1.4.1.3. <i>Quagga Routing Software Suite</i>	14
1.4.1.4. <i>XORP (eXtensible Open Router Platform)</i>	15
1.4.1.5. <i>OpenWrt</i>	15
1.4.1.6. <i>FRRouting</i>	15
1.5. Tarjeta Madre (Mainboard)	16
1.6. Software de verificación.....	18

1.6.1.	Generadores de tráfico	18
1.6.1.1.	<i>Ostinato</i>	18
1.6.2.	Analizadores de tráfico	20
1.6.2.1.	<i>Tipos de analizadores</i>	21
1.6.2.2.	<i>Wireshark</i>	23
1.6.3.	Generador de rutas vRIN	25

CAPÍTULO 2

2.	MARCO METODOLÓGICO	26
2.1.	Introducción	26
2.2.	Parámetros a determinar	26
2.2.1.	Análisis de Hardware	28
2.2.1.1.	<i>Tipos de tarjetas madres afines</i>	28
2.2.2.	Análisis de Software	30
2.2.2.1.	<i>Software de enrutamiento</i>	30
2.2.2.2.	<i>Software de pruebas</i>	31
2.2.3.	Selección y obtención de tecnologías HW/SW	32
2.2.3.1.	<i>Selección y obtención de tecnología HW</i>	32
2.2.3.2.	<i>Selección y obtención de tecnologías SW</i>	32
2.2.4.	Instalación de Hardware	32
2.2.4.1.	<i>Conexión de componentes</i>	32
2.2.4.2.	<i>Ensamblaje de componentes</i>	33
2.2.5.	Instalación de Software	35
2.2.5.1.	<i>Instalación de Ubuntu Server 16.04.3 LTS</i>	35
2.2.5.2.	<i>Instalación FRRouting</i>	35
2.2.6.	Pruebas de Interoperabilidad	38
2.2.6.1.	<i>Prueba 1: OSPF en IPv4</i>	39
2.2.6.2.	<i>Prueba 2: OSPF y RIPv2 aplicando redistribución de rutas en IPv4</i>	40
2.2.6.3.	<i>Prueba 3: OSPF en IPv6</i>	40

2.2.6.4.	<i>Prueba 4: OSPF y RIPng aplicando redistribución de rutas en IPv6</i>	41
2.2.7.	<i>Pruebas de Rendimiento</i>	42
2.2.7.1.	<i>Prueba 5: Generación de tráfico con Ostinato en IPv4.</i>	44
2.2.7.2.	<i>Prueba 6: Generación de tráfico con Ostinato en Ipv6</i>	44
2.2.7.3.	<i>Prueba 7: Generación de rutas BGP con vRIN en IPv4</i>	45

CAPÍTULO 3

3.	MARCO DE RESULTADOS, ANÁLISIS Y DISCUSIÓN DE RESULTADOS	47
3.1.	Análisis de resultados obtenidos	47
3.1.1.	<i>Pruebas de interoperabilidad</i>	47
3.1.1.1.	<i>Prueba 1: OSPF en IPv4</i>	47
3.1.1.2.	<i>Prueba 2: OSPF y RIPv2 aplicando redistribución de rutas en IPv4</i>	51
3.1.1.3.	<i>Prueba 3: OSPF en IPv6</i>	53
3.1.1.4.	<i>Prueba 4: OSPF y RIPng aplicando redistribución de rutas en IPv6</i>	55
3.1.1.5.	<i>Resultados finales de interoperabilidad</i>	58
3.1.2.	<i>Pruebas de rendimiento</i>	59
3.1.2.1.	<i>Prueba 5: Generación de tráfico con Ostinato en IPv4.</i>	59
3.1.2.2.	<i>Prueba 6. Generación de tráfico con Ostinato en IPv6</i>	63
3.1.2.3.	<i>Prueba 7: Análisis con vRIN en IPv4</i>	66
3.2.	Comparación de precios de plataformas de enrutamiento	69
	CONCLUSIONES	70
	RECOMENDACIONES	71

BIBLIOGRAFÍA

ANEXOS

ÍNDICE DE TABLAS

Tabla 1-1: Clasificación Plataformas Enrutamiento	13
Tabla 2-1: Clasificación de tarjetas madre	17
Tabla 1-2: Requerimientos mínimos.....	29
Tabla 2-2: Calificación final de requerimientos.	29
Tabla 3-2: Requerimientos necesarios	30
Tabla 4-2: Calificación Routing Software.....	31
Tabla 5-2: Pasos de Instalación	35
Tabla 1-3: Resultados finales	58
Tabla 2-3: Tiempos de conmutación en IPv4	62
Tabla 3-3: Tiempos de conmutación en IPv6	65
Tabla 4-3: Memoria consumida.....	69
Tabla 5-3: Comparación de precios	69

ÍNDICE DE FIGURAS

Figura 1-1: Enrutamiento estático configuración	8
Figura 2-1: Interconexión de Sistemas Autónomos.....	10
Figura 3-1: Conexión BGP.....	11
Figura 4-1: Clasificación Protocolos de Enrutamiento Dinámico	12
Figura 5-1: Interfaz De Ostinato.....	20
Figura 6-1: Analizador de tráfico mediante línea de comandos	22
Figura 7-1: Analizador de tráfico mediante interfaz gráfica	22
Figura 8-1: Analizador de tráfico de tipo hardware.....	23
Figura 9-1: Interfaz de Wireshark	24
Figura 10-1: Interfaz de vRIN.	25
Figura 1-2: Escenario protocolo OSPF.....	38
Figura 2-2: Escenario OSPF en IPv4.....	39
Figura 3-2: Escenario redistribución OSPF & RIP.....	40
Figura 4-2: Escenario OSPF en IPv6.....	41
Figura 5-2: Escenario redistribución OSPF & RIP en IPv6	42
Figura 6-2: Escenarios para prueba de rendimiento	43
Figura 7-2: Escenarios para prueba de rendimiento mediante generación de rutas.....	43
Figura 8-2: Escenarios para prueba de rendimiento	44
Figura 9-2: Escenarios para prueba de rendimiento	45
Figura 10-2: Ejemplo de generación de rutas con BGP	46
Figura 1-3: Ping desde PC1.....	48
Figura 2-3: Traceroute desde PC1	48
Figura 3-3: Ping desde PC2.....	49
Figura 4-3: Traceroute desde PC2.....	49
Figura 5-3: Configuración Pc1	57
Figura 6-3: Pruebas de conectividad en Pc1	57

Figura 7-3: Inyección de tráfico en Ostinato.....	59
Figura 8-3: Wireshark Escenario 1 Jetway	60
Figura 9-3: Wireshark Escenario 2 Jetway	60
Figura 10-3: Wireshark Escenario 1 Cisco	61
Figura 11-3: Wireshark Escenario 2 Cisco	61
Figura 12-3: Wireshark Escenario 1 Jeway.....	63
Figura 13-3: Wireshark Escenario 2 Jetway.....	64
Figura 14-3: Wireshark Escenario 1 Cisco	64
Figura 15-3: Wireshark Escenario 2 Cisco	65
Figura 16-3: Generación de 120 rutas en Cisco	67
Figura 17-3: Generación de 200 rutas en Cisco	68

ÍNDICE DE GRÁFICOS

Gráfico 1-2: The Deming Cycle	27
Gráfico 2-2: The Deming Cycle	27
Gráfico 1-3: Tiempo de conmutación en IPv4	62
Gráfico 2-3: Tiempo de conmutación en IPv6	66

ÍNDICE DE FOTOGRAFÍAS

Fotografía 1-2: Conexión de la tarjeta madre	33
Fotografía 2-2: Fabricación del armazón	34
Fotografía 3-2: Instalación de componentes	34
Fotografía 1-3: Ping desde Plataforma	50
Fotografía 2-3: Traceroute desde Plataforma	51
Fotografía 3-3: Configuración de RIPv2 en Jetway	52
Fotografía 4-3: Pruebas de conectividad	52
Fotografía 5-3: Análisis de saltos de rutas	53
Fotografía 6-3: Configuración IPv6	53
Fotografía 7-3: Conectividad con OSPF en IPv6	54
Fotografía 8-3: Análisis de saltos en IPv6	54
Fotografía 9-3: Configuración OSPF y RIPng en IPv6	55
Fotografía 10-3: Pruebas de conectividad OSPF y RIPng	56
Fotografía 11-3: Pruebas de conectividad OSPF y RIPng	56
Fotografía 12-3: Generación de 120 rutas en Jetway	67
Fotografía 13-3: Generación de 200 rutas en Jetway	68

RESUMEN

El objetivo del trabajo de titulación fue el análisis de rendimiento e interoperabilidad de la plataforma Hardware/Software (HW/SW) de routing opensource. Se instaló un sistema operativo denominado FRR de software libre en un mainboard de marca JETWAY y se realizaron diferentes pruebas de rendimiento e interoperabilidad con otros dispositivos de red. Utilizamos el método científico en el desarrollo del software. La plataforma Hw/Sw cuenta con un Sistema Operativo Linux-Ubuntu 16.04 en el cual se configuró FRRouting; un conjunto de protocolos de enrutamiento IP que incluye daemons de protocolos de enrutamiento dinámico, siendo una versión actualizada de Quagga. Se hicieron varias pruebas de interoperabilidad ejecutando protocolo de internet versión 4 (IPv4) y protocolo de internet versión 6 (IPv6), en diversos escenarios, con varios protocolos de enrutamiento dinámicos existentes en la actualidad y redistribución, para el escenario se usaron equipos de prueba de fabricantes diferentes como Mikrotik, Cisco. También se verificó el rendimiento del terminal estudiando dos casos muy importantes: al recibir tráfico inyectado mediante la herramienta de pruebas Ostinato y al determinar el comportamiento en dicha prueba mediante Wireshark obteniéndose la velocidad de transmisión, la cantidad de paquetes perdidos, etc. También se utilizó Inyector de Ruta Virtual (vRIN), un sistema generador, que permitió enviar peticiones a las rutas creadas y conocer la velocidad de transmisión a la que funciona el dispositivo, estas pruebas fueron comparadas con un escenario de prueba de iguales características, de marca Cisco. El dispositivo ha logrado ser 100% interoperable y de un alto rendimiento frente a diferentes escenarios planteados, este equipo representa una opción mucho más económica y rentable para empresas que no requieran de un área dedicada a redes avanzadas. Es recomendable verificar la compatibilidad de los sistemas opensource con las placas de hardware, en el momento de instalar estas plataformas. .

PALABRAS CLAVE: <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <REDES DE COMPUTADORES>, <INTEROPERABILIDAD DE REDES DE COMPUTADORES>, <RENDIMIENTO>, <PROTOCOLOS DE ENRUTAMIENTO>, <TRÁFICO DE DATOS>, <PAQUETES>, <HARDWARE>, <SOFTWARE>..

SUMMARY

This work aimed at analyzing the performance and interoperability of Hardware/Software (HW/SW) open source platform. An operating system called FRRouting (FRR) free software was installed in a Jetway mainboard and several performance and interoperability tests with other network devices were done. The scientific method was used to develop the software. The HW/SW platform has the Linux-Ubuntu 16.04 operating system, in which FRR – a set of routing protocols IP including daemons for dynamic routing protocols that is a current version of Quagga – was configured. Several interoperability tests were carried out using an internet protocol version 4 (IPv4) and internet protocol version 6 (IPv6) in different scenarios with several current existing dynamic routing protocols and with redistribution. For the scenario, testing equipment from different developers, such as Mikrotik, Cisco were used. The terminal performance was verified studying two important cases: receiving injected traffic from the Ostinato testing tools and determining the behavior during the test with Wireshark; the transmission speed and the quantity of packet loss, and so on were obtained. A virtual route injector (vRIN) – a generator system that permitted to send created route requests and to know the device transmission speed – was used. These tests were compared to Cisco testing scenario including similar features. The device is 100% interoperable and has a high performance in different posed scenarios. This equipment is less expensive and more profitable for enterprises that do not require specific areas of advanced networks. It is recommended to verify the compatibility of the open source systems with hardware mainboard when installing them.

KEYWORDS: <TECHNOLOGY AND ENGINEERING SCIENCES>, <COMPUTER NETWORKS>, <INTEROPERATION OF NETWORK OF COMPUTERS>, <PERFORMANCE>, <ROUTING PROTOCOLS>, <DATA TRAFFIC>, <PACKETS>, <HARDWARE>, <SOFTWARE>.

INTRODUCCIÓN

Como todos sabemos internet no es un fenómeno nuevo, pero si tenemos la seguridad que ha revolucionado al mundo totalmente a como se veía a 30 años atrás, se ha convertido en un fenómeno global, y se vincula estrechamente con las comunicaciones, este fenómeno influye ampliamente en la mayor parte de los ámbitos de la sociedad. Nadie se imaginaba a sus inicios que 20 años después se convertiría en un invento tan necesario como el teléfono o la televisión.

Un avance científico o novedad que hace décadas tardaba años en poder implantarse a nivel mundial, puede ser ahora copiado y distribuido en cuestión de horas.

El internet ha revolucionado no solo en el ámbito científico o empresarial, también se ha vinculado al ámbito más personal y ha logrado abrir nuevas experiencias en el uso de mensajería instantánea como son las redes sociales o los chats.

Según el artículo científico de Manuel Castells Internet y la Sociedad de la Red presentado en la Universidad de Catalunya “La geografía de los usuarios hoy día se caracteriza todavía por tener un alto nivel de concentración en el mundo desarrollado. En ese sentido, digamos que las tasas de penetración de Internet se acercan al 50% de la población en Estados Unidos, en Finlandia y en Suecia, están por encima del 30-35% en Gran Bretaña y oscilan entre el 20-25% en Francia y Alemania. Luego está la situación española en torno a un 14%, Cataluña un 16-17%. En todo caso, los países de la OCDE en su conjunto, el promedio de los países ricos, estarían, en estos momentos, en un 25-30%, mientras que, en el conjunto del planeta, está en menos del 3% y, obviamente, si analizamos situaciones como la africana, como la de Asia del sur, está en menos del 1% de la población. En primer lugar, existe una gran disparidad de penetración en el mundo, pero, por otro lado, las tasas de crecimiento en todas partes, con excepción de África subsahariana, son altísimas, lo cual quiere decir que los núcleos centrales, también en el mundo subdesarrollado, estarán conectados dentro de cinco a siete años a Internet” (INTERNET Y LA SOCIEDAD RED, 2001 pág. 5), esto es una consecuencia de los altos costos para una implementación de redes con dispositivos de grandes marcas comerciales como Cisco o Juniper.

Durante la búsqueda de información relacionada con el presente trabajo de titulación, se pudo constatar que existen trabajos de titulación, tesis, proyectos de fin de carrera, artículos en revistas, que realizan la evaluación de redes de comunicaciones a través de equipos de interconexión. Entre

los más importantes y los más valorados en el mercado tenemos: Cisco, Juniper, Alcatel-Lucent y Huawei.

El trabajo de titulación desarrollara una plataforma de enrutamiento de hardware y software en código abierto para poder inter operar con dispositivos de diferentes fabricantes.

En el campo de la configuración de equipos de enrutamiento, los administradores de red siempre se han enfrentado al problema de falta de plataformas abiertas que permitan hacer pruebas de los diferentes algoritmos y protocolos nacies. En general, las herramientas existentes no son abiertas y solo permiten configuraciones ya establecidas, sin poder hacer cambios significativos que permitan probar nuevas funcionalidades.

Un Router interconecta redes de computadoras y las plataformas para ellos no han sido, tradicionalmente, sistemas abiertos. Debido a esto una gran parte del trabajo que realizan los investigadores de redes es a través de simulaciones y se desconoce cuánta aplicación práctica pueda tener. A su vez, la compra de un router profesional es muchas veces imposible por parte de usuarios de redes, por lo que una solución factible es utilizar las computadoras para que hagan las veces de routers, pero las plataformas para ello no son sencillas de implementar. (Sánchez, 2004, p. 1) En este proyecto se introduce FRROUTING como una plataforma viable para el desarrollo de routers

Entre las principales ventajas que posee FRRouting es poder contar con un soporte BGP Secure Routing Extension BGP-SRx, un parámetro muy indispensable en una plataforma de enrutamiento, este al trabajar con interoperabilidad analiza los diversos casos que se presenten en la red de tráfico continuo.

Para alcanzar un buen rendimiento se definirá varias prioridades para las diferentes partes de FRRouting, en el nivel de usuario se ha priorizado la extensibilidad y la robusticidad sobre el rendimiento. Ocurre al contrario en el subsistema de bajo nivel, donde se ha priorizado el rendimiento, pues en este subsistema se llevan a cabo algoritmos sobre cada uno de los paquetes y brinda las funcionalidades a las capas superiores, pero esta no es manipulada directamente por usuarios (aplicaciones o personas), por lo que su prioridad es el mejor rendimiento del sistema.

Este proyecto pretende desarrollarse sobre una plataforma de enrutamiento interoperable de código abierto y luego realizar un estudio más detallado, mostrando sus características, proceso de instalación, funcionalidades, limitaciones y algunas prácticas sencillas de su funcionamiento.

FORMULACIÓN DEL PROBLEMA

¿Cuál es el nivel de rendimiento e interoperabilidad de la plataforma de Routing OpenSource a desarrollarse?

¿Cuáles son las plataformas de Routing OpenSource existentes?

¿Cuál es porcentaje y como se midió la interoperabilidad de la plataforma a implementarse?

¿Cuál es el porcentaje de rendimiento y su comportamiento ante el tráfico de datos?

¿Cuáles son las herramientas de medida utilizadas para obtener el porcentaje de rendimiento de la plataforma desarrollada?

JUSTIFICACION

Hoy en día, una plataforma de código abierto que permita hacer pruebas con algoritmos o protocolos que están implantando en el campo de las redes es casi imposible encontrar; ya que al permitir modificar nuevas funcionalidades exige requerimiento tanto de hardware como de software.

Considerando que, en los últimos años el crecimiento de las redes ha sido algo exorbitante. Se requiere una plataforma que permita probar las diferentes funcionalidades protocolos que se están creando cada día.

Mediante las observaciones que se ha realizado sobre las diferentes plataformas de enrutamiento se ha tomado en consideración que FRRouting es la más adecuada, y se ha tomado en cuenta tres aspectos muy importantes:

- Puede ejecutar un protocolo de enrutamiento OSPF o BGP de manera muy versátil.
- Hoy en día es una plataforma de enrutamiento muy utilizada y conocida por los desarrolladores de OpenSource.

- La sintaxis de configuración FRRouting es casi idéntica a la sintaxis de Cisco IOS.

Para la implementación del Hardware y después de una búsqueda exhaustiva se ha visto conveniente utilizar la tarjeta “Jetway NF533D4-1900 Intel Celeron J1900 3.5" SBC Motherboard w/ 4 x Intel GB LAN” ya que su costo no es tan elevado frente a diferentes Mainboard con similares características; y de acuerdo con las especificaciones técnicas del Mainboard Intel, esta tarjeta es la más adecuada para nuestro proyecto.

Una vez implementado la plataforma de enrutamiento, para realizar las pruebas de interoperabilidad se presentarán varios escenarios propuestos, y para evaluar el rendimiento utilizaremos un generador de tráfico y rutas y analizaremos su comportamiento en la red.

Gracias a la investigación denominada INFLUENCIA DEL TAMAÑO DE PAQUETES SOBRE LA PÉRDIDA DE PAQUETES EN UN ENLACE por autoría de Nomar Alexander Noroño Rodríguez, “Ostinato nos muestra una gran ventaja, la cual es que tiene soporte para Linux y Windows, además ayuda con una interfaz amigable y permite crear diferentes tipos de tráfico a la vez y genera reportes detallados de los eventos tiempo real”. (Noroño, 2012, p. 2)

También la Investigación nombrada FUNDAMENTACIÓN TEÓRICA Y EXPLORACIÓN DE GENERADORES DE TRÁFICO MULTIPLATAFORMA realizada por los autores ANDRÉS FELIPE CASTAÑO MEJÍA, LINA MARCELA OCAMPO PELÁEZ, MICHAEL STEVENS OROZCO PIEDRAHITA los cuales compararon diferentes generadores de tráfico en sus conclusiones citan que: “Ostinato es un buen generador de tráfico, ya que maneja diferentes tipos de tráfico como IPV4, SIP, HTTP, TCP, UDP, ICMP, IGMP, NNTP y entrega resultados detallados debido a su trabajo conjunto con Wireshark; además tiene una GUI muy amigable, lo que facilita la configuración de direcciones origen y destino y número de paquetes a la hora de generar diferentes tipos de tráfico; por otro lado es un generador de tráfico más fácil de manipular para el usuario, ya que está bajo la plataforma Windows”. (Castaño, et al., 2011, p. 80)

En vista que Ostinato ofrece varias prestaciones como la creación y configuración de múltiples flujos, Hemos considerado conveniente la utilización de este software con licencia GNU para la creación de tráfico en nuestros escenarios propuestos, además como herramienta complementaria utilizaremos WireShark para analizar la información generada por dicho escenario.

En el presente proyecto, se buscará una manera eficaz para el desarrollo de una plataforma de enrutamiento sobre un Mainboard de bajo costo, que posea ciertas características específicas como el número de puertos LAN, esto debido que en la actualidad para los desarrolladores de protocolos naces no existe una plataforma de código abierto que permita hacer pruebas y analizar dichos protocolos.

Por consiguiente, en dicho proyecto y mediante los escenarios propuestos se desea evidenciar todos los posibles casos que se puedan dar en una transmisión de datos en una red como pueden ser: perdidas de paquetes, latencia, jitter, etc. De esta manera poder administrar correctamente los recursos disponibles.

OBJETIVOS

OBJETIVOS GENERALES

Evaluar el rendimiento e interoperabilidad de una plataforma HW/SW de Routing OpenSource.

OBJETIVOS ESPECÍFICOS

- Estudiar las soluciones de Routing OpenSource y Comerciales para determinar sus ventajas y desventajas.
- Desarrollar la plataforma de HW/SW de Routing.
- Implementar un escenario de pruebas de interoperabilidad para garantizar el nivel de rendimiento del mismo.
- Verificar el adecuado funcionamiento de la plataforma de desarrollo.

CAPÍTULO 1

MARCO TEÓRICO REFERENCIAL

1.1. Redes de Computadoras

1.1.1. *Introducción*

En el siglo XX, en las décadas de los 70 y 80 se dio una sinergia entre campo informático y el de Telecomunicaciones, dando a lugar un cambio drástico en el área tecnológica que pudo comunicar información más detallada entre computadoras denominado Redes de Computadores gracias a aparición del modelo TCP/IP. (Stallings, 2004 pág. 10), esta revolución nos ha brindado hasta la actualidad grandes hechos como:

- Existe una gran similitud entre el procesamiento de la información en los PCs y las comunicaciones de datos de los sistemas de conmutación; así como también la transmisión de datos, de audio o multimedia es muy similar en sus características de procesamiento. Razón por la cual las diferencias entre las redes de área local, metropolitanas y de área amplia, se han difuminado.
- Por otra parte las organizaciones de normalización técnicas y tecnológicas tienden q los sistemas fabricados se hagan accesibles de manera fácil y uniforme a escala mundial.

1.2. Protocolos de enrutamiento.

Son un conjunto de reglas utilizadas que permiten a un router comunicarse con otro router y así poder compartir información del enrutamiento, estas tareas se realizan con el fin de generar las tablas de enrutamiento.

Actualmente existe dos tipos de enrutamiento el estático y el dinámico los cuales explicaremos a continuación

1.2.1. *Enrutamiento estático*

En este tipo de enrutamiento la información de todos los routers debe introducirse manualmente y no es fácil mantener las tablas de enrutamiento estáticas, esto es porque el router no puede actualizarse por sí solo a los cambios que se den en la topología de la red.

Sin embargo este tipo de enrutamiento es muy ventajoso en situaciones como al instalar un servidor ISP, al existir una sola conexión con el servicio el cliente no debe conocer la información detrás de servicio. (GuilleSQL, 2007, p. 1)

También existe el enrutamiento predeterminado, una ruta estática que se encarga de enviar tráfico a rutas desconocidas para el router, esta es la forma más sencilla de enrutamiento conectado a un único punto de salida y se debe indicar como red de destino 0.0.0.0/0.0.0.0. (GuilleSQL, 2007, p. 2)

La configuración del protocolo de enrutamiento estático lo podemos observar en el ejemplo de la figura 1-1:

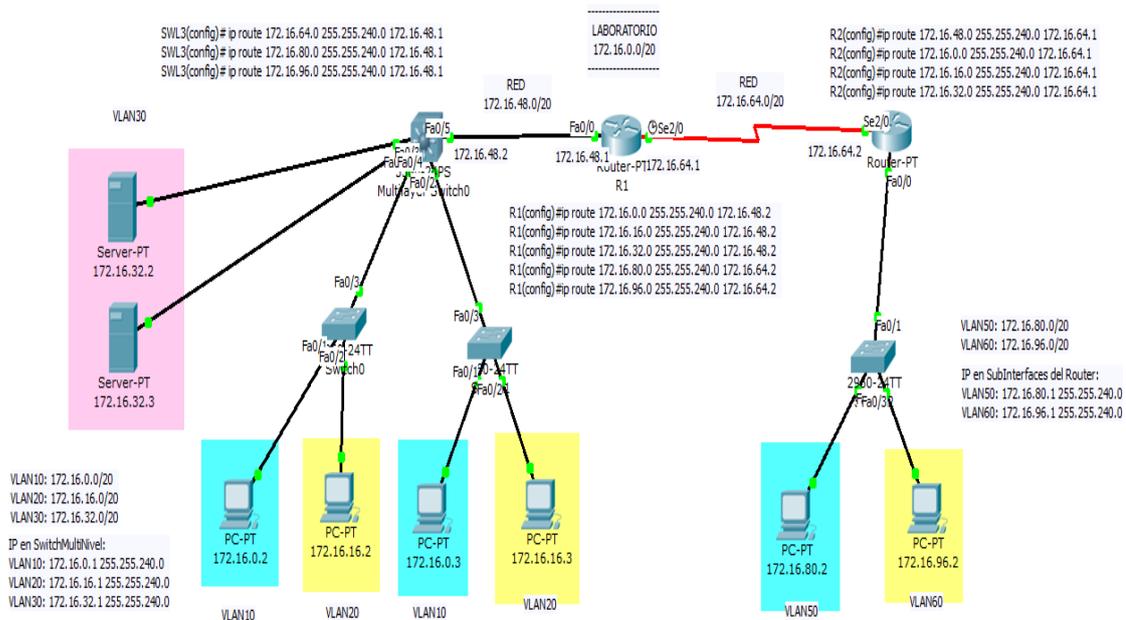


Figura 1-1: Enrutamiento estático configuración

Fuente: <http://blog.juliopari.com/>

Una de las grandes ventajas que tiene este tipo de enrutamiento es el bajo consumo del CPU, también posee una programación muy sencilla y por esta razón es más factible de entender para el Administrador de la red.

Aunque también posee grandes debilidades como la vulnerabilidad frente a errores, y siempre necesita del administrador para actualizar la información que cambia en la red ya que no se adecua a redes en crecimiento.

1.2.2. Enrutamiento dinámico

Este tipo de enrutamiento es usado desde finales de los años 80, el primer protocolo en aparecer fue RIP, sin embargo a medida que las redes iban creciendo este protocolo tuvo que adaptarse a este crecimiento y evoluciono a RIPv2, lastimosamente aun con la nueva versión este protocolo no puede transmitir información que supere 15 saltos de rutas y además no es escalable, es decir no puede ajustarse a un crecimiento de la red sin perder la calidad del servicio. (Ana, 2015, p. 1)

Las redes empezaron a crecer en forma exponencial y para poder cumplir esta necesidad se desarrollaron nuevos protocolos como es OSPF (Protocolo abierto de la ruta más corta), también

se desarrolló IS-IS, y la compañía en redes CISCO desarrollo su propio protocolo de gateway interior IGRP, este fue mejorado convirtiéndose en el protocolo EIGRP.

Al poseer redes de internet y para poder hacer enrutamiento se vio en la necesidad de crear rutas fronterizas, como es el caso de BGP que se usa entre proveedores de servicios y sus clientes en el momento de intercambiar información. (Ana, 2015, p. 1)

Una de las grandes ventajas de este tipo de enrutamiento es que es fácilmente adaptable a los crecimientos de la red además de ser muy usado en redes con vario routers interconectados ya que es independiente de tamaño de la red. (Ana, 2015, p. 2)

Aunque estas ventajas consumen varios recursos como la CPU y la RAM también tiene un alto grado de vulnerabilidad por lo que es recomendable usar parámetros adicionales para brindar seguridad al sistema, haciendo su programación mucho más compleja.

1.3. Sistemas Autónomos

Realizando un análisis a los casos de enrutamiento mencionados anteriormente, y en concordancia con el autor (Kurose, 2013, p. 380) en su libro “Computer Networking”, hemos visto la red simplemente como una colección de enrutadores interconectados. Un enrutador era indistinguible de otro en el sentido de que todos los enrutadores ejecutaban el mismo algoritmo de enrutamiento para calcular rutas de enrutamiento a través de toda la red.

Si nos damos cuenta que un conjunto homogéneo de enrutadores estableciendo el mismo protocolo de enrutamiento es un análisis demasiado simplista, por ejemplo el internet tiene extensos números de host trabajando simultáneamente, al almacenar la información de los enrutadores utilizaríamos demasiada memoria en ese proceso, esto ocasionaría que no existiría demasiado ancho de banda para poder enviar paquetes de datos.

Es muy importante también tener en cuenta aspectos fundamentales como el deseo de una empresa al ejecutar un algoritmo de enrutamiento a su elección, ocultar las características de una organización interna desde el exterior de la red.

Estos problemas pueden resolverse organizando enrutadores en sistemas autónomos (AS), con cada AS formado por un grupo de enrutadores que normalmente están bajo el mismo control administrativo (por ejemplo, operado por el mismo ISP o perteneciente a la misma red de la empresa). Los enrutadores dentro del mismo AS ejecutan el mismo algoritmo de enrutamiento y tienen información acerca de cada uno. (Kurose, 2013, p. 381)

El protocolo de enrutamiento que está corriendo dentro de un sistema autónomo es denominado “Protocolo de enrutamiento del sistema Intra-autonómico”, y es necesario conectar Sistemas Autónomos entre sí, y así uno o más de los enrutadores en un sistema autónomo tendrá la tarea adicional de ser responsable de reenviar paquetes a destinos fuera de este, estos enrutadores se llaman enrutadores de puerta de enlace. (Kurose, 2013, p. 382)

En la figura 2-1 detallamos un ejemplo de tres sistemas autónomos conectados entre sí, donde podemos observar los enrutadores de puerta de enlace, hay que tener en cuenta que en los sistemas autónomos diferentes no necesariamente están configurados el mismo protocolo de enrutamiento.

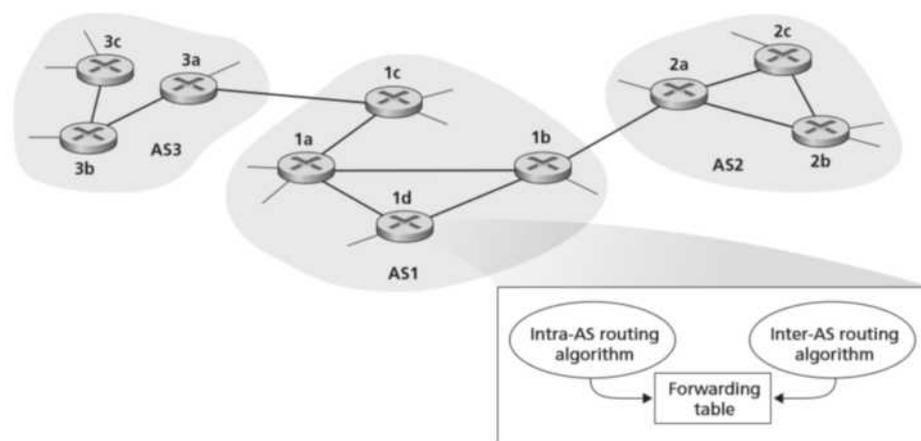


Figura 2-1: Interconexión de Sistemas Autónomos

Fuente: (J. Kurose 2001)

1.3.1. BGP

BGP es extremadamente complejo; libros enteros se han dedicado al tema y muchos problemas todavía no se entienden bien. (Yanuzzi, 2005, p. 55). No obstante, BGP es un protocolo

sumamente útil en el campo del internet, se puede decir que es el protocolo que lo combina todo y por esta razón es necesario entender cómo funciona en la RED, utilizando la figura 3-1 como ejemplo:

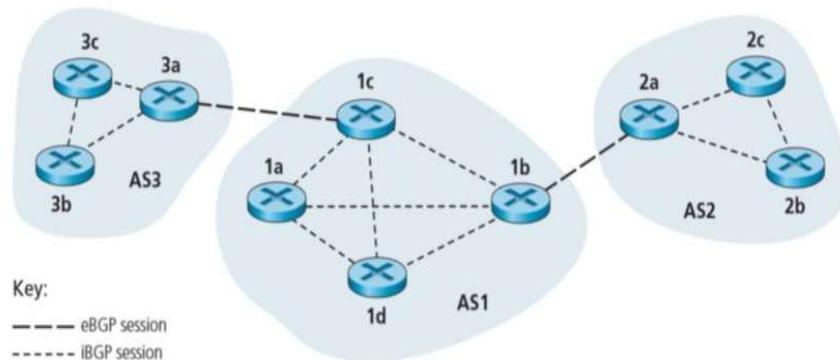


Figura 3-1: Conexión BGP

Fuente: (J. Kurose 2001)

BGP Permite a cada AS aprender qué destinos son accesibles a través de sus AS vecinos. En BGP, los destinos no son hosts sino prefijos sumariados, cada uno de los cuales representa una subred o una colección de subredes. (Kurose, 2013, p. 391).

En la figura 4-1 podemos apreciar una resumida clasificación de los diferentes protocolos de enrutamiento dinámico.

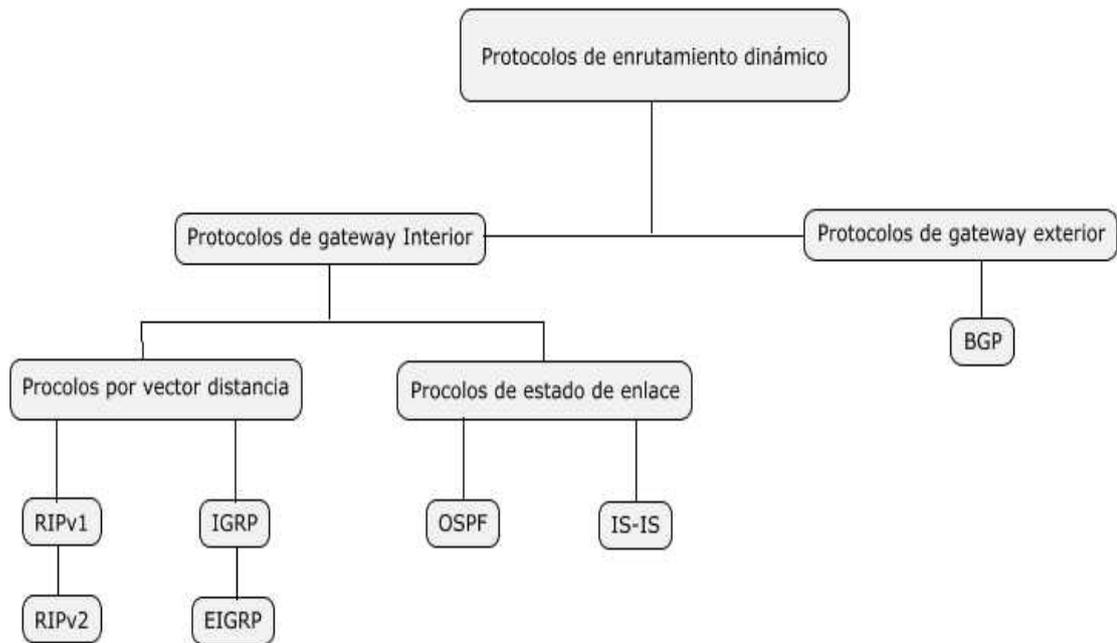


Figura 4-1: Clasificación Protocolos de Enrutamiento Dinámico

Realizado por: Laguna Andrés, Lugmania Juan, 2017

1.4. Plataformas de enrutamiento

Una plataforma de enrutamiento son programas en los cuales podemos implementar las capacidades de un enrutador y programar diferentes protocolos de la red, en estos programas se pueden implementar routers y firewalls (propios de capa 3) con una buena eficiencia, incluyendo la posibilidad de configurarlos vía web. (Aguirre & Martín, 2017, p. 25).

En la actualidad existen dos tipos esenciales de plataformas de enrutamiento:

- Las plataformas de Enrutamiento con licencia de paga.
- Las plataformas de Enrutamiento OpenSource.

La clasificación de estas plataformas se detalla en la tabla 1-1 que se muestra a continuación:

Tabla 1-1: Clasificación Plataformas Enrutamiento

PLATAFORMAS DE ENRUTAMIENTO			
Sistemas Operativos	Basado en Linux	Gratis	Floppyfw, FREESCO, IPCop, IPFire, LEDE, libreCMC, OpenWrt, Zeroshell, VyOS.
		Propietario	DD-WRT FRITZ!OS, RouterOS, SmoothWall, Tomato, AirOS & EdgeOS, Vyatta, ExtremeXOS, Alliedware Plus.
	Basado en FreeBSD	Gratis	m0n0wall, BSD Router Project, pfSense.
		Partly proprietary	JunOS
	Propietario	Cisco IOS, ExtremeWare, NX-OS, TiMOS, VRP.	
Routing daemons	Babel, B.A.T.M.A.N, BIRD, GNU Zebra, OpenBGPD, OpenOSPF, Quagga, XORP.		
Other software	Cjdns, Captive portal, Neighbornode, Advanced Direct Connect, Mumble, Tor		

Realizado por: Laguna Andrés, Lugmania Juan, 2017

1.4.1. Plataformas de enrutamiento OpenSource

Hoy en día, con la ayuda del software libre se han implementado herramientas o aplicaciones utilizando diferentes metodologías para su creación, estas herramientas reemplazan procesos o funciones que desarrollan dispositivos en el área de las telecomunicaciones, como es el caso de los routers, que son los encargados de la interconexión de redes y enrutamiento de paquetes. (Rivera & Sosa, 2014, p. 20)

Las principales plataformas virtuales que cumplen con las funciones de enrutamiento y además puedan brindar al usuario oportunidad de hacer configuraciones y pruebas de funcionalidad e interoperabilidad según la necesidad del hardware son las siguientes:

- Router modular click (click modular router)
- Vyatta System
- Quagga Routing Software Suite

- XORP (eXtensible Open Router Platform)
- OpenWrt
- FRRouting

1.4.1.1. *Router modular click (click modular router)*

Click es una nueva arquitectura de para el diseño e implementación de enrutadores configurables y flexibles. Un router click se ensambla a partir de módulos de procesamiento de paquetes llamados elementos. Elementos individuales que implementan funciones de router simples como paquete clasificación, gestión de colas, programación, y la interconexión con los dispositivos de red. (Rivera & Sosa, 2014, p. 21)

1.4.1.2. *Vyatta System*

Es un sistema operativo de código abierto para los routers y firewalls basados en Debian GNU / Linux. Se ejecuta en hardware x86 estándar y ofrece varias funciones relacionadas con la red, tales como enrutamiento IPv4 e IPv6, protocolos de enrutamiento dinámico, filtrado de tráfico con estado, IPS y más. Sus funciones se configuran y gestionan a través de la interfaz de línea de comando similar a los dispositivos de red de Juniper TM. (Rivera & Sosa, 2014, p. 22)

En el año 2012 esta plataforma fue adquirida por Brocade Communications System, un año después su nombre cambio a Brocade Vyatta 5400 vRouter y su última versión comercial ya no se basó en código abierto, a pesar de eso en octubre de 2013 un grupo independiente de desarrolladores trabajó en la versión de código abierto denominándolo VyOS.

1.4.1.3. *Quagga Routing Software Suite*

Quagga es una suite de software de enrutamiento, que proporciona implementaciones de OSPFv2, OSPFv3, RIP v1 y v2, RIPng y BGP-4 para las plataformas Unix, especialmente FreeBSD, Linux, Solaris y NetBSD. Es un fork de GNU Zebra que fue desarrollado por Kunihiko Ishiguro. (Rivera & Sosa, 2014, p. 22)

1.4.1.4. *XORP (eXtensible Open Router Platform)*

Es la única plataforma de enrutamiento de código abierto extensible de la industria. XORP ofrece una plataforma con todas las funciones que implementa los protocolos de enrutamiento IPv4 e IPv6 y una plataforma unificada para configurarlos. Es la única plataforma de código abierto para ofrecer capacidad de multidifusión integrada. (Rivera & Sosa, 2014, p. 22)

1.4.1.5. *OpenWrt*

Un sistema OpenWRT es una plataforma abierta gratuita GNU/Linux y es usada para dispositivos integrados, además proporciona un sistema de archivos totalmente modificable con gestión de paquetes. (Kciuk, 2014, p. 1)

Esto lo libera de la selección y configuración de la aplicación proporcionada por el proveedor y le permite personalizar el dispositivo mediante el uso de paquetes para adaptarse a cualquier aplicación. Para el desarrollador, OpenWRT es el marco para construir una aplicación sin tener que construir un firmware completo a su alrededor; para los usuarios, esto significa la capacidad de personalización completa, para usar el dispositivo de formas nunca antes previstas. (OpenWRT, 2017, p. 1).

1.4.1.6. *FRRouting*

FRRouting (FRR) es un conjunto de protocolos de enrutamiento IP para plataformas Linux y Unix que incluye daemons de protocolo para BGP, IS-IS, LDP, OSPF, PIM y RIP. (Linux Foundation, 2017, p. 1)

La perfecta integración de FRR con las pilas de red IP Linux / Unix nativas lo hace aplicable a una amplia variedad de casos de uso, incluida la conexión hosts, máquinas virtuales, servicios de red, LAN conmutación y enrutamiento. (Linux Foundation, 2017, p. 1)

FRR tiene sus raíces en el proyecto Quagga. De hecho, fue iniciado por muchos antiguos desarrolladores de Quagga que combinaron sus esfuerzos para mejorar la pila de protocolos disponibles. (Linux Foundation, 2017, p. 1)

1.5. Tarjeta Madre (Mainboard)

La tarjeta madre es la parte principal de un computador ya que nos sirve de alojamiento de los demás componentes permitiendo que estos interactúen entre si y puedan realizar procesos, esta tarjeta madre se escoge según nuestras necesidades. (Galeon.com, 2007, p. 1)

Existe una amplia gama de tarjetas madres acorde a nuestros requerimientos los cuales consideraremos los puntos más importantes:

- Uno de los principales puntos a considerar es encontrar un mainboard con 4 puertos LAN como mínimo, ya que este mainboard va a funcionar como una plataforma de routing y es necesario la utilización de varios puertos Ethernet.
- También se debe tomar en cuenta el procesador y la memoria que la tarjeta tenga integradas para poder tener una transición eficaz de paquetes.
- Además hay que tener en consideración la lista de sistemas operativos que este hardware pueda soportar, especialmente debe poder soportar una plataforma Linux, porque vamos a trabajar con un software Open Source.
- Finalmente uno de los puntos fuertes de estas consideraciones es el costo, pues al tener un costo elevado la plataforma se equipararía a un equipo de gama alta.

Para un análisis más detallado hemos clasificado las tarjetas madres de los principales fabricantes especificando las características que posee cada tarjeta, resaltando los puntos esenciales mencionados anteriormente, desde sus aspectos técnicos hasta sus aspectos económicos los cuales ilustraremos en la tabla 2-1 que presentamos a continuación:

Tabla 2-1: Clasificación de tarjetas madre

Modelo	Q190G4 4 LAN Mini PC	Jetway NF533 3.5" SBC Motherboard	Firewall Micro Appliance
Procesador	Intel Celeron Processor J1900 (Quad-Core 2M Cache, 2 GHz, up to 2.41 GHz)	Intel® Celeron J1900, 2.0GHz Quad-Core SoC, 10W, Bay Trail	Intel Atom E3845 Quad Core CPU @ 1.91GHz, 64 bit
Memoria	Support DDR3L 1333/1600 MHz, 1.35V 1 x DDR3 DIMM Memory Slot Max.	Onboard 4GB DDR3L-1333	Includes 8GB PC-1600 DDR3L RAM in a single SO-DIMM
LAN	4 x Intel I211-AT- 10/100/1000 Controller	4 x Intel I211-AT Gigabit LAN	4x Intel NIC ports
Almacenamiento	S01: only supports 1 * mSATA SSD	<ul style="list-style-type: none"> • 1 SATA2 3Gb/s Connector. • 1 Shared mSATA/mini-PCI Express Combo slot (Full-size) 	Includes 128GB mSATA SSD
Lista de Soporte de sistemas operativos	Q190G4 support Windows, Linux, pfSense, Sophos, VyOS, Linux iptables, Untangle, ect.	<ul style="list-style-type: none"> • WINDOWS: Windows 7, 8, 8.1, 10 32bit/64bit, Windows Embedded Standard 7,8 32bit/64bit. • LINUX: CentOS 6.G5, Debian 7.4, Fedora 20, openSUSE 13.1, Ubuntu 13.1, pfSense 2.2, RedHat 7.0, Android 4.4.2_r1-ia2. 	Most open source software firewall solutions, including pfSense, VyOS, OPNsense, OpenBSD, Security Onion, Untangle, CentOS, Linux, and Windows.
Connectors I/O	<ul style="list-style-type: none"> • 1 x Mini pcie port (only for WIFI, only for USB singal) • 1 x SIM Card slot • 1 x SATA power connector • 1 x DDR3L SO-DIMM Memory Slot 	<ul style="list-style-type: none"> • 1 +12V 2.5MM Diameter Locking DC Power Jack • 1 AT/ATX Mode jumper • 1 VGA Header (2560X1600 @ 60Hz) • 1 Serial Port (RS232) 	<ul style="list-style-type: none"> • 1x USB 2.0 port, 1x USB 3.0 port • VGA port • COM Port • VESA mount for mounting to a wall or the back of a monitor • Power supply included with US cord
Costo	US\$179.00	US\$279.80	US\$419.00

Realizado por: Laguna Andrés, Lugmania Juan, 2017

1.6. Software de verificación

Para analizar el correcto funcionamiento de la plataforma implementada hemos utilizado diferentes softwares de prueba los cuales son:

- Generadores de tráfico
- Analizadores de tráfico
- Generadores de rutas.

1.6.1. *Generadores de tráfico*

Un generador de tráfico es un programa muy útil en el área de los enrutadores, nos sirven para simular el tráfico de ciertas aplicaciones o servicios de red sin necesidad de tenerlos instalados o corriendo para probar ciertos parámetros de nuestra red como ancho de banda, memoria consumida por un tipo de servicio en un enrutador, etc.

Entre los generadores de tráfico más usados están:

- TREX Traffic Generator
- OSTINATO
- D-ITG
- IRIS
- PACKET PLAYER
- PACKET PLAYERPIPO E2H

1.6.1.1. *Ostinato*

Ostinato es un generador de paquetes, además de ser un analizador de tráfico de red con una GUI amigable. También una potente API de Python para la automatización de pruebas de red. Crea y

envía paquetes de varias secuencias con diferentes protocolos a diferentes velocidades. Piense en ello como "Wireshark in Reverse". Ostinato es un software de código abierto con licencia bajo GNU GPLv3 (Srivats, 2017, p. 1).

Ostinato tiene como objetivo, proporcionar un generador de tráfico y una herramienta de prueba de red para cada ingeniero de red y desarrollador, algo que hoy en día no es posible con los equipos de prueba de redes comerciales existentes. Con la herramienta adecuada, los desarrolladores e ingenieros de redes pueden hacer mejor su trabajo y mejorar la calidad de los productos de red. (Srivats, 2017, p. 1)

Sus principales características las encontramos en la página oficial de Ostinato (Srivats P 2017) las cuales son:

- Es muy útil para pruebas de carga de red y pruebas funcionales
- Crear y configurar múltiples transmisiones
- Configura las tasas de transmisión, ráfagas y número de paquetes
- Estadísticas, tasas de recepción, transmisión a nivel de interfaz para monitoreo y medición de red en tiempo real
- Emulación de dispositivo de red (ARP e ICMP) para múltiples hosts IP para ayudar en la simulación de tráfico de red
- Soporte para los protocolos estándar más comunes
- ARP, IPv4, IPv6, IP-in-IP aka IP Tunneling (6over4, 4over6, 4over4, 6over6)
- TCP, UDP, ICMPv4, ICMPv6, IGMP, MLD
- Cualquier protocolo basado en texto (HTTP, SIP, RTSP, NNTP, etc.)
- Establezca un valor para cualquier campo de cualquier protocolo
- Varíe los campos de paquetes en los paquetes en tiempo de ejecución, por ejemplo, cambiando las direcciones IP / MAC
- Apila protocolos en cualquier orden arbitraria para probar casos de error
- Hex Dump proporcionado por el usuario, puede especificar algunos o todos los bytes en un paquete

- El control exclusivo de un puerto para evitar que el sistema operativo envíe paquetes perdidos proporciona un entorno de prueba controlado (solo Windows)
- Captura tráfico de red (necesita Wireshark para ver los paquetes capturados)
- Multiplataforma: se puede ejecutar en Windows, Linux, BSD y Mac OS X
- Es un Software de código abierto

En la figura 5-1 podemos apreciar el interfaz de Ostinato.

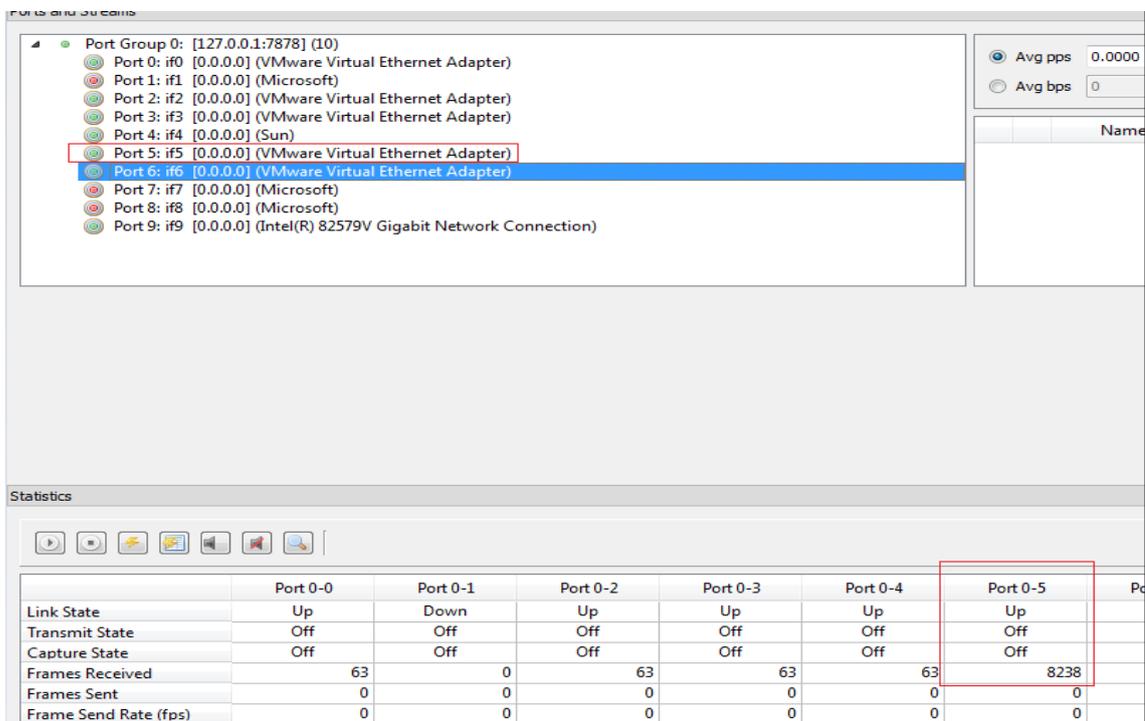


Figura 5-1: Interfaz De Ostinato

Fuente. <https://networkbackyard.files.wordpress.com/>

1.6.2. Analizadores de tráfico

Una de las principales tareas que debe realizar un administrador de red es la monitorización de tráfico, esta tarea es la que nos puede ayudar a determinar algún error que esté sucediendo en la comunicación de la red, o también gracias a este análisis podemos ver si existe algún tipo de tráfico no deseado, además de poder comprobar si los paquetes están llegando a los dispositivos finales sin inconvenientes.

En las redes corporativas este análisis debe incluir la salida hacia Internet, ya que ese suele ser el enlace con mayor carga de tráfico y por tanto el más complejo de gestionar. En las redes industriales, la complejidad radica en la cantidad de protocolos diferentes que se utilizan, por lo que es necesario interpretar múltiples tramas diferentes separándolas por tipo de protocolo. (INCIBE, 2010, p. 1)

1.6.2.1. *Tipos de analizadores*

Como nos explica el INSTITUTO NACIONAL DE CIBERSEGURIDAD DE ESPAÑA S.A. (INCIBE, 2010, p. 1) en su blog “Analizadores de red en sistemas de control”, Las herramientas utilizadas para llevar a cabo el análisis de red pueden ser tanto hardware como software.

- Analizadores mediante Software:

Los analizadores mediante software poseen análisis solo para protocolos extendidos y documentados, comúnmente son de herramienta de licencia libre.

Dependiendo de la herramienta será más o menos necesaria la intervención de un operador que la gestione. El analizador de red software más conocido tal vez sea Wireshark, pero existen otras muchas con funciones muy similares, como Windump o TCP Dump. (INCIBE, 2010, p. 1)

Para el caso de utilizar este tipo de analizadores hay que redirigir el tráfico que se va analizar y esto se lo puede realizar mediante dos métodos diferentes:

- Puerto Hub o concentrador: Se trata de conectar un concentrador en la red, y en uno de los puertos libres se sitúa el analizador. La cantidad de tráfico que se puede analizar es limitada. (INCIBE, 2010, p. 1)

- Puerto espejo: Se configura un puerto espejo en un Switch de la red, así el tráfico se redirija a dicho puerto, que será donde escuche el analizador de red. Todo el tráfico que pasa por el Switch será copiado y enviado al puerto espejo para que el analizador pueda monitorizar todo el tráfico. (INCIBE, 2010, p. 1)

Dos ejemplos de analizadores de tráfico los podemos observar en la figuras 6-1 y 7-1.

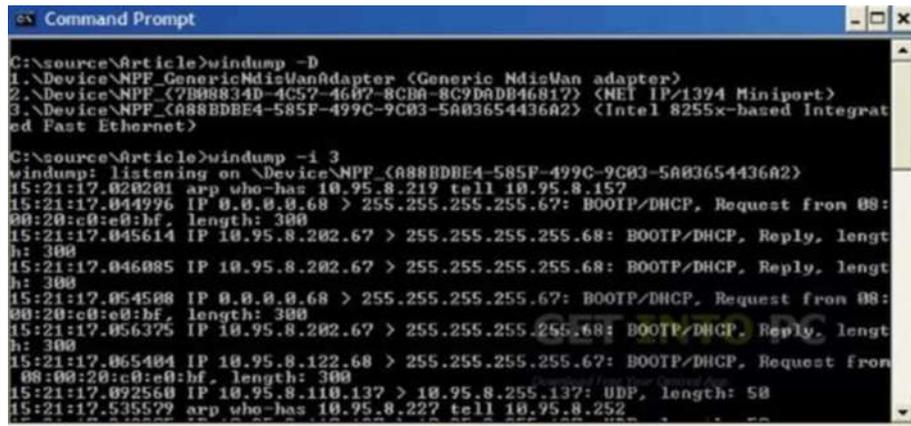


Figura 6-1: Analizador de tráfico mediante línea de comandos

Fuente. (INCIBE 2017)

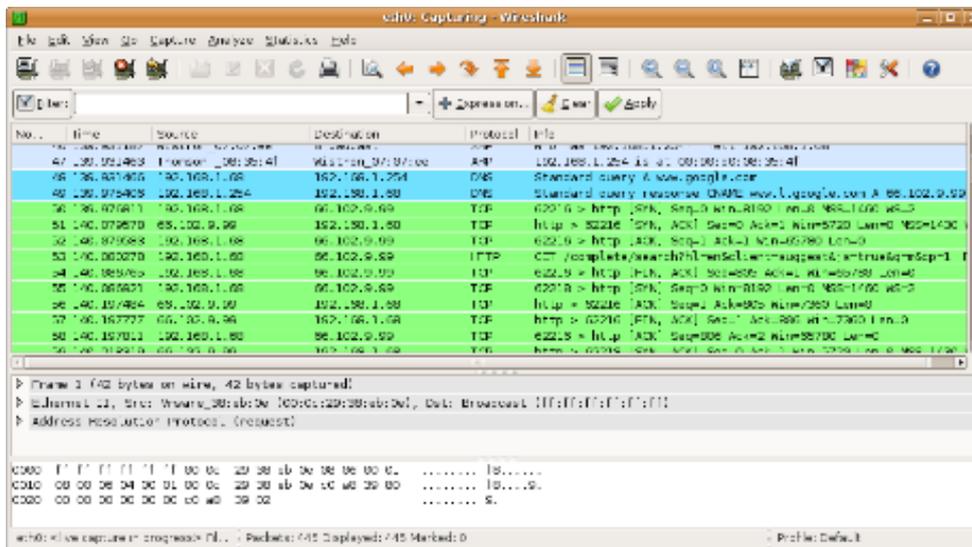


Figura 7-1: Analizador de tráfico mediante interfaz gráfica

Fuente. (INCIBE 2017)

Los equipos hardware para analizar tráfico son más orientados a protocolos específicos y su costo depende de la cantidad de protocolos que pueda detectar.

Así, mientras que en entornos corporativos suelen utilizarse los analizadores software, en los sistemas de control industrial suelen ser en su mayoría de tipo hardware. (INCIBE, 2010, p. 1). En la figura 8-1 se ilustra un analizador de tráfico de tipo hardware.

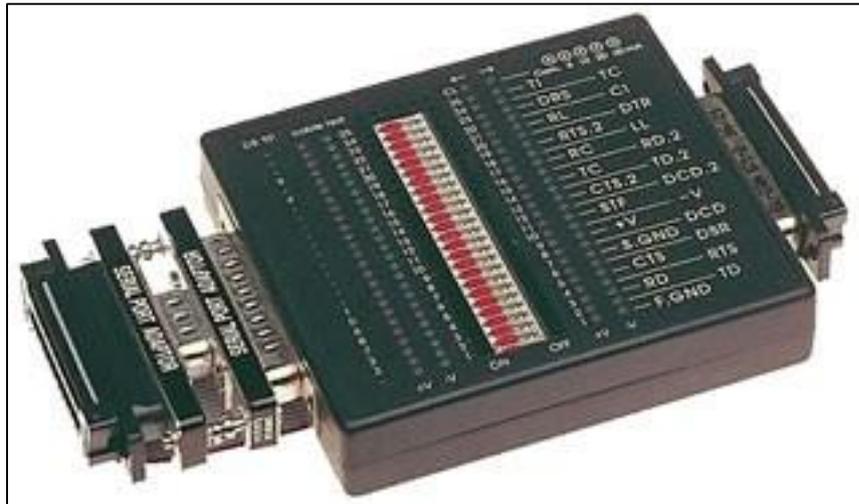


Figura 8-1: Analizador de tráfico de tipo hardware

Fuente. (INCIBE 2017)

1.6.2.2. *Wireshark*

Wireshark es el analizador de protocolo de red más importante y ampliamente utilizado en el mundo. Le permite ver lo que está sucediendo en su red a un nivel microscópico. El desarrollo de Wireshark prospera gracias a las contribuciones voluntarias de expertos en redes en todo el mundo y es la continuación de un proyecto iniciado por Gerald Combs en 1998. (Gerald Combs, 1998, p. 1)

En la página oficial de “Wireshark” (Gerald Combs, 1998, p. 1), podemos encontrarnos con las siguientes características:

- Inspección profunda de cientos de protocolos.
- Captura en vivo y análisis fuera de línea
- Navegador de paquetes estándar de tres paneles

1.6.3. *Generador de rutas vRIN*

VRIN es un dispositivo VM capaz de inyectar rutas en una red, construido en Debian 8 y Quagga. Este programa aun esta en desarrollo y razón por la cual aún posee ciertos errores en el momento de ejecutarse. (Dosztal, 2016, p. 1)

Este software genera 32 rutas IPv4 estáticas y las redistribuye en los protocolos de enrutamiento seleccionados; por lo tanto, las rutas generadas aparecen como externo (por ejemplo, "O E2"), vRIN ha sido probado con rutas de 1M BGP, al generarse los archivos de configuración demoraron 22 minutos; la VM usó 700MB de memoria (Slashdot, Media, 2017, p. 1)

Es capaz de soportar diversos protocolos de enrutamiento como: BGP, OSPF, RIP.

Como vRIN es una herramienta que aún se encuentra en desarrollo lastimosamente aún no existe demasiada información, actualmente se está trabajando en el desarrollo y mejoramiento de la herramienta para trabajar en Ipv6, en la figura 10-1 podemos apreciar el interfaz de vRIN.

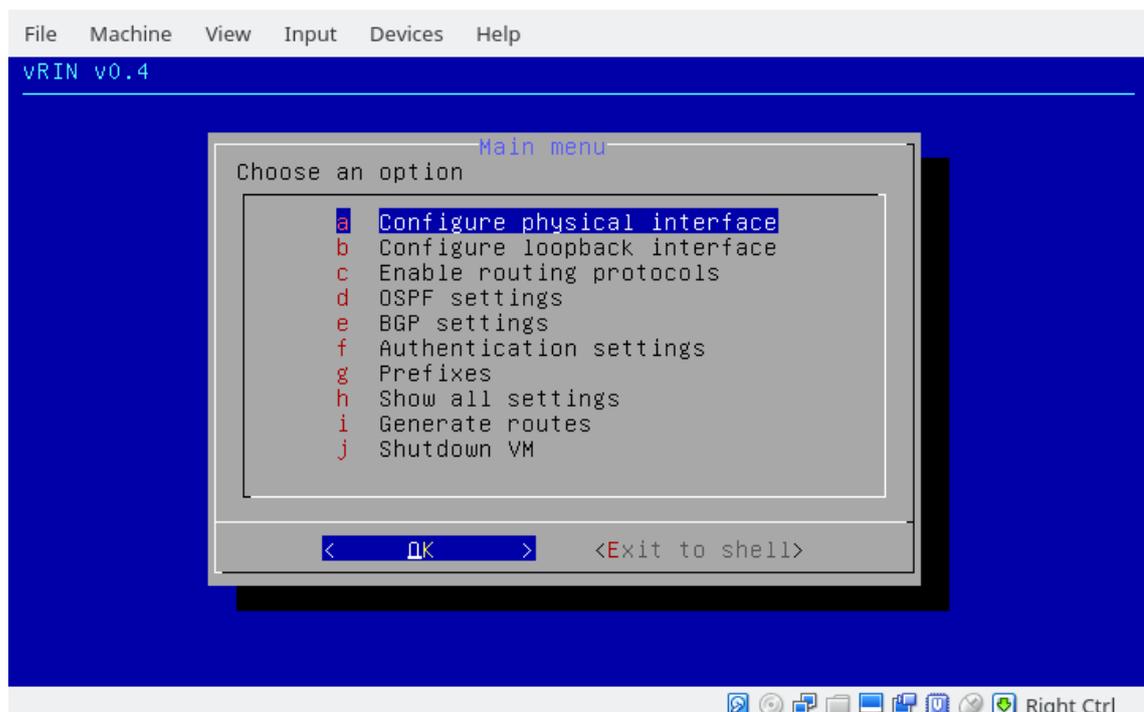


Figura 10-1: Interfaz de vRIN.

Fuente. <https://gns3.com/api/v2/assets/photo/5674a70d54a682a5a1a33b28/>

CAPÍTULO 2

MARCO METODOLÓGICO

2.1. Introducción

En este capítulo detalla todo lo relacionado a los procesos realizados para la investigación, obtención y elaboración de la plataforma routing Open Source, para lo cual se tuvo en consideración muchos aspectos, la mejor tecnología, materiales y equipos para obtener un funcionamiento eficaz, en lo cual se ha realizado un prototipo evaluando el rendimiento e interoperabilidad de dicha plataforma.

2.2. Parámetros a determinar

Como es de conocimiento los equipos de enrutamiento que actualmente se encuentran en el mercado tienen costos muy elevados, de tal forma tener una red que cuente con equipos de una misma marca es algo muy poco usual, igualmente sabiendo que la interoperabilidad entre equipos y sus diversos protocolos debe ser eficiente y eficaz, esta se volvió un gran dilema para los administradores de redes.

Otro parámetro que se ha tomado a consideración es el ciclo Deming también conocido como el ciclo PDCA (Plan, Do, Act, Check), para realizar diferentes medidas en el proceso de búsqueda, implementación y configuración de la plataforma Open Source y así facilitar una elaboración fácil de las actividades. Como se observa en el gráfico 1-2.

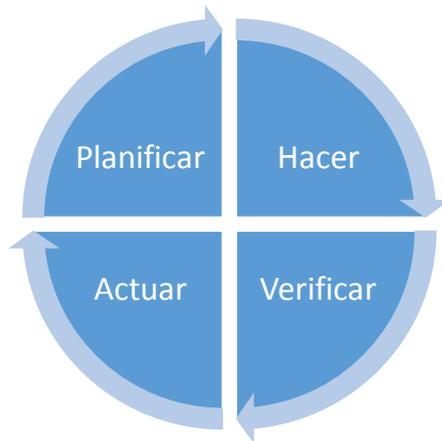


Gráfico 1-2: The Deming Cycle

Realizado por: Laguna Andrés, Lugmania Juan, 2017

Fuente: (Arveson 1998)

En el ciclo planear, hacer, verificar, actuar, se puede tomar un control eficiente de los parámetros a considerar para el desarrollo de la plataforma routing Open Source detallado a continuación en el gráfico 2-2.

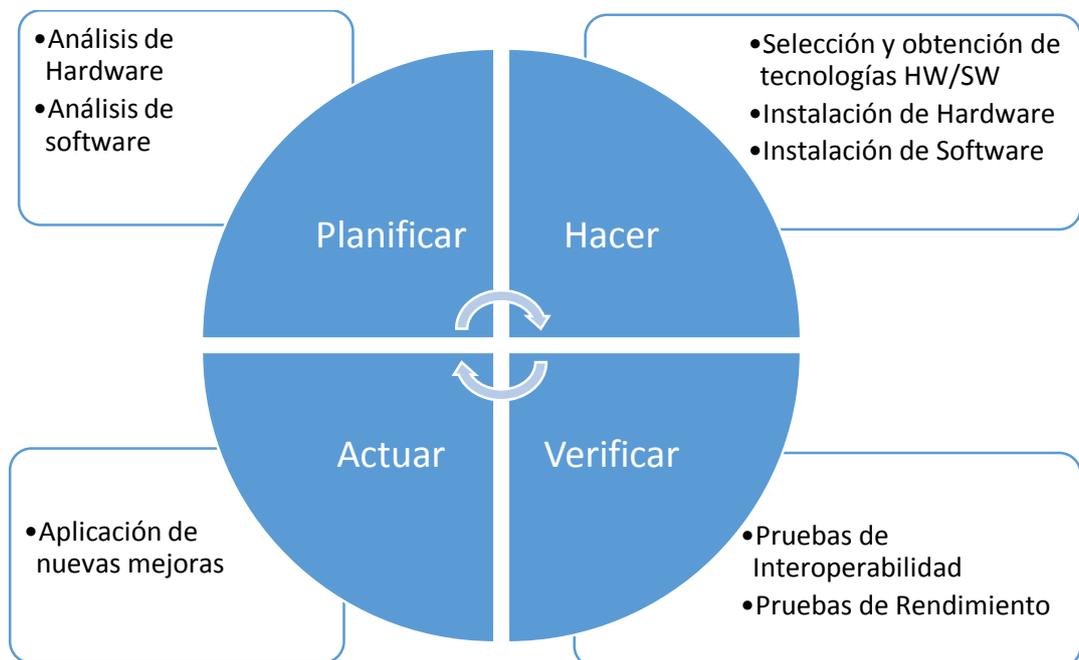


Gráfico 2-2: The Deming Cycle

Realizado por: Laguna Andrés, Lugmania Juan, 2017

Fuente: (Arveson 1998)

2.2.1. *Análisis de Hardware*

Partiendo desde el punto de vista técnico se ha podido observar diferentes tarjetas madres que existen en el mercado, muchos de ellos poseen diferentes características y una función específica, lo cual nos lleva a realizar un análisis costo beneficio para poder elegir una tarjeta adecuada para el desarrollo de este trabajo de titulación.

Un punto importante que se debe tomar en cuenta es que la tarjeta madre a utilizar debe tener como mínimo 4 puertos Ethernet para realizar la interconexión entre equipos.

Se ha tomado en consideración utilizar una tarjeta madre de placa pura o integra. Al tener esta tarjeta se obtendrá un menor costo realizar el ensamblado de los diferentes componentes a utilizar, además se pueden suscitar inconvenientes al momento de instalar los drivers de compatibilidad. Pero esto se puede realizar en los buses de expansión.

“Los buses de expansión son los encargados de transportar la información desde el chipset hasta los zócalos de expansión. El tipo y cantidad de buses y zócalos varía en cada modelo de MotherBoard. En equipos de gama baja a media no se suelen utilizar los dos o tres zócalos de expansión disponibles, ya que desde hace años las tarjetas madres incorporan las interfaces de uso más frecuentes: tarjeta de video, interfaz de audio, placa de red Ethernet, etc.” (Richarte, 2012, p. 40)

Por otro lado, al tener una tarjeta madre de placa integrada va aumentar el costo de adquisición, pero al mismo tiempo se va tener una garantía que su funcionamiento sea óptimo.

2.2.1.1. *Tipos de tarjetas madres afines*

Existe una amplia gama de tarjetas madres acorde a nuestros requerimientos mínimos los cuales consideraremos los puntos más importantes, como se muestra en la tabla 1-2.

Tabla 1-2: Requerimientos mínimos

Procesador	2 GHz
Memoria	4 GB
Numero de puertos LAN	4
Sistema Operativo	LINUX: CentOS, Fedora, Ubuntu

Realizado por: Laguna Andrés, Lugmania Juan, 2017

Tomando en cuenta todos los aspectos mencionados se ha procedido a la búsqueda del equipo Hardware con las características requeridas. Como se pudo evidenciar en el capítulo 1 en la tabla 2-1 sobre los modelos más accesibles, evaluaremos cuál de las tarjetas madres es la más apropiada.

Calificaremos de 0 a 1 siendo 0 el valor más bajo, y 1 el valor más alto para evaluar las necesidades, tomando en cuenta la tabla 1-2 si cumple con estos requerimientos igual o superior se calificara con “1” caso contrario con “0”, como se muestra a continuación en la tabla 2-2.

Tabla 2-2: Calificación final de requerimientos.

Modelo	Q190G4 4 LAN Mini PC	Jetway NF533 3.5" SBC Motherboard	Firewall Micro Appliance
Procesador	1	1	0
Memoria	1	1	1
LAN	1	1	1
Almacenamiento	1	1	1
Lista de Soporte de sistemas operativos	0	1	1
Connectors I/O	1	1	1
TOTAL	5	6	5

Realizado por: Laguna Andrés, Lugmania Juan, 2017

2.2.2. *Análisis de Software*

2.2.2.1. *Software de enrutamiento*

Tomando como punto de partida los diferentes tipos de plataformas Routing, muchas de ellas cuenta con características que se pueden aprovechar al máximo; siempre y cuando estas sean utilizadas adecuadamente. Por consiguiente se ha tomado en consideración los aspectos más importantes yendo a la par con los requerimientos mínimos. Como se muestra a continuación en la tabla 3-2

Tabla 3-2: Requerimientos necesarios

Open Source	Indispensable
Documentación	Indispensable
Adaptable	Indispensable
Actual	Indispensable

Realizado por: Laguna Andrés, Lugmania Juan, 2017

Una de las soluciones más eficientes al tener un Software de enrutamiento no compatible, es realizar la instalación previa de un Sistema Operativo GNU, en el cual se podrá realizar la instalación de los Daemons para su funcionamiento.

Los Sistemas operativos deben ser modo consola, Así no habrá gran consumo de recursos en la tarjeta madre. Calificaremos con 0 y 1 siendo 0 el valor más bajo, y 1 el valor más alto para evaluar los requerimientos necesarios, tomando en cuenta la tabla 1-2 si cumple con los requerimientos de forma igual o superior se calificara con “1” caso contrario con “0”, como se muestra a continuación en la tabla 4-2

Tabla 4-2: Calificación Routing Software

Datos	Quagga	FRRouting	VyOS	OpenWrt
Open Source	1	1	1	1
Documentación	1	1	1	0
Adaptable	1	1	0	1
Actual	0	1	0	1
TOTAL	3	4	2	3

Realizado por: Laguna Andrés, Lugmania Juan, 2017

Como se puede apreciar, se asignó calificaciones a cada una de las cualidades que las plataformas poseen y así se obtuvo la plataforma idónea para el presente trabajo.

Del mismo modo para el análisis se usará Wireshark uno de los programas más utilizados a nivel mundial para capturar tráfico de una red en tiempo real, con todas estas herramientas podremos simular un ambiente ideal y con todos los requerimientos que exige una red, para así poder analizar los resultados finales.

2.2.2.2. *Software de pruebas*

En la verificación se ha tomado a consideración los diferentes programas para inyectar tráfico siendo Trex y Ostinato los más utilizados hoy en día. Estos programas son utilizados para crear paquetes, generar de tráfico de red, también se utilizó vRIN para inyectar un gran número de rutas en la red.

Del mismo modo para el análisis se usará Wireshark uno de los programas más utilizados a nivel mundial para capturar tráfico de una red en tiempo real, con todas estas herramientas podremos simular un ambiente ideal y con todos los requerimientos que exige una red, para así poder analizar los resultados finales.

2.2.3. Selección y obtención de tecnologías HW/SW

2.2.3.1. Selección y obtención de tecnología HW

Luego de haber realizado el análisis de las diferentes tarjetas y sus diversas características para poder realizar el presente trabajo de titulación como se muestra en la tabla 4-2, la tarjeta que obtuvo el mayor peso fue la JETWAY NF533.

2.2.3.2. Selección y obtención de tecnologías SW

De la misma forma luego de haber hecho el mismo análisis del hardware para la selección del software, la plataforma FRRouting obtuvo la mayor calificación según sus características, razón por la cual fue la plataforma usada en el presente trabajo.

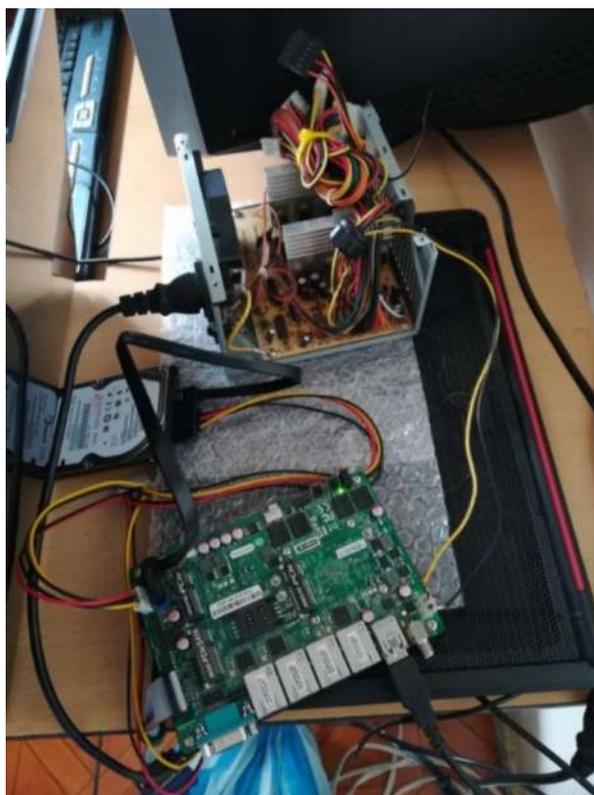
2.2.4. Instalación de Hardware

2.2.4.1. Conexión de componentes

La tarjeta madre JETWAY NF533 posee dos entradas para conectar el voltaje los cuales son:

- +12V 2.5MM Diameter Locking DC Power Jack
- AT/ATX Mode jumper

Para la suministración de voltaje de la tarjeta se decidió utilizar la entrada AT/ATX Mode jumper, brindando una fuente de alimentación constante y lo más estable posible, se utilizó una fuente AT usada en los CPU's de escritorio. Como se muestra a continuación en la fotografía 1-2



Fotografía 1-2: Conexión de la tarjeta madre

Jetway con fuente AT

Realizado por: Laguna Andrés, Lugmania Juan, 2017

2.2.4.2. *Ensamblaje de componentes*

Después de haber realizado un diseño tentativo del modelo del chasis del dispositivo, se recurrió a fabricación de un armazón metálico a medida el cual fue fabricado por el Sr. Cristian Montaluisa Ingeniero Mecatrónico de la Universidad de las Fuerzas Armadas ESPE, el cual mediante su cortadora de plasma CNC supo realizar bajo medidas exactas la fabricación y colocación de dicho armazón para nuestro dispositivo de enrutamiento. Como se muestra a continuación en las fotografías 2-2 y 3-2



Fotografía 2-2: Fabricación del armazón

Realizado por: Laguna Andrés, Lugmania Juan, 2017



Fotografía 3-2: Instalación de componentes

Realizado por: Laguna Andrés, Lugmania Juan, 2017

2.2.5. Instalación de Software

2.2.5.1. Instalación de Ubuntu Server 16.04.3 LTS

Una vez finalizado el ensamblaje del dispositivo se procedió a la instalación del sistema operativo Ubuntu Server 16.04.3 LTS, sistema operativo que se instaló de una forma muy similar a las instalaciones en CPU's común y corrientes

2.2.5.2. Instalación FRRouting

Ya instalado el sistema operativo UBUNTU y verificando que este funcione correctamente se instaló FRRouting siguiendo una serie de pasos y comandos los cuales se ilustra en la tabla 5-2.

Tabla 5-2: Pasos de Instalación

Orden	Descripción	Comando
1	Descargar FRR de repositorio oficial	<pre>\$ cd ~/ \$ git clone https://github.com/freerangerouting/frr.git \$ cd frr/ \$ git checkout -b stable/3.0 origin/stable/3.0</pre>
2	Instalar los requerimientos necesarios	<pre>\$ apt-get install git autoconf automake libtool make gawk libreadline-dev \ texinfo dejagnu pkg-config libpam0g-dev libjson-c-dev bison flex \ python-pytest libc-ares-dev python3-dev libsystemd-dev</pre>
3	Crear e instalar	<pre>\$/bootstrap.sh \$/configure \ --prefix=/usr \ --localstatedir=/var/run/frr \ --sbindir=/usr/sbin \</pre>

		<pre> --sysconfdir=/etc/frr \ --enable-multipath=64 \ --enable-user=frr \ --enable-group=frr \ --enable-vty-group=frrvty \ --enable-configfile-mask=0640 \ --enable-logfile-mask=0640 \ --enable-systemd=yes \$ make \$ sudo make install </pre>
4	Añadir al usuario y al grupo FRR	<pre> \$ sudo groupadd -g 92 frr \$ sudo groupadd -r -g 85 frrvty \$ sudo adduser --system --ingroup frr --home /var/run/frr/ \ --gecos "FRR suite" --shell /sbin/nologin frr \$ sudo usermod -a -G frrvty frr </pre>
5	Crear los archivos de configuración FRR vacíos y los directorios necesarios:	<pre> \$ sudo install -m 755 -o frr -g frr -d /var/log/frr \$ sudo install -m 775 -o frr -g frrvty -d /etc/frr \$ sudo install -m 640 -o frr -g frr /dev/null /etc/frr/zebra.conf \$ sudo install -m 640 -o frr -g frr /dev/null /etc/frr/bgpd.conf \$ sudo install -m 640 -o frr -g frr /dev/null /etc/frr/ospfd.conf \$ sudo install -m 640 -o frr -g frr /dev/null /etc/frr/ospf6d.conf \$ sudo install -m 640 -o frr -g frr /dev/null /etc/frr/isisd.conf \$ sudo install -m 640 -o frr -g frr /dev/null /etc/frr/ripd.conf \$ sudo install -m 640 -o frr -g frr /dev/null /etc/frr/ripngd.conf \$ sudo install -m 640 -o frr -g frr /dev/null /etc/frr/pimd.conf \$ sudo install -m 640 -o frr -g frr /dev/null /etc/frr/ldpd.conf </pre>

		<pre>\$ sudo install -m 640 -o frr -g frr /dev/null /etc/frr/nhrpd.conf \$ sudo install -m 640 -o frr -g frr /dev/null /etc/frr/vtysh.conf \$ sudo mkdir /var/run/frr \$ sudo chown -R frr:frr /var/run/frr/</pre>
6	<p>Editar /etc/sysctl.conf y quitar los comentarios para habilitar IPv4 & IPv6</p>	<pre># Uncomment the next line to enable packet forwarding for IPv4 net.ipv4.ip_forward=1 # Uncomment the next line to enable packet forwarding for IPv6 # Enabling this option disables Stateless Address Autoconfiguration # based on Router Advertisements for this host net.ipv6.conf.all.forwarding=1</pre>
7	<p>Instalar el systemd service</p>	<pre>sudo install -m 644 tools/frr.service /etc/systemd/system/frr.service sudo install -m 644 tools/etc/default/frr /etc/default/frr sudo install -m 644 tools/etc/frr/daemons /etc/frr/daemons sudo install -m 644 tools/etc/frr/daemons.conf /etc/frr/daemons.conf sudo install -m 644 tools/etc/frr/frr.conf /etc/frr/frr.conf sudo install -m 644 -o frr -g frr tools/etc/frr/vtysh.conf /etc/frr/vtysh.conf</pre>
8	<p>Habilitar los daemons editando /etc/frr/daemons y cambia los valores de NO a YES por ejemplo:</p>	<pre>zebra=yes bgpd=yes ospfd=yes ospf6d=yes ripd=yes ripngd=yes isisd=yes</pre>
9	<p>Habilitar el systemd service</p>	<pre>systemctl enable frr</pre>

10	Iniciar el systemd service	systemctl start fir systemctl status fir
11	Para poder ingresar al FRR debes ingresar como usuario y poner el siguiente comando	Vtysh

Realizado por: Laguna Andrés, Lugmania Juan, 2017

Una vez ingresado el comando “vtysh” como se muestra en el paso 11 de la tabla 5-2 se puede proceder a realizar la configuración de los diferentes protocolos.

2.2.6. Pruebas de Interoperabilidad

Para realizar las comprobaciones de interoperabilidad, se hicieron diversas pruebas basadas en un escenario en común, utilizando dispositivos de enrutamiento marca Cisco y Mikrotik. Los resultados se podrán apreciar en el Capítulo 3, el escenario utilizado es el que se muestra en la fotografía 1-2.

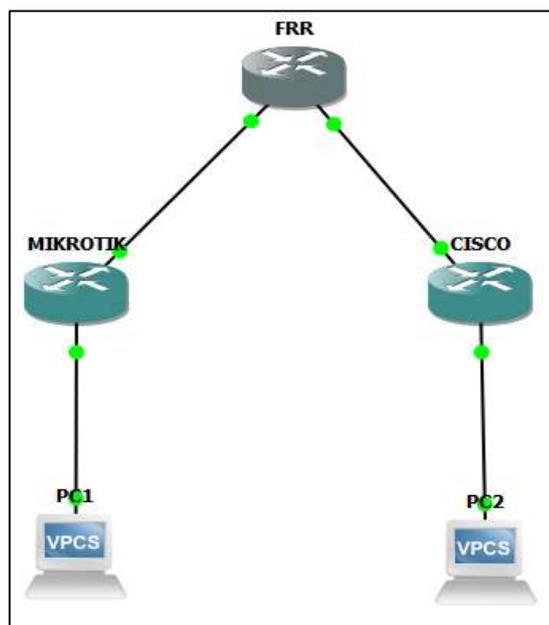


Figura 1-2: Escenario protocolo OSPF

Realizado por: Laguna Andrés, Lugmania Juan, 2017

Como podemos observar en la figura mostrada anteriormente, los equipos usados son de diferentes fabricantes, además en este escenario basándonos en los conceptos de enrutamiento y para comprobar una correcta interoperabilidad hicimos cuatro pruebas de conectividad los cuales detallamos a continuación:

- OSPF en IPv4
- OSPF y RIPv2 aplicando redistribución de rutas en IPv4
- OSPF en IPv6
- OSPF y RIPv6 aplicando redistribución de rutas en IPv6

2.2.6.1. Prueba 1: OSPF en IPv4

En primera instancia se levanta el protocolo OSPF en toda la red para la compartición de información en los routers Mikrotik, cisco y Jetway. Como se muestra en la figura 2-2.

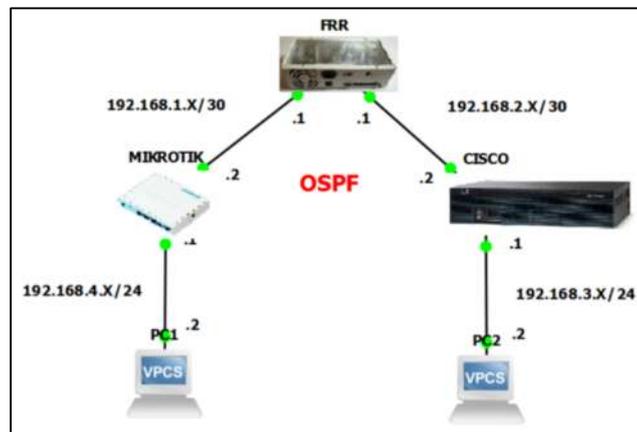


Figura 2-2: Escenario OSPF en IPv4

Realizado por: Laguna Andrés, Lugmania Juan, 2017

Realizando pruebas de conectividad mediante el comando “ping”, y conjuntamente analizando las rutas aplicando el comando “tracert” en los Host al igual que en los dispositivos de enrutamiento, se verifico la correcta interoperabilidad de los enrutadores cuyos resultados se detallan en el capítulo 3

2.2.6.2. Prueba 2: OSPF y RIPv2 aplicando redistribución de rutas en IPv4

De modo similar, se levanta el protocolo RIP & OSPF para la compartición de información en los routers Mikrotik, Cisco y Jetway. Los cuales se encuentran conectados de la siguiente manera como se muestra en la figura 3-2.

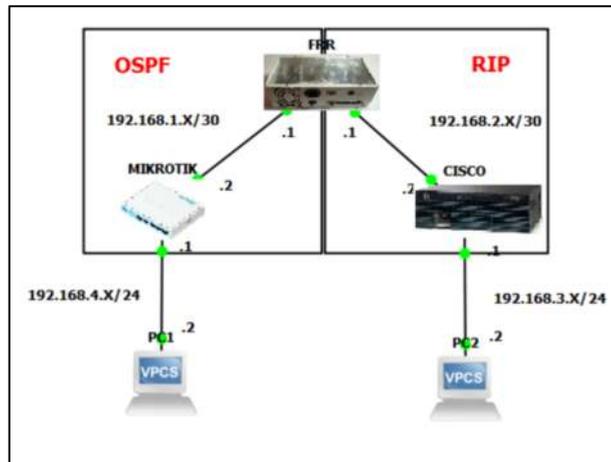


Figura 3-2: Escenario redistribución OSPF & RIPv2 en IPv4

Realizado por: Laguna Andrés, Lugmania Juan, 2017

De manera similar a la Prueba 1 se realizó pruebas de conectividad con el comando “ping” y se analizó los saltos en las rutas que da un paquete de información mediante “tracert”.

2.2.6.3. Prueba 3: OSPF en IPv6

En primera instancia se levantó el protocolo OSPF en toda la red con Protocolo de internet IPv6 y mediante los comandos mencionados anteriormente se procedió a la verificación de la interoperabilidad, para la realización de la prueba nos guiamos en el escenario mostrado a en la figura 4-2.

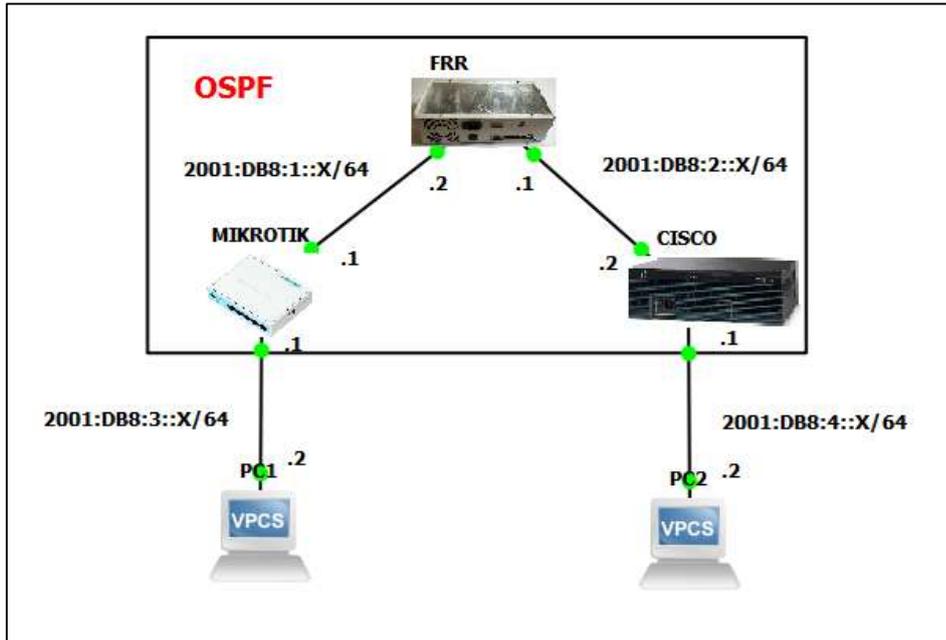


Figura 4-2: Escenario OSPF en IPv6

Realizado por: Laguna Andrés, Lugmania Juan, 2017

La correcta interoperabilidad de este escenario fue verificada al igual que las pruebas 1 y 2 con los comandos “ping” y “tracert” y sus resultados serán detallados más a fondo en el capítulo 3.

2.2.6.4. Prueba 4: OSPF y RIPng aplicando redistribución de rutas en IPv6

Esta prueba uso la misma configuración que el apartado 2.2.6.2. Pero se configuró el protocolo IPv6 como se muestra en la figura 5-2.

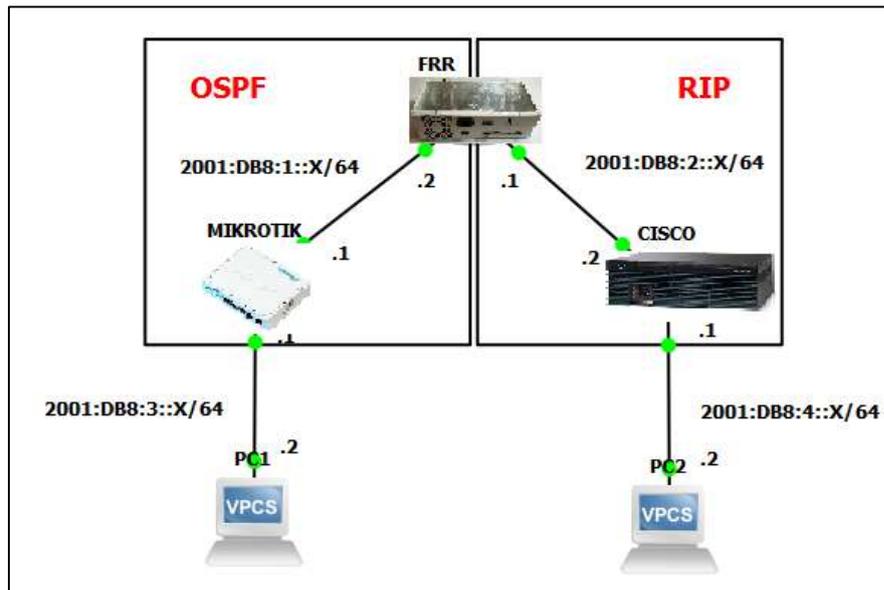


Figura 5-2: Escenario redistribución OSPF & RIP en IPv6

Realizado por: Laguna Andrés, Lugmania Juan, 2017

Las pruebas de interoperabilidad fueron realizadas de manera similar a la prueba 3 y sus resultados serán detallados en el capítulo 3.

2.2.7. Pruebas de Rendimiento

Las pruebas de rendimiento fueron basadas en dos aspectos, el rendimiento durante la generación de tráfico y el rendimiento ante la generación de rutas, para ello se realizaron tres tipos de pruebas que son:

- Generación de tráfico con Ostinato en IPv4.
- Generación de tráfico con Ostinato en Ipv6.
- Generación de rutas BGP con vRIN en IPv4.

Para realizar la prueba de generación de tráfico se usó una comparativa entre dos escenarios, en un escenario se conectó nuestro dispositivo de enrutamiento, y en el otro escenario se desarrolló una conexión de equipos marca CISCO como se muestra en la figura 6-2

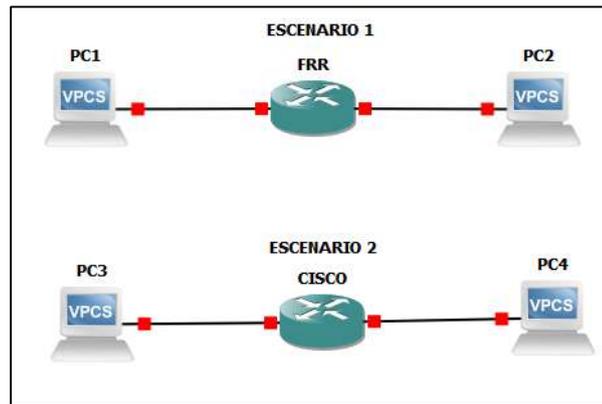


Figura 6-2: Escenarios para prueba de rendimiento mediante generación de tráfico

Realizado por: Laguna Andrés, Lugmania Juan, 2017

En la prueba de rendimiento mediante generación de rutas la comparativa se hizo conectando directamente el programa vRIN hacia la plataforma de enrutamiento FRRouting versus la conexión de vRIN conectado a un equipo Cisco, en estas pruebas se generó tanto 120 como 200 rutas en cada escenario propuesto como se ilustra en la figura 7-2:



Figura 7-2: Escenarios para prueba de rendimiento mediante generación de rutas

Realizado por: Laguna Andrés, Lugmania Juan, 2017

Para esta prueba hay que tomar en consideración que el programa vRIN es un sistema en desarrollo, y aún no se ha comprobado completamente su funcionamiento para el protocolo Ipv6, tras este motivo solo se realizó un escenario de prueba en IPv4, en el que se comparó directamente la plataforma de enrutamiento FRRouting frente a un enrutador marca Cisco, en esta prueba se inyectaron hasta 200 rutas para observar parámetros como consumo de memoria o tiempo en crear las rutas.

2.2.7.1. Prueba 5: Generación de tráfico con Ostinato en IPv4.

Usando Ostinato para generar tráfico inyectando 1'600.000 paquetes, y WireShark para analizar el rendimiento a dicho tráfico, pudimos analizar el comportamiento de cada una de las redes, para las pruebas se utilizó las mismas direcciones Ipv4 con el fin de que los escenarios tengan la mayor similitud en su configuración, como se puede apreciar en la figura 8-2

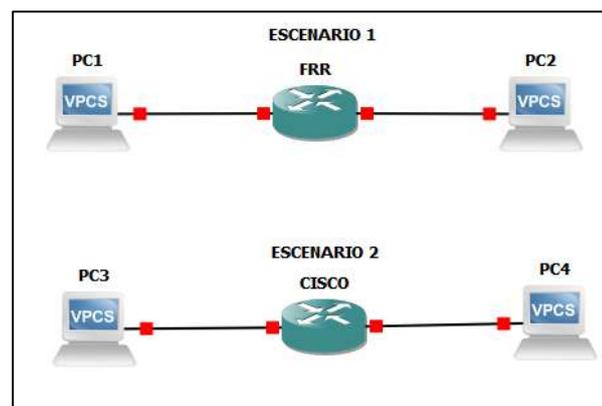


Figura 8-2: Escenarios para prueba de rendimiento en IPv4

Realizado por: Laguna Andrés, Lugmania Juan, 2017

2.2.7.2. Prueba 6: Generación de tráfico con Ostinato en Ipv6

Al igual que la prueba 5 se inyectaron entre 1'600.000 paquetes UDP, usando una comparativa con los mismos escenarios propuestos se levantó el protocolo de internet IPv6, y mediante Wireshark se analizó el comportamiento de la plataforma en este protocolo, su configuración se muestra en la figura 9-2.

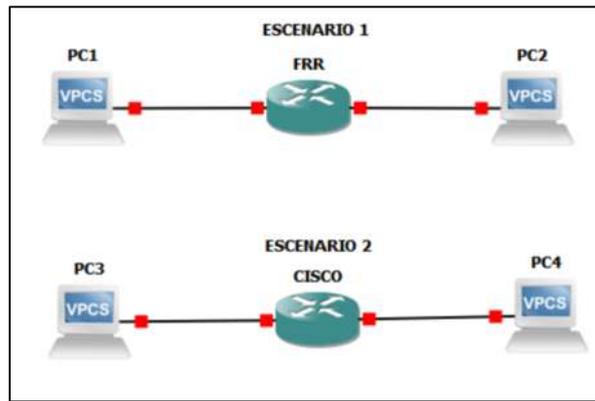


Figura 9-2: Escenarios para prueba de rendimiento en Ipv6

Realizado por: Laguna Andrés, Lugmania Juan, 2017

Los escenarios son similares con respecto a las configuraciones los resultados de estas pruebas se discutirán en el capítulo 3.

2.2.7.3. Prueba 7: Generación de rutas BGP con vRIN en IPv4

Utilizando los mismos escenarios planteados anteriormente e inyectando las mismas cantidades de rutas se hizo un análisis pero con un protocolo de enrutamiento de un sistema autónomo como BGP.

Como podemos observar en la figura 10-2 la configuración para la generación de rutas es muy parecida en ambos casos vRIN nos da la opción de escoger simplemente el protocolo que deseamos utilizar para generar dichas rutas, los resultados de estas pruebas están detalladas en el capítulo 3 de este escrito.

```

All settings
Physical interface IPs: 192.168.10.10 |
Physical interface masks: 24 | 64
Physical interface networks: 192.168.10.0 |
Loopback interface IPs: 2.2.2.2 | 2001:ad02::2
BGP: on
BGP local AS: 65000
BGP remote AS: 65001
BGP neighbor IPs: 192.168.10.1 |
OSPFv2: off
OSPFv3: off
RIP: off
RIPng: off
Auth type: None
First generated prefixes: 198.0.0.0 |
Number of prefixes: 200
< OK >

```

Figura 10-2: Ejemplo de generación de rutas con BGP

Realizado por: Laguna Andrés, Lugmania Juan, 2017

CAPÍTULO 3

MARCO DE RESULTADOS, ANÁLISIS Y DISCUSIÓN DE RESULTADOS

En el presente capítulo se describe el análisis de los resultados obtenidos después de la Evaluación de rendimiento e interoperabilidad de una plataforma HW/SW de routing OpenSource, a través de diversas pruebas con varios escenarios de interoperabilidad y rendimiento frente a fabricantes de dispositivos de redes conocidos como las empresas CISCO y MIKROTIK.

3.1. Análisis de resultados obtenidos

3.1.1. Pruebas de interoperabilidad

3.1.1.1. Prueba 1: OSPF en IPv4

Una vez levantado el escenario detallado en el capítulo dos, se procedió a realizar peticiones desde los dos host a varias direcciones de la red, nuestro análisis se basó en dos parámetros: Ping para poder comprobar conectividad y Traceroute para poder observar detalladamente el número de saltos que recorre el paquete, obteniendo los resultados mostrados en las figuras 1-3 y 2-3 mostradas a continuación:

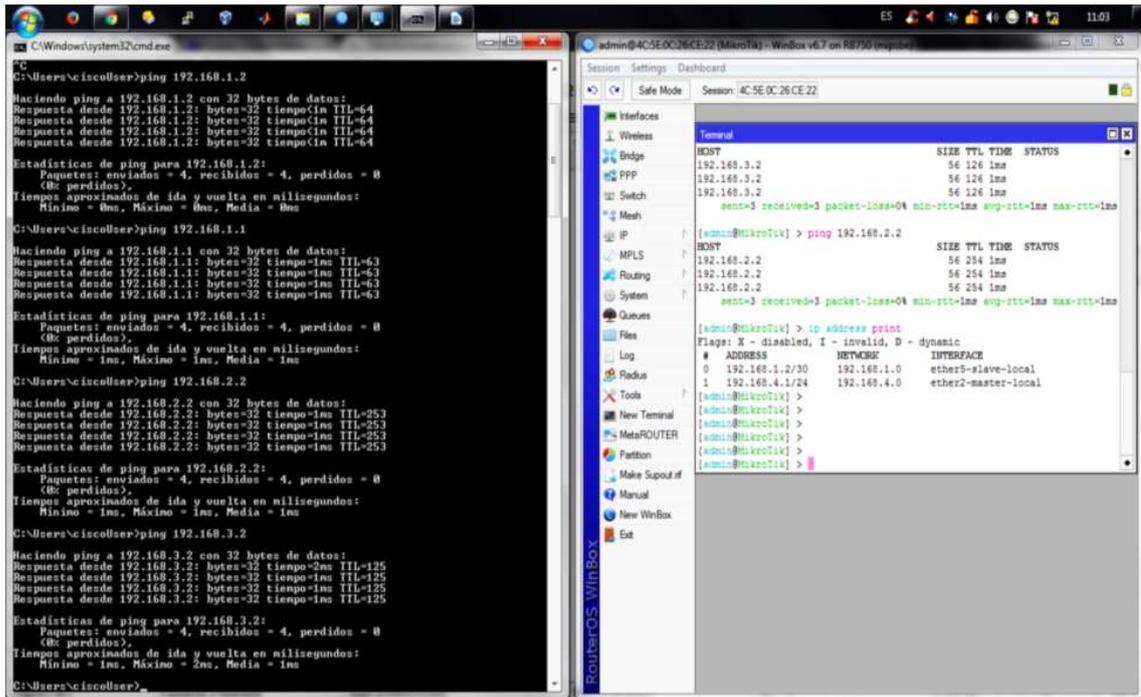


Figura 1-3: Ping desde PC1

Realizado por: Laguna Andrés, Lugmania Juan, 2017

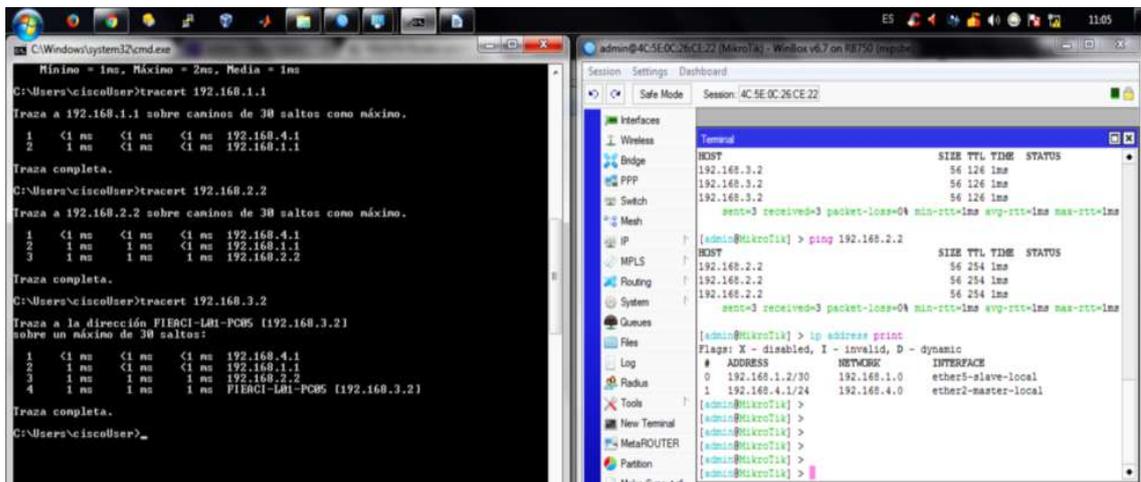


Figura 2-3: Traceroute desde PC1

Realizado por: Laguna Andrés, Lugmania Juan, 2017

De igual forma se comprobó que exista un ping exitoso desde el Pc2 conectado al router CISCO hacia el Pc1, analizando los mismos parámetros los resultados se muestran en las figuras 3-3 y 4-3 mostradas a continuación:

```

COM1 - PuTTY
redundancy

!
interface Embedded-Service-Engine0/0
no ip address
shutdown
!
interface GigabitEthernet0/0
ip address 192.168.2.2 255.255.255.252
duplex auto
speed auto
!
interface GigabitEthernet0/1
ip address 192.168.3.1 255.255.255.0
duplex auto
speed auto
!
interface GigabitEthernet0/2
no ip address
shutdown
duplex auto
speed auto
!
interface Serial0/0/0
no ip address
shutdown
clock rate 2000000
!
interface Serial0/0/1
no ip address
!

C:\Windows\system32\cmd.exe
C:\Users\nciscoUser>
C:\Users\nciscoUser>ping 192.168.3.1
Haciendo ping a 192.168.3.1 con 32 bytes de datos:
Respuesta desde 192.168.3.1: bytes=32 tiempo<1m TTL=255
Estadísticas de ping para 192.168.3.1:
Paquetes: enviados = 4, recibidos = 4, perdidos = 0
(0% perdidos).
Tiempo aproximado de ida y vuelta en milisegundos:
Mínimo = 0ms, Máximo = 0ms, Media = 0ms
C:\Users\nciscoUser>ping 192.168.2.1
Haciendo ping a 192.168.2.1 con 32 bytes de datos:
Respuesta desde 192.168.2.1: bytes=32 tiempo<1m TTL=63
Estadísticas de ping para 192.168.2.1:
Paquetes: enviados = 4, recibidos = 4, perdidos = 0
(0% perdidos).
Tiempo aproximado de ida y vuelta en milisegundos:
Mínimo = 0ms, Máximo = 0ms, Media = 0ms
C:\Users\nciscoUser>ping 192.168.1.2
Haciendo ping a 192.168.1.2 con 32 bytes de datos:
Respuesta desde 192.168.1.2: bytes=32 tiempo=1ms TTL=62
Estadísticas de ping para 192.168.1.2:
Paquetes: enviados = 4, recibidos = 4, perdidos = 0
(0% perdidos).
Tiempo aproximado de ida y vuelta en milisegundos:
Mínimo = 1ms, Máximo = 1ms, Media = 1ms
C:\Users\nciscoUser>ping 192.168.4.2
Haciendo ping a 192.168.4.2 con 32 bytes de datos:
Respuesta desde 192.168.4.2: bytes=32 tiempo=1ms TTL=125
Estadísticas de ping para 192.168.4.2:
Paquetes: enviados = 4, recibidos = 4, perdidos = 0
(0% perdidos).
Tiempo aproximado de ida y vuelta en milisegundos:
Mínimo = 1ms, Máximo = 1ms, Media = 1ms
C:\Users\nciscoUser>

```

Figura 3-3: Ping desde PC2

Realizado por: Laguna Andrés, Lugmania Juan, 2017

```

COM1 - PuTTY
redundancy

!
interface Embedded-Service-Engine0/0
no ip address
shutdown
!
interface GigabitEthernet0/0
ip address 192.168.2.2 255.255.255.252
duplex auto
speed auto
!
interface GigabitEthernet0/1
ip address 192.168.3.1 255.255.255.0
duplex auto
speed auto
!
interface GigabitEthernet0/2
no ip address
shutdown
duplex auto
speed auto
!
interface Serial0/0/0
no ip address
shutdown
clock rate 2000000
!
interface Serial0/0/1
no ip address
!

C:\Windows\system32\cmd.exe
C:\Users\nciscoUser>
C:\Users\nciscoUser>ping 192.168.4.2
Haciendo ping a 192.168.4.2 con 32 bytes de datos:
Respuesta desde 192.168.4.2: bytes=32 tiempo=1ms TTL=125
Estadísticas de ping para 192.168.4.2:
Paquetes: enviados = 4, recibidos = 4, perdidos = 0
(0% perdidos).
Tiempo aproximado de ida y vuelta en milisegundos:
Mínimo = 1ms, Máximo = 1ms, Media = 1ms
C:\Users\nciscoUser>tracert 192.168.3.1
Traza a 192.168.3.1 sobre camino de 30 saltos como máximo.
 1 <1 ms <1 ms <1 ms 192.168.3.1
Traza completa.
C:\Users\nciscoUser>tracert 192.168.2.1
Traza a 192.168.2.1 sobre camino de 30 saltos como máximo.
 1 <1 ms <1 ms <1 ms 192.168.3.1
 2 <1 ms <1 ms <1 ms 192.168.2.1
Traza completa.
C:\Users\nciscoUser>tracert 192.168.1.2
Traza a 192.168.1.2 sobre camino de 30 saltos como máximo.
 1 <1 ms <1 ms <1 ms 192.168.3.1
 2 <1 ms <1 ms <1 ms 192.168.2.1
 3 <1 ms <1 ms <1 ms 192.168.1.2
Traza completa.
C:\Users\nciscoUser>tracert 192.168.4.2
Traza a 192.168.4.2 sobre camino de 30 saltos como máximo.
 1 <1 ms <1 ms <1 ms 192.168.3.1
 2 <1 ms <1 ms <1 ms 192.168.2.1
 3 <1 ms <1 ms <1 ms 192.168.1.2
 4 <1 ms <1 ms <1 ms 192.168.4.2
Traza completa.
C:\Users\nciscoUser>

```

Figura 4-3: Traceroute desde PC2

Realizado por: Laguna Andrés, Lugmania Juan, 2017

Como podemos observar en las figuras la conectividad hacia las diversas direcciones ha sido exitoso tanto desde la Pc1 como la Pc2, brindando una comunicación exitosa de nuestra placa con los equipos routers de diversos fabricantes, además también podemos observar mediante el comando “Traceroute” los saltos que realiza el paquete desde su origen hacia su destino.

Para un análisis más a fondo mostraremos también las peticiones realizadas por nuestra plataforma a las diversas direcciones de la red, estas peticiones se detallan a continuación en las fotografías 1-3 y 2-3:

```
end
Jetway# ping 192.168.2.2
PING 192.168.2.2 (192.168.2.2) 56(84) bytes of data:
64 bytes from 192.168.2.2: icmp_seq=1 ttl=255 time=0.576 ms
64 bytes from 192.168.2.2: icmp_seq=2 ttl=255 time=0.571 ms
64 bytes from 192.168.2.2: icmp_seq=3 ttl=255 time=0.426 ms
64 bytes from 192.168.2.2: icmp_seq=4 ttl=255 time=0.359 ms
^C
--- 192.168.2.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 0.359/0.483/0.576/0.093 ms
Jetway# ping 192.168.3.2
PING 192.168.3.2 (192.168.3.2) 56(84) bytes of data:
64 bytes from 192.168.3.2: icmp_seq=1 ttl=127 time=1.74 ms
64 bytes from 192.168.3.2: icmp_seq=2 ttl=127 time=0.486 ms
64 bytes from 192.168.3.2: icmp_seq=3 ttl=127 time=0.765 ms
64 bytes from 192.168.3.2: icmp_seq=4 ttl=127 time=0.799 ms
^C
--- 192.168.3.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 0.486/0.949/1.749/0.478 ms
Jetway# ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data:
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=0.218 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=0.300 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=0.308 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=64 time=0.258 ms
^C
```

Fotografía 1-3: Ping desde Plataforma

Realizado por: Laguna Andrés, Lugmania Juan, 2017

```
Jetway#  
Jetway# traceroute 192.168.2.2  
traceroute to 192.168.2.2 (192.168.2.2), 30 hops max, 60 byte packets  
1 192.168.2.2 (192.168.2.2) 0.855 ms * *  
Jetway# traceroute 192.168.3.2  
traceroute to 192.168.3.2 (192.168.3.2), 30 hops max, 60 byte packets  
1 192.168.2.2 (192.168.2.2) 0.621 ms 0.587 ms 0.513 ms  
2 * * *  
3 * * *  
4 * * *  
5 * 192.168.3.2 (192.168.3.2) 0.713 ms 0.695 ms  
Jetway# traceroute 192.168.1.2  
traceroute to 192.168.1.2 (192.168.1.2), 30 hops max, 60 byte packets  
1 192.168.1.2 (192.168.1.2) 0.368 ms 0.332 ms 0.411 ms  
Jetway# traceroute 192.168.4.2  
traceroute to 192.168.4.2 (192.168.4.2), 30 hops max, 60 byte packets  
1 192.168.1.2 (192.168.1.2) 0.330 ms 0.293 ms 0.534 ms  
2 * * *  
3 * * *  
4 * * *  
5 * * *  
6 192.168.4.2 (192.168.4.2) 1.873 ms 0.636 ms 0.614 ms  
Jetway#
```

Fotografía 2-3: Traceroute desde Plataforma

Realizado por: Laguna Andrés, Lugmania Juan, 2017

3.1.1.2. Prueba 2: OSPF y RIPv2 aplicando redistribución de rutas en IPv4

Como se explicó en el capítulo anterior, para realizar esta prueba se configuro el protocolo de enrutamiento RIPv2 en la plataforma de enrutamiento JETWAY como se puede apreciar en la fotografía 3-3.

G

```

Jetway# show run
Building configuration...

Current configuration:
!
frr version 3.1-dev-MyDunFRRVersion
frr defaults traditional
hostname Jetway
username cunulus nopassword
!
service integrated-vtysh-config
!
log syslog informational
!
interface enp4s0
 ip address 192.168.2.1/30
!
interface enp7s0
 ip address 192.168.1.1/30
!
router rip
 default-information originate
 redistribute ospf
 network 192.168.2.0/30
 network enp4s0
!
router ospf
 redistribute rip
 network 192.168.1.0/30 area 0
 default-information originate
!
line vty
!

```

Fotografía 3-3: Configuración de RIPv2 en Jetway

Realizado por: Laguna Andrés, Lugmania Juan, 2017

Se comprobó un correcto funcionamiento de conectividad entre los dispositivos mediante redistribución de protocolos, sus pruebas podemos comprobarlos en las fotografías 4-3 y 5-3 que se detalla a continuación.

```

Jetway# ping 192.168.2.2
PING 192.168.2.2 (192.168.2.2) 56(84) bytes of data:
64 bytes from 192.168.2.2: icmp_seq=1 ttl=255 time=0.557 ms
64 bytes from 192.168.2.2: icmp_seq=2 ttl=255 time=0.367 ms
64 bytes from 192.168.2.2: icmp_seq=3 ttl=255 time=0.441 ms
64 bytes from 192.168.2.2: icmp_seq=4 ttl=255 time=0.428 ms
^C
--- 192.168.2.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 0.367/0.448/0.557/0.070 ms
Jetway# ping 192.168.3.2
PING 192.168.3.2 (192.168.3.2) 56(84) bytes of data:
64 bytes from 192.168.3.2: icmp_seq=1 ttl=127 time=0.581 ms
64 bytes from 192.168.3.2: icmp_seq=2 ttl=127 time=0.814 ms
64 bytes from 192.168.3.2: icmp_seq=3 ttl=127 time=0.810 ms
64 bytes from 192.168.3.2: icmp_seq=4 ttl=127 time=0.804 ms
^C
--- 192.168.3.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 0.581/0.752/0.814/0.100 ms
Jetway# ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data:
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=0.584 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=0.432 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=0.346 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=64 time=0.348 ms
^C
--- 192.168.1.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 0.346/0.427/0.584/0.098 ms
Jetway# ping 192.168.4.2
PING 192.168.4.2 (192.168.4.2) 56(84) bytes of data:
64 bytes from 192.168.4.2: icmp_seq=1 ttl=127 time=1.20 ms
64 bytes from 192.168.4.2: icmp_seq=2 ttl=127 time=0.837 ms
64 bytes from 192.168.4.2: icmp_seq=3 ttl=127 time=1.06 ms
64 bytes from 192.168.4.2: icmp_seq=4 ttl=127 time=0.945 ms
^C

```

Fotografía 4-3: Pruebas de conectividad

Realizado por: Laguna Andrés, Lugmania Juan, 2017

```
Jetway# traceroute 192.168.3.2
traceroute to 192.168.3.2 (192.168.3.2), 30 hops max, 60 byte packets
 1 192.168.2.2 (192.168.2.2) 0.598 ms 0.598 ms 0.638 ms
 2 * * *
 3 * * *
 4 * * *
 5 * * *
 6 192.168.3.2 (192.168.3.2) 0.648 ms 0.414 ms 0.395 ms
Jetway# traceroute 192.168.1.2
traceroute to 192.168.1.2 (192.168.1.2), 30 hops max, 60 byte packets
 1 192.168.1.2 (192.168.1.2) 0.500 ms 0.463 ms 0.431 ms
Jetway# traceroute 192.168.4.2
traceroute to 192.168.4.2 (192.168.4.2), 30 hops max, 60 byte packets
 1 192.168.1.2 (192.168.1.2) 0.498 ms 0.484 ms 0.472 ms
 2 192.168.4.2 (192.168.4.2) 1.055 ms * *
Jetway#
```

Fotografía 5-3: Análisis de saltos de rutas

Realizado por: Laguna Andrés, Lugmania Juan, 2017

3.1.1.3. Prueba 3: OSPF en IPv6

Como se detalló en el capítulo dos para poder verificar que el dispositivo Jetway sea completamente interoperable, se levantó un escenario con protocolos de internet IPv6, este dispositivo no tuvo inconvenientes para trabajar con este protocolo, la configuración del dispositivo se puede apreciar en la fotografía 6-3 mostrado a continuación:

```
Jetway# show run
Building configuration...

Current configuration:
!
!
frr version 3.1-dev-MyOwnFRRVersion
frr defaults traditional
hostname Jetway
username cumulus nopassword
!
service integrated-vtysh-config
!
log syslog informational
!
interface enp7s0
 ipv6 address 2001:db8:1::2/64
!
interface enp8s0
 ipv6 address 2001:db8:2::1/64
!
router ospf
!
router ospf6
 interface enp7s0 area 0.0.0.0
 interface enp8s0 area 0.0.0.0
!
line vty
```

Fotografía 6-3: Configuración IPv6

Realizado por: Laguna Andrés, Lugmania Juan, 2017

También se pudo comprobar un correcto funcionamiento de conectividad mediante el comando “ping”, además se verifico las rutas que sigue un paquete que se envía a través de la red configurada estas pruebas se realizó mediante el comando “tracert”, el dispositivo tuvo un correcto funcionamiento e interoperabilidad muy eficaz en este protocolo que en la actualidad tiende a reemplazar a IPv4, los resultados de estas pruebas los podemos apreciar detalladamente en las fotografías 7-3 y 8-3.

```

Jetway# ping ipv6 2001:db8:3::2
PING 2001:db8:3::2(2001:db8:3::2) 56 data bytes
64 bytes from 2001:db8:3::2: icmp_seq=1 ttl=63 time=0.812 ms
64 bytes from 2001:db8:3::2: icmp_seq=2 ttl=63 time=0.878 ms
64 bytes from 2001:db8:3::2: icmp_seq=3 ttl=63 time=0.858 ms
64 bytes from 2001:db8:3::2: icmp_seq=4 ttl=63 time=0.816 ms
^C
--- 2001:db8:3::2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/ndev = 0.812/0.841/0.878/0.027 ms
Jetway# ping ipv6 2001:db8:4::2
PING 2001:db8:4::2(2001:db8:4::2) 56 data bytes
64 bytes from 2001:db8:4::2: icmp_seq=1 ttl=63 time=1.98 ms
64 bytes from 2001:db8:4::2: icmp_seq=2 ttl=63 time=0.690 ms
64 bytes from 2001:db8:4::2: icmp_seq=3 ttl=63 time=0.626 ms
64 bytes from 2001:db8:4::2: icmp_seq=4 ttl=63 time=0.694 ms
^C
--- 2001:db8:4::2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/ndev = 0.626/0.997/1.981/0.569 ms
Jetway# ping ipv6 2001:db8:1::1
PING 2001:db8:1::1(2001:db8:1::1) 56 data bytes
64 bytes from 2001:db8:1::1: icmp_seq=1 ttl=64 time=0.745 ms
64 bytes from 2001:db8:1::1: icmp_seq=2 ttl=64 time=0.430 ms
64 bytes from 2001:db8:1::1: icmp_seq=3 ttl=64 time=0.417 ms
64 bytes from 2001:db8:1::1: icmp_seq=4 ttl=64 time=0.424 ms
^C
--- 2001:db8:1::1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/ndev = 0.417/0.504/0.745/0.139 ms
Jetway# ping ipv6 2001:db8:2::2
PING 2001:db8:2::2(2001:db8:2::2) 56 data bytes
64 bytes from 2001:db8:2::2: icmp_seq=1 ttl=64 time=0.519 ms
64 bytes from 2001:db8:2::2: icmp_seq=2 ttl=64 time=0.640 ms
64 bytes from 2001:db8:2::2: icmp_seq=3 ttl=64 time=0.523 ms
64 bytes from 2001:db8:2::2: icmp_seq=4 ttl=64 time=0.520 ms
^C
--- 2001:db8:2::2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/ndev = 0.519/0.550/0.640/0.056 ms

```

Fotografía 7-3: Conectividad con OSPF en IPv6

Realizado por: Laguna Andrés, Lugmania Juan, 2017

```

Jetway#
Jetway#
Jetway#
Jetway#
Jetway# tracer ipv6 2001:db8:3::2
traceroute to 2001:db8:3::2 (2001:db8:3::2) from 2001:db8:2:0:230:18ff:fec9:36f7, 30 hops max, 24 byte packets
 1 2001:db8:1::1 (2001:db8:1::1) 0.523 ns 0.758 ns 0.347 ns
 2 2001:db8:3::2 (2001:db8:3::2) 0.839 ns 0.823 ns 0.838 ns
Jetway# tracer ipv6 2001:db8:4::2
traceroute to 2001:db8:4::2 (2001:db8:4::2) from 2001:db8:2:0:230:18ff:fec9:36f8, 30 hops max, 24 byte packets
 1 2001:db8:2::2 (2001:db8:2::2) 0.678 ns 0.818 ns 0.539 ns
 2 2001:db8:4::2 (2001:db8:4::2) 0.877 ns 1.462 ns 1.244 ns
Jetway#

```

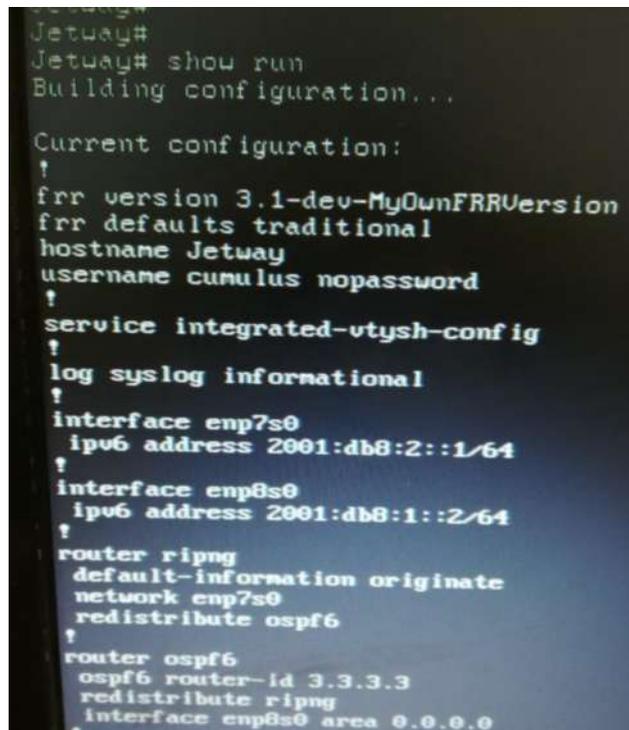
Fotografía 8-3: Análisis de saltos en IPv6

Realizado por: Laguna Andrés, Lugmania Juan, 2017

Para analizar un completa interoperabilidad también se aplicó redistribución en Ipv6 como podemos observar a continuación.

3.1.1.4. Prueba 4: OSPF y RIPng aplicando redistribución de rutas en IPv6

En la fotografía 9-3 se puede apreciar la configuración de RIPng y OSPF en el enrutador, y aplicando redistribución el dispositivo tuvo una gran facilidad de conexión mostrando una completa interoperabilidad entre dispositivos de diversos fabricantes.



```
Jetway#
Jetway# show run
Building configuration...

Current configuration:
!
frr version 3.1-dev-MyOwnFRRVersion
frr defaults traditional
hostname Jetway
username cumulus nopassword
!
service integrated-vtysh-config
!
log syslog informational
!
interface emp7s0
 ipv6 address 2001:db8:2::1/64
!
interface emp8s0
 ipv6 address 2001:db8:1::2/64
!
router ripng
 default-information originate
 network emp7s0
 redistribute ospf6
!
router ospf6
 ospf6 router-id 3.3.3.3
 redistribute ripng
 interface emp8s0 area 0.0.0.0
!
```

Fotografía 9-3: Configuración OSPF y RIPng en IPv6

Realizado por: Laguna Andrés, Lugmania Juan, 2017

Los resultados de la correcta interoperabilidad los podemos observar en las fotografías 10-3 y 11-3 mostrados a continuación.

```
line vty
↑
end
Jetway#
Jetway# ping ipv6 2001:db8:4::2
PING 2001:db8:4::2(2001:db8:4::2) 56 data bytes
64 bytes from 2001:db8:4::2: icmp_seq=1 ttl=63 time=1.16 ms
64 bytes from 2001:db8:4::2: icmp_seq=2 ttl=63 time=1.32 ms
64 bytes from 2001:db8:4::2: icmp_seq=3 ttl=63 time=1.13 ms
64 bytes from 2001:db8:4::2: icmp_seq=4 ttl=63 time=1.33 ms
^C
--- 2001:db8:4::2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/ndev = 1.131/1.238/1.333/0.099 ms
Jetway# ping ipv6 2001:db8:3::2
PING 2001:db8:3::2(2001:db8:3::2) 56 data bytes
64 bytes from 2001:db8:3::2: icmp_seq=1 ttl=63 time=0.678 ms
64 bytes from 2001:db8:3::2: icmp_seq=2 ttl=63 time=0.752 ms
64 bytes from 2001:db8:3::2: icmp_seq=3 ttl=63 time=0.768 ms
64 bytes from 2001:db8:3::2: icmp_seq=4 ttl=63 time=0.761 ms
^C
```

Fotografía 10-3: Pruebas de conectividad OSPF y RIPng

Realizado por: Laguna Andrés, Lugmania Juan, 2017

```
Jetway# tracer ipv6 2001:db8:3::2
traceroute to 2001:db8:3::2 (2001:db8:3::2) from 2001:db8:1:0:230:18ff:fec9:36f8, 30 hops max, 24 byte packets
 1 2001:db8:1::1 (2001:db8:1::1) 0.78 ms 0.504 ms 0.437 ms
 2 2001:db8:3::2 (2001:db8:3::2) 0.409 ms * 0.403 ms
Jetway# tracer ipv6 2001:db8:4::2
traceroute to 2001:db8:4::2 (2001:db8:4::2) from 2001:db8:2:0:230:18ff:fec9:36f7, 30 hops max, 24 byte packets
 1 2001:db8:2::2 (2001:db8:2::2) 0.596 ms 0.367 ms 0.249 ms
 2 2001:db8:4::2 (2001:db8:4::2) 0.863 ms * 0.866 ms
Jetway#
```

Fotografía 11-3: Pruebas de conectividad OSPF y RIPng

Realizado por: Laguna Andrés, Lugmania Juan, 2017

Para una demostración más detallada de la conectividad en este este escenario ponemos a continuación las pruebas de conectividad desde los host en las figuras 5-3 y 6-3.

```

C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.
C:\Users\juan>ipconfig

Configuración IP de Windows

Adaptador de Ethernet Conexión de área local 3:

Estado de los medios. . . . . = medios desconectados
Sufijo DNS específico para la conexión. . :

Adaptador de Ethernet Conexión de área local:

Sufijo DNS específico para la conexión. . :
Dirección IPv6 . . . . . = 2001:db8:4::2
Dirección IPv6 . . . . . = 2001:db8:4:0:2092:9fa6:292d:d185
Dirección IPv6 temporal. . . . . = 2001:db8:4:0:b049:43bd:7c23:2bb
Vínculo: dirección IPv6 local. . . . . = fe80::2092:9fa6:292d:d185%11
Dirección IPv4 de configuración automática: 169.254.209.133
Máscara de subred . . . . . = 255.255.0.0
Puerta de enlace predeterminada . . . . = 2001:db8:4::1
                                           fe80::669e:f3ff:fe30:bfd1%11

```

Figura 5-3: Configuración Pc1

Realizado por: Laguna Andrés, Lugmania Juan, 2017

```

C:\Users\juan>
C:\Users\juan>tracert 2001:db8:2::1

Traza a 2001:db8:2::1 sobre caminos de 30 saltos como máximo.

 1  <1 ns  <1 ns  <1 ns  2001:db8:4::1
 2  3 ns   <1 ns  <1 ns  2001:db8:2::1

Traza completa.
C:\Users\juan>
C:\Users\juan>
C:\Users\juan>
C:\Users\juan>ping 2001:db8:2::1

Haciendo ping a 2001:db8:2::1 con 32 bytes de datos:
Respuesta desde 2001:db8:2::1: tiempo=1ns
Respuesta desde 2001:db8:2::1: tiempo=1ns
Respuesta desde 2001:db8:2::1: tiempo=1ns
Respuesta desde 2001:db8:2::1: tiempo=1ns

Estadísticas de ping para 2001:db8:2::1:
Paquetes: enviados = 4, recibidos = 4, perdidos = 0
(0% perdidos).
Tiempos aproximados de ida y vuelta en milisegundos:
Mínimo = 1ns, Máximo = 1ns, Media = 1ns

C:\Users\juan>ping 2001:db8:1::1

Haciendo ping a 2001:db8:1::1 con 32 bytes de datos:
Respuesta desde 2001:db8:1::1: tiempo=2ns
Respuesta desde 2001:db8:1::1: tiempo=1ns
Respuesta desde 2001:db8:1::1: tiempo=1ns
Respuesta desde 2001:db8:1::1: tiempo=1ns

Estadísticas de ping para 2001:db8:1::1:
Paquetes: enviados = 4, recibidos = 4, perdidos = 0
(0% perdidos).
Tiempos aproximados de ida y vuelta en milisegundos:
Mínimo = 1ns, Máximo = 2ns, Media = 1ns

```

Figura 6-3: Pruebas de conectividad en Pc1

Realizado por: Laguna Andrés, Lugmania Juan, 2017

Al haber finalizado las pruebas de interoperabilidad presentamos a continuación un breve resumen y una comparativa para poder observar un nivel de interoperabilidad que posee el dispositivo.

3.1.1.5. *Resultados finales de interoperabilidad*

Para poder tener un mejor entendimiento hemos decidido realizar un cuadro comparativo, en el cual se muestra en que escenarios hubo buena conectividad, esto con el fin de obtener el nivel de interoperabilidad, la tabla se muestra continuación a continuación.

Tabla 1-3: Resultados finales

Protocolo	OSPF	OSPF y RIP
IPv4	✓	✓
IPv6	✓	✓

Realizado por: Laguna Andrés, Lugmania Juan, 2017

Como se pudo apreciar en los resultados, el dispositivo de enrutamiento tuvo un funcionamiento apropiado a todos los escenarios propuestos, y como se observa en la tabla la plataforma soporta los protocolos de internet comúnmente utilizados en la actualidad, en otras palabras se puede decir que la plataforma de enrutamiento tuvo una completa interoperabilidad.

Después de analizar las imágenes vistas anteriormente, todas las peticiones hechas desde las PCs y la Plataforma son realizadas con éxito, además podemos apreciar que los paquetes enviados atraviesan por cada una de las direcciones que estas necesitan sin ningún inconveniente, es decir al observar la tabla expuesta se confirma que la plataforma es 100% interoperable con dispositivos de diversas marcas en una red, y se vincula fácilmente con los protocolos de enrutamiento existentes en la actualidad.

3.1.2. Pruebas de rendimiento

Con la utilización de Wireshark pudimos analizar el tráfico inyectado a la red mediante Ostinato, y así comparar sus resultados entre los dos escenarios propuestos en el Capítulo II, como se explicó en dicho capítulo se inyectó 1'600.000 paquetes como se puede observar en la figura 7-3 mostrada a continuación.

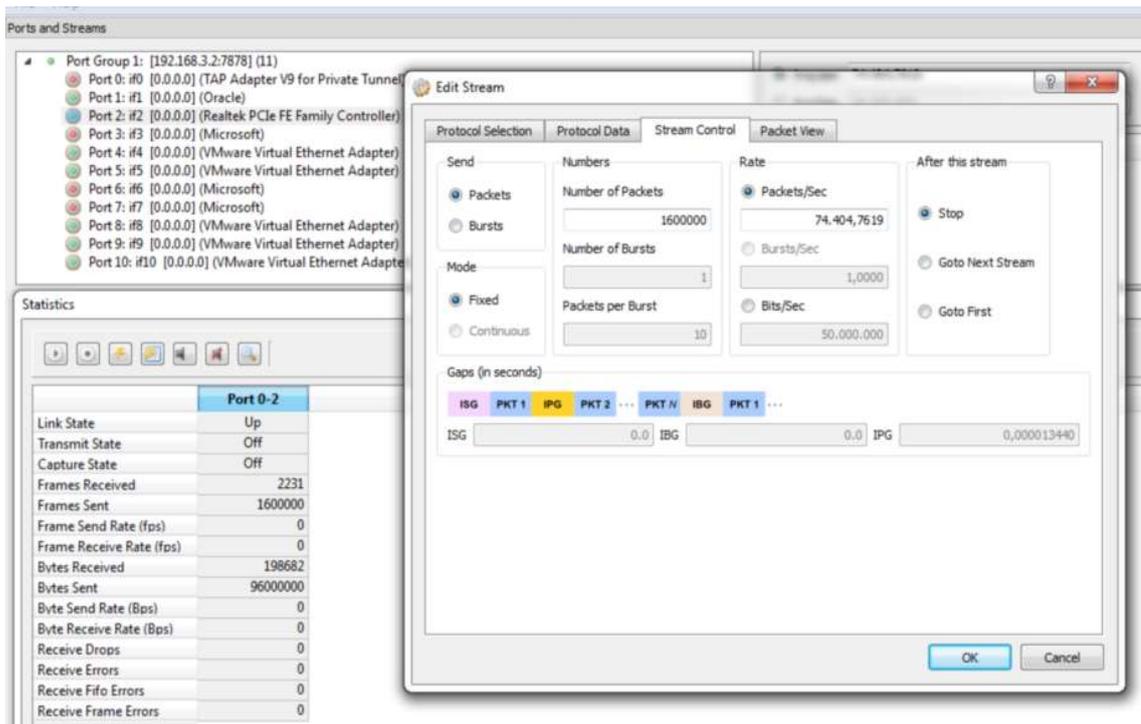


Figura 7-3: Inyección de tráfico en Ostinato

Realizado por: Laguna Andrés, Lugmania Juan, 2017

3.1.2.1. Prueba 5: Generación de tráfico con Ostinato en IPv4.

Al inyectar tráfico en los dos escenarios, y después de haber analizado los datos se obtuvo una mayor eficiencia en la transmisión de paquetes en el primer escenario, haciéndonos llegar a la conclusión que la plataforma Jetway es mucho más eficiente en la transmisión de datos que los routers Cisco 2911, los resultados de dichas pruebas las podemos apreciar en las figuras 8-3, 9-3, 10-3 y 11-3.

No.	Time	Source	Destination	Protocol	Length	Info
1600000	15.879626	192.168.3.2	192.168.4.2	UDP	60	0 → 0 Len=18
1600001	15.879633	192.168.3.2	192.168.4.2	UDP	60	0 → 0 Len=18

Frame 1600000: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
 Encapsulation type: Ethernet (1)
 Arrival Time: Oct 5, 2017 13:16:46.959500000 Hora est. Pacífico, Sudamérica
 [Time shift for this packet: 0.000000000 seconds]
 Epoch Time: 1507227406.959500000 seconds
 [Time delta from previous captured frame: 0.00005000 seconds]
 [Time delta from previous displayed frame: 0.00005000 seconds]
 [Time since reference or first frame: 15.879626000 seconds]
 Frame Number: 1600000
 Frame Length: 60 bytes (480 bits)
 Capture Length: 60 bytes (480 bits)
 [Frame is marked: False]
 [Frame is ignored: False]
 [Protocols in frame: eth:ethertype:ip:udp:data]
 [Coloring Rule Name: UDP]
 [Coloring Rule String: udp]

Ethernet II, Src: Dell_b2:9e:be (e0:db:55:b2:9e:be), Dst: Cisco_2f:52:01 (f4:0f:1b:2f:52:01)
 Destination: Cisco_2f:52:01 (f4:0f:1b:2f:52:01)
 Source: Dell_b2:9e:be (e0:db:55:b2:9e:be)
 Type: IPv4 (0x0800)

Internet Protocol Version 4, Src: 192.168.3.2, Dst: 192.168.4.2
 0100 = Version: 4
 0101 = Header Length: 20 bytes (5)
 Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
 Total Length: 46
 Identification: 0x04d2 (1234)
 Flags: 0x00
 Fragment offset: 0
 Time to live: 127
 Protocol: UDP (17)

```

0000 f4 0f 1b 2f 52 01 e0 db 55 b2 9e be 08 00 45 00 .../R... U.....E.
  
```

Figura 8-3: Wireshark Escenario 1 Jetway

Realizado por: Laguna Andrés, Lugmania Juan, 2017

No.	Time	Source	Destination	Protocol	Length	Info
1600000	16.388594	192.168.3.2	192.168.4.2	UDP	60	0 → 0 Len=18
1600001	16.388602	192.168.3.2	192.168.4.2	UDP	60	0 → 0 Len=18

Frame 1600000: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
 Encapsulation type: Ethernet (1)
 Arrival Time: Oct 5, 2017 13:11:30.495037000 Hora est. Pacífico, Sudamérica
 [Time shift for this packet: 0.000000000 seconds]
 Epoch Time: 1507227090.495037000 seconds
 [Time delta from previous captured frame: 0.00005000 seconds]
 [Time delta from previous displayed frame: 0.00005000 seconds]
 [Time since reference or first frame: 16.388594000 seconds]
 Frame Number: 1600000
 Frame Length: 60 bytes (480 bits)
 Capture Length: 60 bytes (480 bits)
 [Frame is marked: False]
 [Frame is ignored: False]
 [Protocols in frame: eth:ethertype:ip:udp:data]
 [Coloring Rule Name: UDP]
 [Coloring Rule String: udp]

Ethernet II, Src: Dell_b2:9e:be (e0:db:55:b2:9e:be), Dst: Cisco_2f:52:01 (f4:0f:1b:2f:52:01)
 Destination: Cisco_2f:52:01 (f4:0f:1b:2f:52:01)
 Source: Dell_b2:9e:be (e0:db:55:b2:9e:be)
 Type: IPv4 (0x0800)

Internet Protocol Version 4, Src: 192.168.3.2, Dst: 192.168.4.2
 0100 = Version: 4
 0101 = Header Length: 20 bytes (5)
 Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
 Total Length: 46
 Identification: 0x04d2 (1234)
 Flags: 0x00

```

0000 f4 0f 1b 2f 52 01 e0 db 55 b2 9e be 08 00 45 00 .../R... U.....E.
0010 00 2e 04 d2 00 00 7f 11 ae 98 c0 a8 03 02 c0 a8 .....
0020 04 02 00 00 00 00 00 1a 77 65 00 00 00 00 00 00 ..... we.....
0030 00 00 00 00 00 00 00 00 00 00 00 00 .....
  
```

Figura 9-3: Wireshark Escenario 2 Jetway

Realizado por: Laguna Andrés, Lugmania Juan, 2017

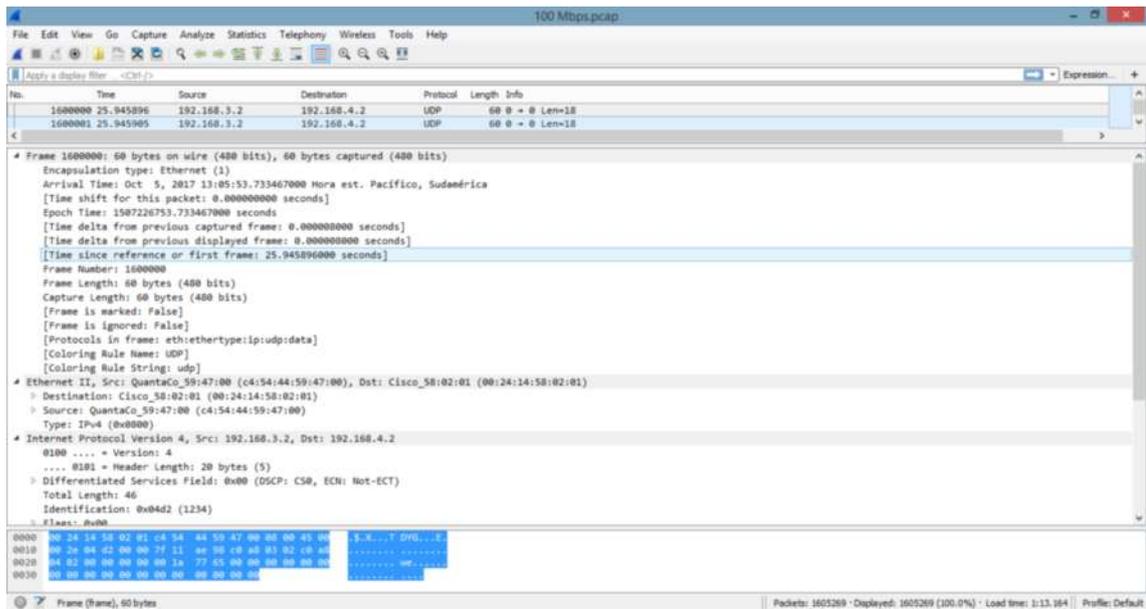


Figura 10-3: Wireshark Escenario 1 Cisco

Realizado por: Laguna Andrés, Lugmania Juan, 2017

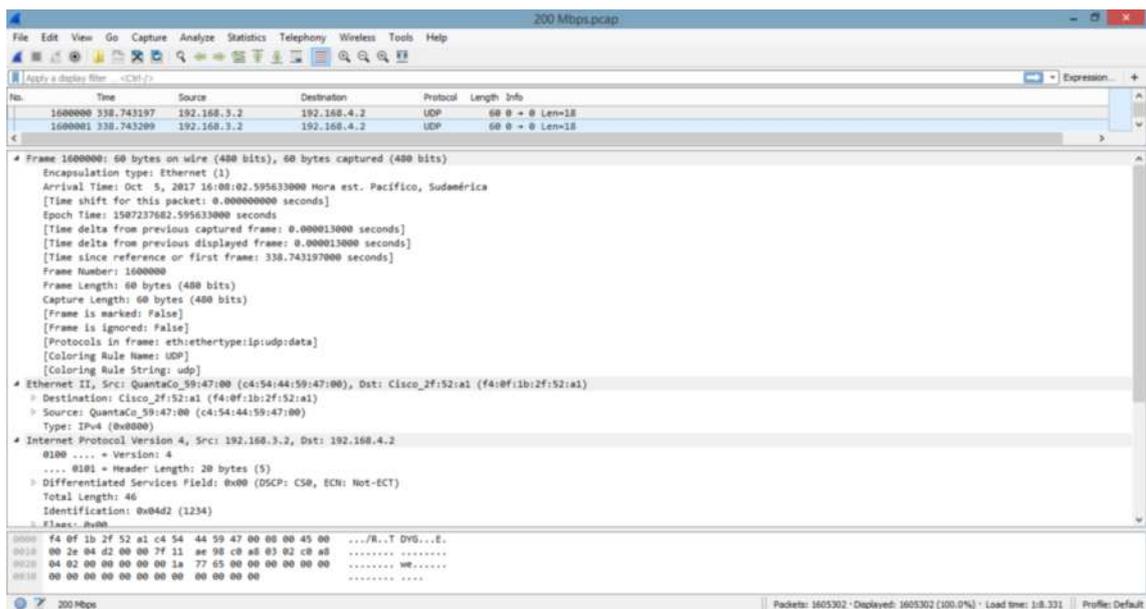


Figura 11-3: Wireshark Escenario 2 Cisco

Realizado por: Laguna Andrés, Lugmania Juan, 2017

Como se puede observar mediante los tiempos de transmisión de los paquetes que nos entrega Wireshark, la transmisión en nuestro dispositivo es mucho mayor; para un mejor entendimiento se ha realizado el cuadro comparativo donde se pone discusión las velocidades de transmisión entre cada escenario. Como podemos observar en la tabla 2-3.

Tabla 2-3: Tiempos de conmutación en IPv4

	Escenario 1		Escenario 2	
	Tiempo de conmutación	Tiempo promedio entre tramas	Tiempo de conmutación	Tiempo promedio entre tramas
50 Mbps	25.12 s	0.014 ms	27.57 s	0.014 ms
90 Mbps	15.53 s	0.007 ms	25.81 s	0.013 ms
100 Mbps	16.38 s	0.005 ms	25.94 s	0.008 ms
200 Mbps	15.87 s	0.005 ms	338 s	0.013 ms

Realizado por: Laguna Andrés, Lugmania Juan, 2017

Utilizando estos resultados detallamos los resultados en una gráfica comparativa ancho de banda vs tiempo mostrada a continuación en el gráfico 1-3.

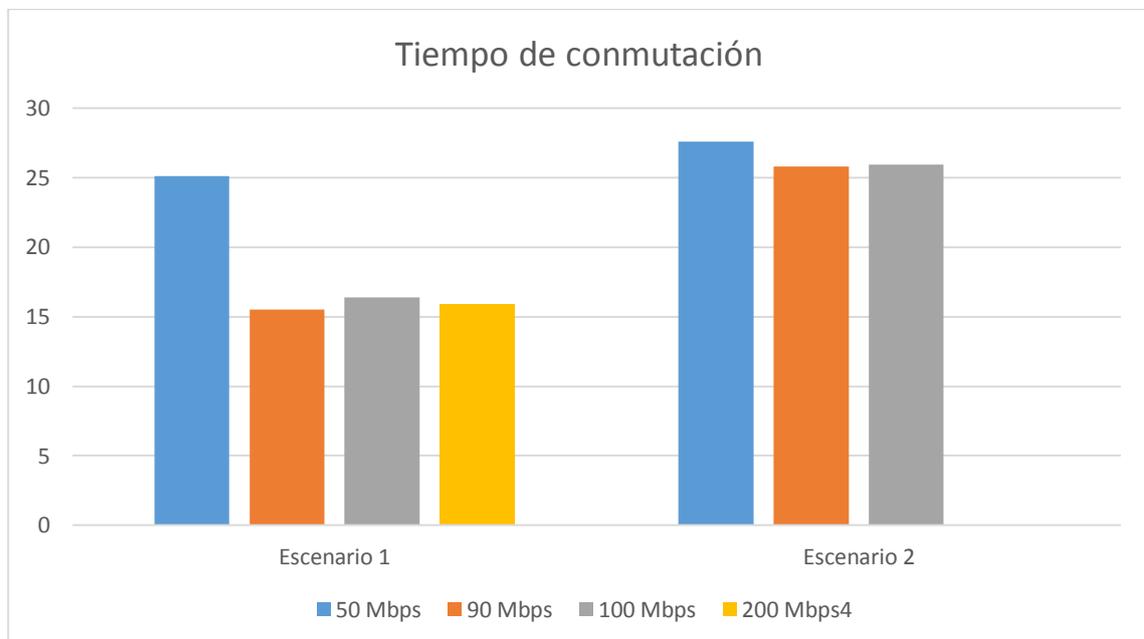


Gráfico 1-3: Tiempo de conmutación en IPv4

Realizado por: Laguna Andrés, Lugmania Juan, 2017

Como se puede apreciar en la tabla, el escenario 1 que incluye la plataforma FRRouting tiene una velocidad de conmutación mucho más elevada que el escenario 2 compuestos por equipos marca Cisco, con estos resultados podemos llegar a la conclusión que nuestro dispositivo de enrutamiento tuvo rendimiento superior, cabe resaltar que estos resultados sean producto que los equipos cisco tengan otras aplicaciones que le consuman demasiados recursos al procesador y la memoria.

3.1.2.2. Prueba 6. Generación de tráfico con Ostinato en IPv6

Al igual que la prueba 5, se inyectó la misma cantidad de paquetes como se indicó en el capítulo 2, después de haber inyectado el tráfico se tuvo los datos en Wireshark mostrados a continuación en las figuras 12-3, 13-3, 14-3 y 15-3.

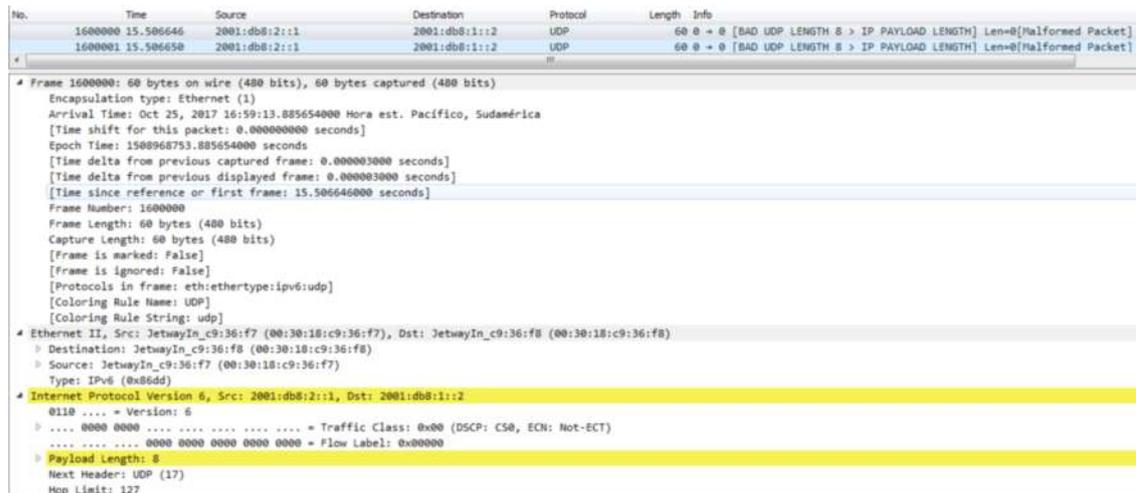


Figura 12-3: Wireshark Escenario 1 Jeway

Realizado por: Laguna Andrés, Lugmania Juan, 2017

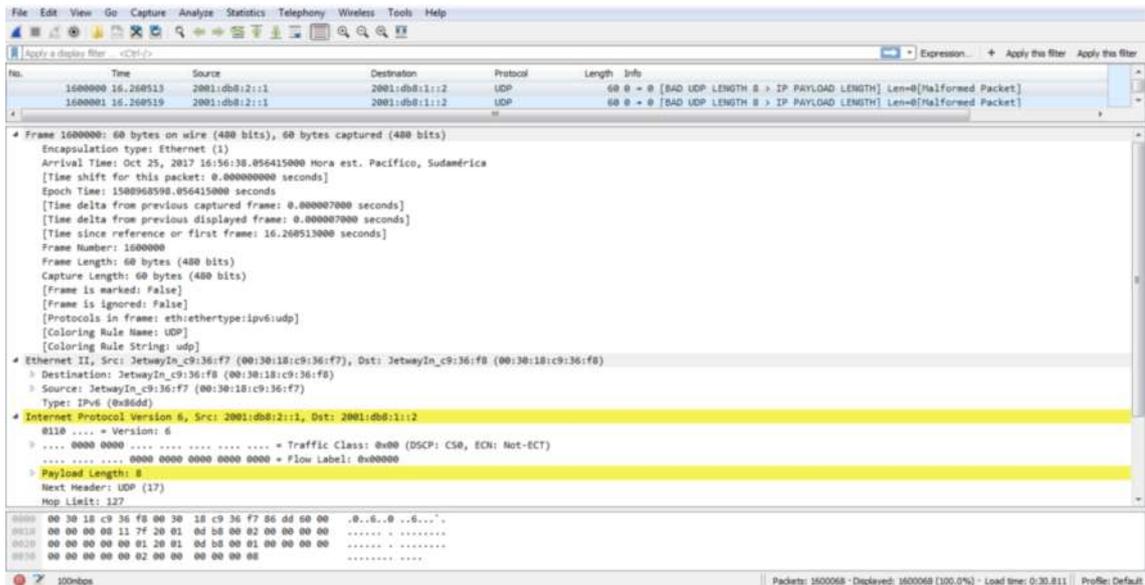


Figura 13-3: Wireshark Escenario 2 Jetway

Realizado por: Laguna Andrés, Lugmania Juan, 2017

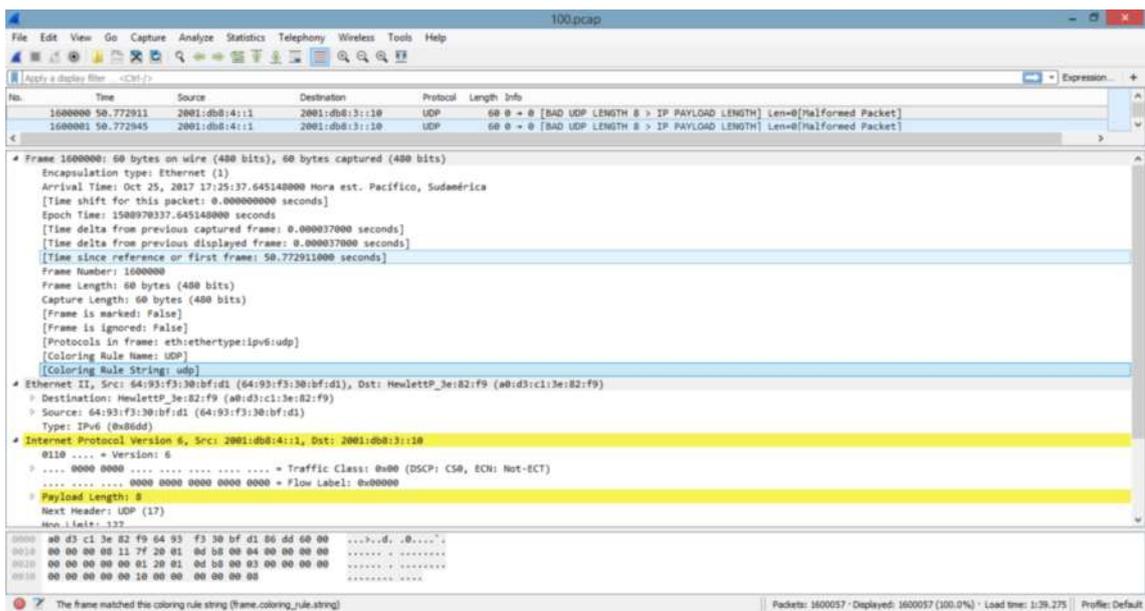


Figura 14-3: Wireshark Escenario 1 Cisco

Realizado por: Laguna Andrés, Lugmania Juan, 2017

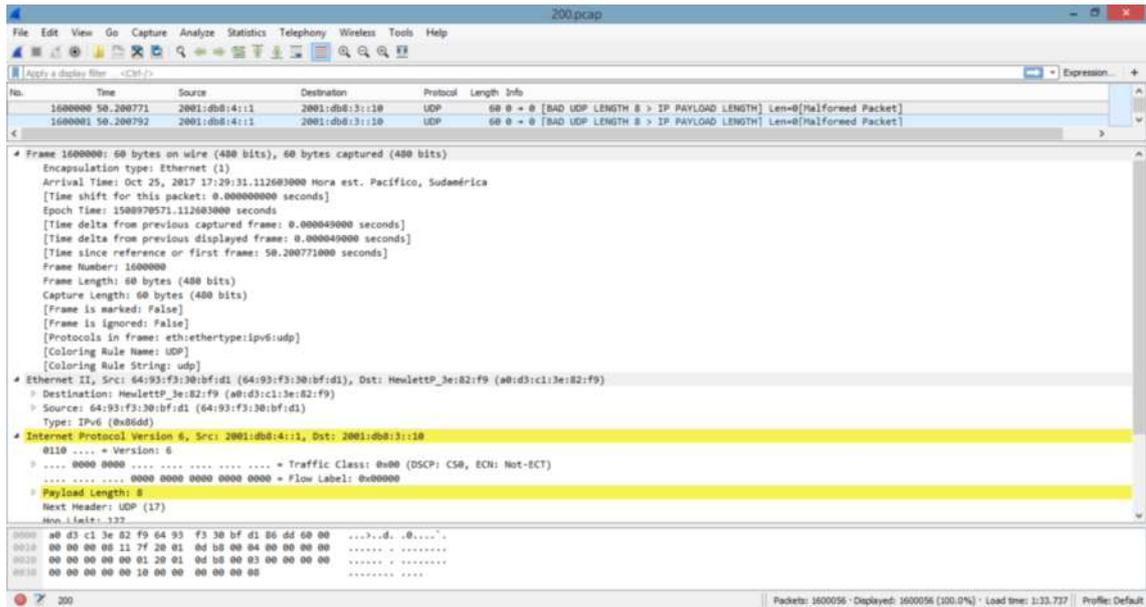


Figura 15-3: Wireshark Escenario 2 Cisco

Realizado por: Laguna Andrés, Lugmania Juan, 2017

Como se puede observar en los tiempos de transmisión tomados en Wireshark la transmisión de los paquetes es mucho más rápida en el dispositivo FRRouting , para un mejor entendimiento se ha realizado el cuadro comparativo donde se pone discusión las velocidades de transmisión entre cada escenario. El cual se muestra en la tabla 3-3.

Tabla 3-3: Tiempos de conmutación en IPv6

	Escenario 1		Escenario 2	
	Tiempo de conmutación	Tiempo promedio entre tramas	Tiempo de conmutación	Tiempo promedio entre tramas
50 Mbps	24.48 s	0.013 ms	53.29 s	0.020 ms
90 Mbps	15.87 s	0.004 ms	43.18 s	0.021 ms
100 Mbps	16.26 s	0.007 ms	50.77 s	0.037 ms
200 Mbps	15.50 s	0.003 ms	50.20 s	0.049 ms

Realizado por: Laguna Andrés, Lugmania Juan, 2017

Utilizando estos resultados detallamos los resultados en una gráfica comparativa ancho de banda vs tiempo mostrada a continuación en el gráfico 2-3.

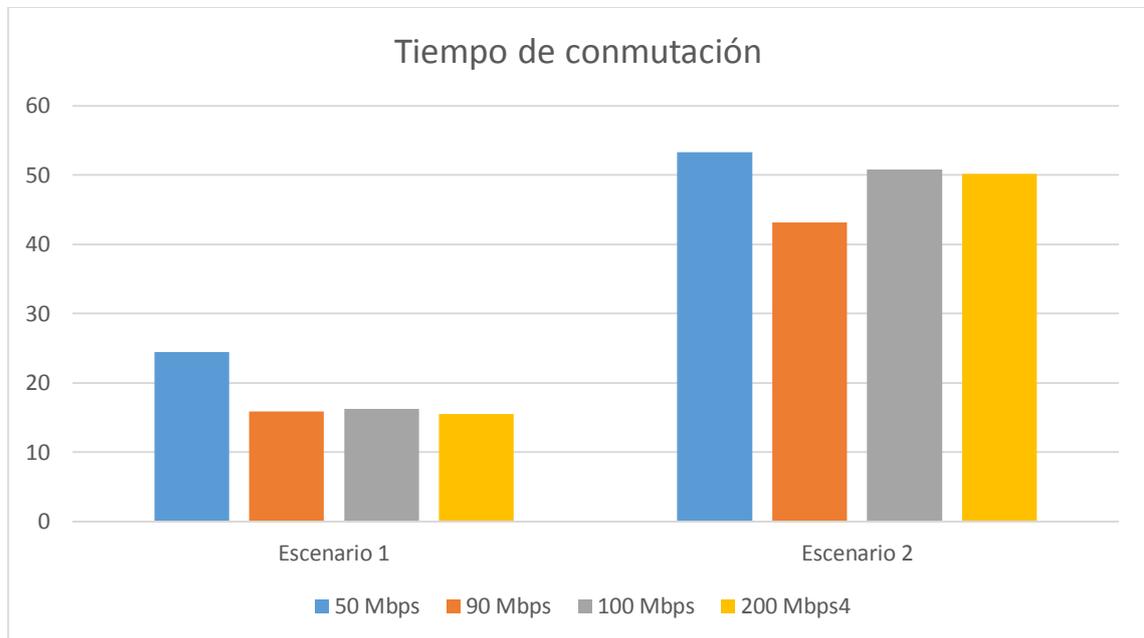


Gráfico 2-3: Tiempo de conmutación en IPv6

Realizado por: Laguna Andrés, Lugmania Juan, 2017

Una vez concluido los análisis mediante generación de tráfico en el protocolo de internet IPv6, se obtuvo que de igual forma que la prueba 5 el rendimiento de la plataforma Jetway fue mucho mayor en comparación al equipo marca Cisco, con estos datos podemos decir que el equipo posee un alto rendimiento y además puede funcionar sin ningún inconveniente en los protocolos de internet más usados en la actualidad

3.1.2.3. Prueba 7: Análisis con vRIN en IPv4

Finalmente, otro parámetro que nos basamos para ver la utilidad de nuestra plataforma fue el análisis de rendimiento mediante generación de rutas, como lo explicamos en el capítulo dos para probar este escenario utilizamos un escenario con un sistema autónomo BGP los resultados los presentamos a continuación

Al generar 120 rutas en BGP en los dos escenarios podemos apreciar los resultados en la figura 16-3 y en la fotografía 12-3.

```

Router#
Router#show ip route summary
IP routing table name is default (0x0)
IP routing table maximum-paths is 32
Route Source      Networks      Subnets      Replicates    Overhead      Memory (bytes)
connected         0             4             0             240           720
static            0             0             0             0             0
ospf 1            0             0             0             0             0
  Intra-area: 0  Inter-area: 0  External-1: 0  External-2: 0
  NSSA External-1: 0  NSSA External-2: 0
bgp 65001         0             121          0             7260          21780
  External: 121  Internal: 0    Local: 0
internal          4
Total             4             125          0             7500          28664
Router#
Router#

```

Figura 16-3: Generación de 120 rutas en Cisco

Realizado por: Laguna Andrés, Lugmania Juan, 2017

```

IPv4 Unicast Summary:
BGP router identifier 192.168.2.1, local AS number 1 vrf-id 0
BGP table version 1657
RIB entries 244, using 36 KiB of memory
Peers 1, using 19 KiB of memory

Neighbor      U      AS  MsgRcvd  MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
192.168.2.2   4      2     64      70       0     0     0 00:00:11      121

Total number of neighbors 1
Jetway# show ip bgp memory
249 RIB nodes, using 37 KiB of memory
123 BGP routes, using 13 KiB of memory
2 Static routes, using 256 bytes of memory
1 Packets, using 56 bytes of memory
123 Adj-Out entries, using 8856 bytes of memory
1 Nexthop cache entries, using 72 bytes of memory
5 BGP attributes, using 1040 bytes of memory
2 BGP AS-PATH entries, using 80 bytes of memory
1 BGP AS-PATH segments, using 24 bytes of memory
3 peers, using 58 KiB of memory
398 hash tables, using 25 KiB of memory
1970 hash buckets, using 62 KiB of memory
Jetway#

```

Fotografía 12-3: Generación de 120 rutas en Jetway

Realizado por: Laguna Andrés, Lugmania Juan, 2017

También se hizo un análisis inyectado 200 rutas BGP y viendo el consumo de memoria, los resultados los encontramos en la figura 17-3 y en la fotografía 13-3.

```

*Oct 17 22:04:29.393: %BGP_SESSION/5/ADICCHANGE: neighbor 192.168.10.10 IPv4 Unicast topology
*Oct 17 22:04:29.507: %BGP/5/ADICCHANGE: neighbor 192.168.10.10 Up show ip route summary
IP routing table name is default. (0x0)
IP routing table maximum-paths is 32
Route Source      Networks      Subnets      Replicates    Overhead      Memory (bytes)
connected         0              4              0              240           720
static            0              0              0              0              0
ospf 1           0              0              0              0              0
  Intra-area: 0 Inter-area: 0 External-1: 0 External-2: 0
  NSSA External-1: 0 NSSA External-2: 0
bgp 65001         0              201            0              12060         36180
  External: 201 Internal: 0 Local: 0
internal         4              0              0              0              9364
Total            4              205            0              12300         46264
Router#
Router#

```

Figura 17-3: Generación de 200 rutas en Cisco

Realizado por: Laguna Andrés, Lugmania Juan, 2017

```

IPv4 Unicast Summary:
BGP router identifier 192.168.2.1, local AS number 1 vrf-id 0
BGP table version 1335
RIB entries 403, using 60 KIB of memory
Peers 1, using 19 KIB of memory

Neighbor      U      AS  MsgRcvd  MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
192.168.2.2  4      2    52      57       0     0     0 00:10:50    201

Total number of neighbors 1
Jetways show ip bgp
attribute-info      dampening          l2vpn             multicast          prefix-list        unicast
cidr-only           filter-list        large-community   neighbors          regexp             update-groups
community           ipv4               large-community-info  nexthop           route-map          view
community-info     ipv6               large-community-list  paths              statistics          views
community-list     json               memory             peer-group         summary            vpn
Jetways show ip bgp memory
408 RIB nodes, using 61 KIB of memory
203 BGP routes, using 22 KIB of memory
2 Static routes, using 256 bytes of memory
1 Packets, using 56 bytes of memory
203 Adj-Out entries, using 14 KIB of memory
1 Nexthop cache entries, using 72 bytes of memory
5 BGP attributes, using 1040 bytes of memory
2 BGP AS-PATH entries, using 80 bytes of memory
1 BGP AS-PATH segments, using 24 bytes of memory
3 peers, using 58 KIB of memory
398 hash tables, using 25 KIB of memory
2129 hash buckets, using 67 KIB of memory
Jetways#

```

Fotografía 13-3: Generación de 200 rutas en Jetway

Realizado por: Laguna Andrés, Lugmania Juan, 2017

Estos resultados los hemos resumido en la tabla que se muestra a continuación en la tabla 4-3:

Tabla 4-3: Memoria consumida

Número de rutas	120 Rutas	200 Rutas
Modelo de equipo		
Jetway	13 KBytes	21,78 KBytes
Cisco	22 KBytes	36,18 KBytes

Realizado por: Laguna Andrés, Lugmania Juan, 2017

Al evidenciar en la tabla de resumen, el equipo Jetway muestra un menor consumo de memoria ante la generación de las rutas, esto se debe al procesador de alto rendimiento incorporada en la tarjeta y al poco consumo de memoria del Sistema operativo, con todos estos parámetros podemos decir que la Plataforma de enrutamiento Hw/ Sw OpenSource es un dispositivo de alto rendimiento ante escenarios de enrutamiento.

3.2. Comparación de precios de plataformas de enrutamiento

Finalmente, una vez analizado que nuestro dispositivo es completamente interoperable y con un óptimo rendimiento, hemos visto la necesidad de realizar una comparativa de los precios que tienen los equipos de enrutamiento frente a nuestro dispositivo, como se observa en la tabla 5-3

Tabla 5-3: Comparación de precios

Marca	Costo
Jetway NF533	\$ 350
Cisco	\$1200 hasta \$16000
Mikrotik	\$1000 hasta \$4000
Juniper	\$1000 hasta \$ 6000
Huawei	\$1000 hasta \$3500

Realizado por: Laguna Andrés, Lugmania Juan, 2017

CONCLUSIONES

- Con el estudio de las plataformas de enrutamiento Opensource, se tiene una opción aproximadamente de un 60% más económica para poder realizar pruebas de conexión de redes de computadores que no estén basados solo en simulaciones.
- Después de las pruebas realizadas en los escenarios de generación de tráfico en IPv4 de nuestro dispositivo en comparativa con los equipos Cisco que posee la Cisco Networking Academy, se comprobó un rendimiento superior de nuestra plataforma de enrutamiento, también se evidenció un crecimiento de este rendimiento de una manera directamente proporcional al aumento del ancho de banda.
- De la misma manera al analizar el tráfico generado en la red utilizando el protocolo de internet IPv6, se demostró que la velocidad de transmisión en nuestra plataforma de enrutamiento aumento aproximadamente un 40% en relación a la otra plataforma de enrutamiento Cisco utilizada para este análisis.
- Al aplicar la técnica de rendimiento mediante generación de rutas, se obtuvo una optimización de memoria en nuestra plataforma notablemente superior en comparación con el equipo Cisco usado para esta comparativa, este resultado puede ser debido a que dicho equipo tiene aplicaciones dedicadas que consumen recurso de memoria al sistema.
- La plataforma de enrutamiento Opensource, representa una opción más viable desde el punto de vista económico para empresas que no requieran un área dedicada a redes avanzadas.

RECOMENDACIONES

- Al momento de adquirir la placa Hardware se recomienda, ver la compatibilidad entre dicha placa y la plataforma de enrutamiento a instalar.
- Es aconsejable realizar más pruebas de interoperabilidad con los equipos de enrutamiento mayormente utilizados en la actualidad (Hp, Juniper, Brocade, Huawei, etc.).
- Como estudiantes familiarizarse con las plataformas de enrutamiento Opensource existentes para así, mediante estos desarrollar sus propios dispositivos de enrutamiento con menor costo y mayor accesibilidad económica.
- Investigar los diversos sistemas de pruebas para rendimiento existentes y de esta forma poder entender el óptimo funcionamiento de nuestros dispositivos de enrutamiento.

BIBLIOGRAFÍA

AGUIRRE, J. & MARTÍN, G. *Diseño e implementación de un algoritmo para la configuración de dos tecnologías de routers, utilizando el software de enrutamiento quagga.* (Tesis) (Pregrado). Universidad Distrital Francisco Jose Candas, Bogotá. 2017. pp. 10-15.

ANA, S., *Protocolos de Enrutamiento Dinámico (CISCO).* [En línea]. [Consulta: 21 octubre 2017]. Disponible en: <https://espaxioinformativo.wordpress.com/2015/08/15/protocolos-de-enrutamiento-dinamico-cisco/>

CASTAÑO, A., OCAMPO, L. & OROZCO, M. *FUNDAMENTACIÓN TEÓRICA Y EXPLORACIÓN DE GENERADORES DE TRAFICO.* (Tesis) (Pregrado). Universidad Católica de Pereira, Facultad de Ciencias Básicas e Ingeniería, Ingeniería en Sistemas y Telecomunicaciones, Colombia. 2011. pp. 4-30.

CASTELLS, Manuel. "Internet y la sociedad red". *Universitat Oberta de Catalunya.* [En línea], 2000 (España), 1(1). pp. 4-10. [Consulta: 18 octubre 2017]. Disponible en: [https://engage.intel.com/servlet/JiveServlet/downloadBody/26111-102-1-31790/INTERNET Y LA SOCIEDAD RED.pdf](https://engage.intel.com/servlet/JiveServlet/downloadBody/26111-102-1-31790/INTERNET_Y_LA_SOCIEDAD_RED.pdf)

DOSZTAL, A. *vRIN - virtual Route Injector.* [En línea]. [Consulta: 27 Octubre 2017]. Disponible en: <https://gns3.com/news/article/vrin-virtual-route-injector>

GALEON. *Mainborad.* [En línea]. [Consulta: 26 Octubre 2017] Disponible en: <http://gloriaveliz2007.galeon.com>

COMBS, Gerald . *Wireshark.* [En línea]. 1998. [Consulta: 27 Octubre 2017]. Disponible en: <https://www.wireshark.org>

GUILLESQ, 2007. *GuilleSQL.* [En línea]. [Consulta: 20 Octubre 2017]. Disponible en: http://www.guillesq.es/Articulos/Manual_Cisco_CCNA.aspx

INCIBE. *INCIBE.* [En línea]. 2010. [Consulta: 26 Octubre 2017]. Disponible en: <https://www.incibe.es>

KCIUK, M. "OpenWRT operating system based controllers for mobile robot and building automation system students projects realization". *IEEE*. [En línea]. 1(1),(2014), (Polonia). [Consulta: 26 Octubre 2017]. Disponible: ieeexplore.ieee.org/docu

KUROSE, J. *Computer Networking*. 6ª ed. Estados Unidos de América. Pearson, 2001, pp. 360-40.

LINUX FUNDATION. *Frrouting*. [En línea]. 2017. [Consulta: 24 Octubre 2017]. Disponible en: <https://frrouting.org>

NOROÑO, N. "Influencia del tamaño de paquetes sobre la pérdida de paquetes en un enlace udp/ip/ieee 802.11ª ". *Universidad Rafael Belloso Chasin*. [En línea], 2012 (Venezuela), 1(1). pp. 7-10 [Consulta: 24 Octubre 2017]. Disponible: <http://publicaciones.urbe.edu/index.php/telematique/article/viewArticle/1606/html>

OPENWRT. *OpenWRT Wireless Freedom*. [En línea]. 2017. [Consulta: 23 Octubre 2017]. Disponible en: <http://wiki.openwrt.org/doc/start>

RICHARTE, J. *MotherBoards*. 1ª ed. Buenos Aires - Argentina. Dalaga, 2012, pp. 98-140.

RIVERA, D. & SOSA, F. *Estudio del funcionamiento y uso de la plataforma de enrutamiento xorp* (Tesis) (Pregrado). Universidad de Pereira, Facultad de Ciencias Básicas e Ingeniería, Ingeniería Electrónica [En línea]. Colombia, 2014. [Consulta: 22 Octubre 2017] Disponible: <http://repositorio.ucp.edu.co:8080/jspui/bitstream/10785/2872/3/CDMIST90.pdf>

SÁNCHEZ, M. *Plataforma factible para routers*. [En línea], 2004. [Consulta: 23 Octubre 2014]. Disponible en: <http://www.ilustrados.com/tema/10776/Plataforma-factible-para-routers.html>

SLASHDOT, MEDIA. *SourceForge*. [En línea], 2017. [Consulta: 23 Octubre 2017]. Disponible en: <https://sourceforge.net/projects/vrin/files/>

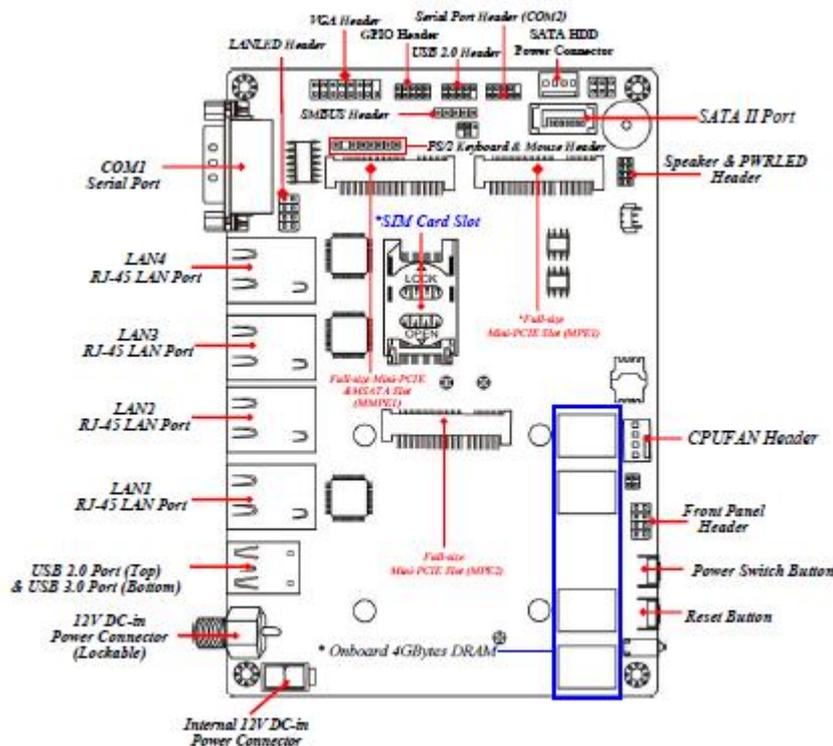
SRIVATS, P. *Ostinato*. [En línea], 2017. [Consulta: 25 Octubre 2017]. Disponible en: <http://ostinato.org>

STALLINGS, W. *Comunicaciones y redes de computadores*. 7ª ed. Madrid- España Pearson Education, 2004, pp. 10-33.

YANUZZI, M. "Open Issues in Interdomain Routing: A Survey" [En línea]. *IEEE*. 19(1), (2005), (Belgica). [Consulta: 25 Octubre 2017]. Disponible en : <http://ieeexplore.ieee.org/document/6920248/>

ANEXOS

Anexo A: Manual

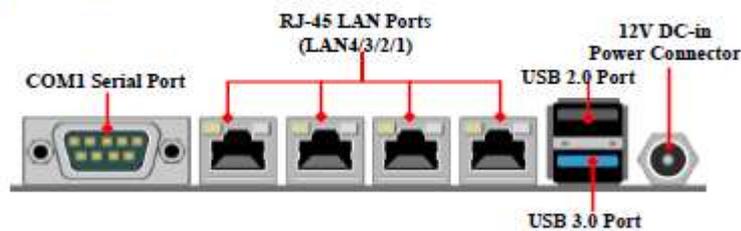


Note: SIM card slot only work when compatible SIM card installed & 3G LAN card installed in MPE1 Mini-PCIE slot.

2-2 Connectors and Headers

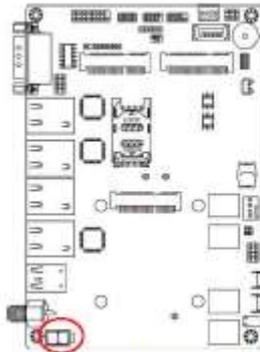
2-2-1 Connectors

(1) Rear I/O Connectors



Icon	Name	Function
	12V DC-in Power Connector	For user to connect compatible power adapter to provide power supply for the system.
	USB 2.0 Port	To connect USB keyboard, mouse or other devices compatible with USB specification.
	USB 3.0 Port	To connect USB keyboard, mouse or other devices compatible with USB specification. USB 3.0 ports supports up to 5Gbps data transfer rate.
	RJ-45 LAN Port	This connector is standard RJ-45 LAN jack for Network connection.
	RS232 Serial Port	Mainly for user to connect external MODEM or other devices that supports Serial Communications Interface.

(2) ATX2P (2-pin Block): Internal 12V DC-in Power Connector



Pin.	Definition
1	GND
2	+12V DC_IN

(3) SATA1 (7-pin Block): SATAII Port connector

SATA1 port is a high-speed SATAII port that supports 3GB/s transfer rate.



Pin No.	Definition
1	GND
2	TXP
3	TXN
4	GND
5	RXN
6	RXP
7	GND



Introducing BIOS

Notice! The BIOS options in this manual are for reference only. Different configurations may lead to difference in BIOS screen and BIOS screens in manuals are usually the first BIOS version when the board is released and may be different from your purchased motherboard. Users are welcome to download the latest BIOS version from our official website.

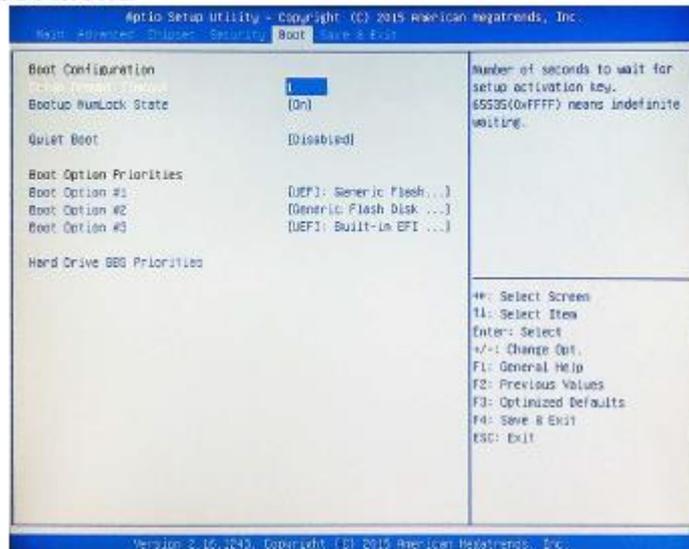
The BIOS is a program located on a Flash Memory on the motherboard. This program is a bridge between motherboard and operating system. When you start the computer, the BIOS program will gain control. The BIOS first operates an auto-diagnostic test called POST (power on self test) for all the necessary hardware, it detects the entire hardware device and configures the parameters of the hardware synchronization. Only when these tasks are completed done it gives up control of the computer to operating system (OS). Since the BIOS is the only channel for hardware and software to communicate, it is the key factor for system stability, and in ensuring that your system performance as its best.

3-1 Entering Setup

Power on the computer and by pressing immediately allows you to enter Setup. If the message disappears before your respond and you still wish to enter Setup, restart the system to try again by turning it OFF then ON or pressing the "RESET" button on the system case. You may also restart by simultaneously pressing <Ctrl>, <Alt> and <Delete> keys. If you do not press the keys at the correct time and the system does not boot, an error message will be displayed and you will again be asked to

Press to enter Setup; press < F7> to enter pop-up Boot menu.

3-10 Boot Menu



Boot Configuration

Setup Prompt Timeout

Use this item to set number of seconds to wait for setup activation key.

Bootup Numlock State

Use this item to select keyboard numlock state.

The optional settings are: [On]; [Off].

Quiet Boot

The optional settings are: [Disabled]; [Enabled].

Boot Option Priorities

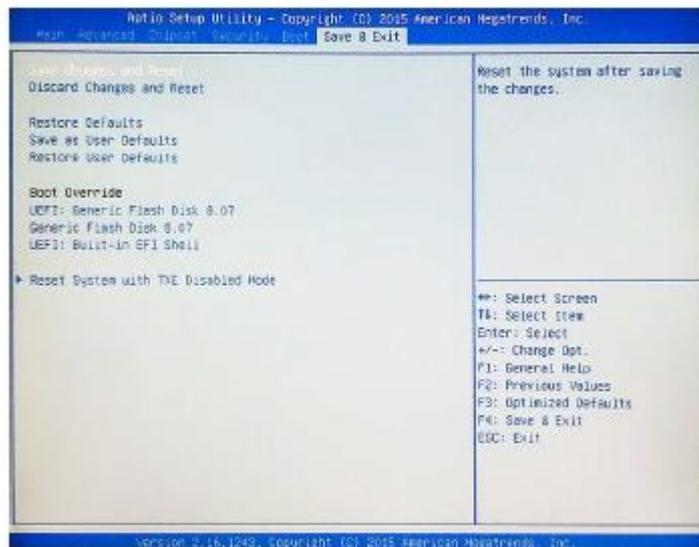
Boot Option #1/ Boot Option #2...

Use this item to decide system boot order from available options.

Hard Drive BBS Priorities

Use this item to set the system boot order of the legacy device in the group.

3-11 Save & Exit Menu



Save Changes and Reset

This item allows user to reset the system after saving the changes.

Discard Changes and Reset

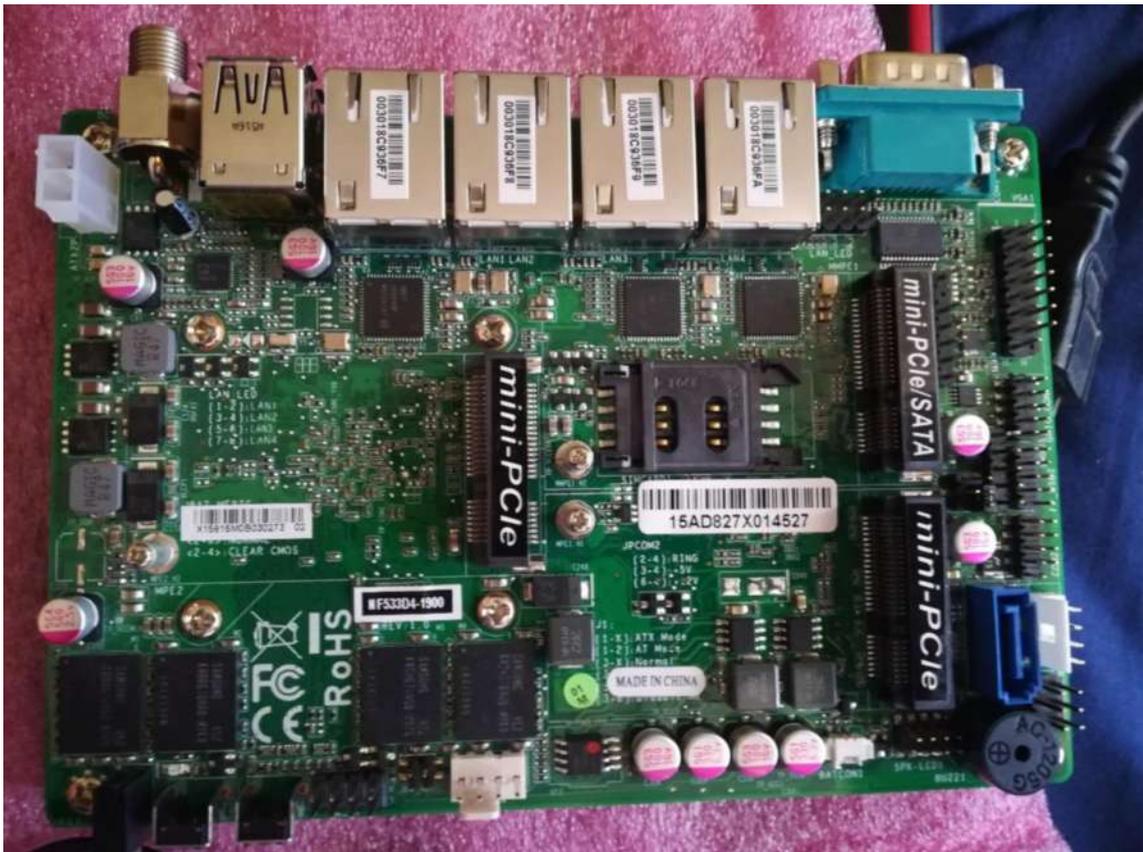
This item allows user to reset the system without saving any changes.

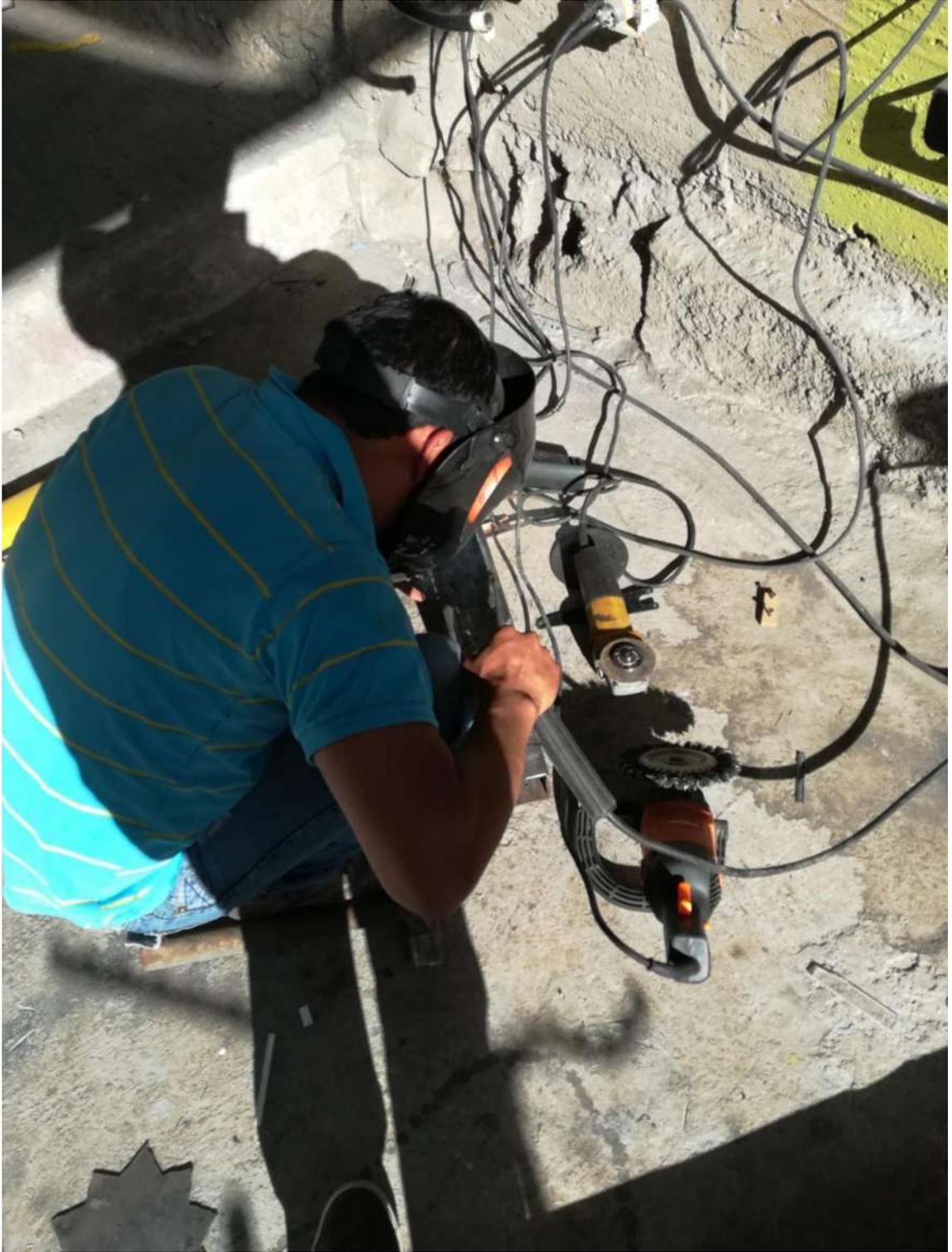
Restore Defaults

Use this item to restore /load default values for all the setup options.

Save as User Defaults

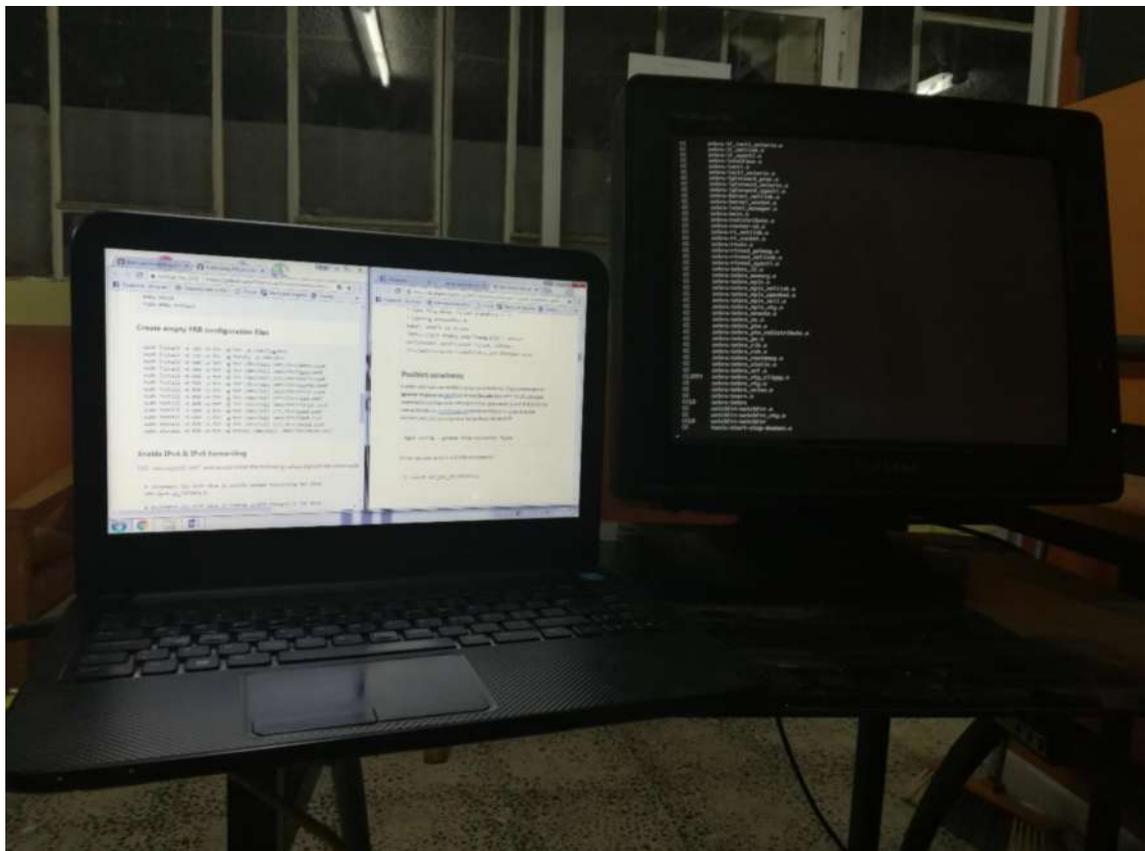
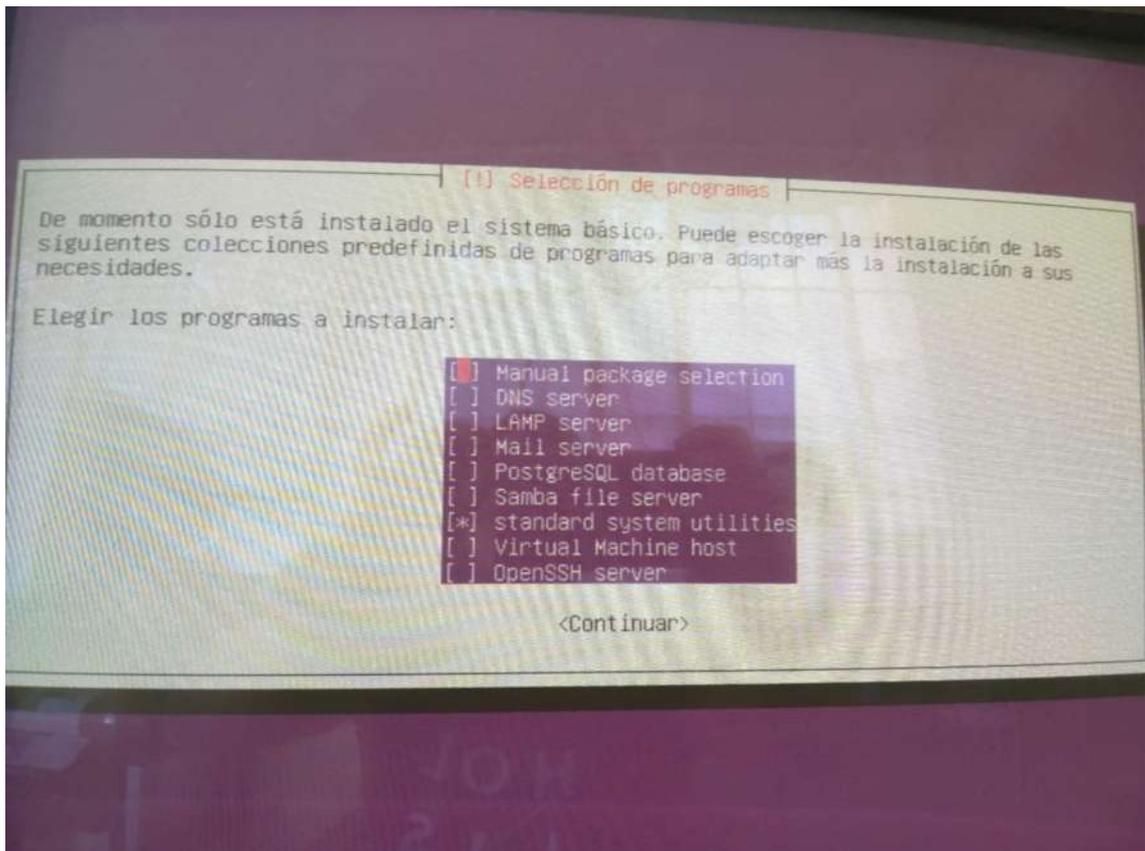
Anexo B: Ensamblaje







Anexo C: Instalación



SyncMaster 793s

```
feb 11 11:46:06 Jetway watchfrr[1561]: nhrpd state -> up : connect succeeded
feb 11 11:46:06 Jetway watchfrr[1561]: ospfd state -> up : connect succeeded
feb 11 11:46:06 Jetway watchfrr[1561]: zebra state -> up : connect succeeded
feb 11 11:46:07 Jetway watchfrr[1561]: ldpd state -> up : connect succeeded
feb 11 11:46:07 Jetway watchfrr[1561]: ripd state -> up : connect succeeded
feb 11 11:46:07 Jetway frr[1441]: Starting Frr monitor daemon: watchfrr.
feb 11 11:46:07 Jetway frr[1441]: Exiting from the script
feb 11 11:46:07 Jetway systemd[1]: Started Cumulus Linux FRR.
```

lines 1-34/34 (END)

```
root@Jetway:~/frr#
```

```
root@Jetway:~/frr#
```

```
root@Jetway:~/frr# systemctl status frr
```

```
• frr.service - Cumulus Linux FRR
```

```
Loaded: loaded (/etc/systemd/system/frr.service; enabled; vendor preset: enabled)
```

```
Active: active (running) since jue 2016-02-11 11:46:07 ECT; 3min 14s ago
```

```
Process: 1441 ExecStart=/usr/lib/frr/frr start (code=exited, status=0/SUCCESS)
```

```
Tasks: 15
```

```
Memory: 26.6M
```

```
CPU: 1.154s
```

```
CGroup: /system.slice/frr.service
```

```
├─1476 /usr/lib/frr/zebra -s 90000000 --daemon -A 127.0.0.1
```

```
├─1483 /usr/lib/frr/bgpd --daemon -A 127.0.0.1
```

```
├─1490 /usr/lib/frr/ripd --daemon -A 127.0.0.1
```

```
├─1497 /usr/lib/frr/ripngd --daemon -A ::1
```

```
├─1504 /usr/lib/frr/ospfd --daemon -A 127.0.0.1
```

```
├─1511 /usr/lib/frr/ospf6d --daemon -A ::1
```

```
├─1518 /usr/lib/frr/isisd --daemon -A 127.0.0.1
```

```
├─1525 /usr/lib/frr/babeld --daemon -A 127.0.0.1
```

```
├─1532 /usr/lib/frr/pimd --daemon -A 127.0.0.1
```

```
├─1539 /usr/lib/frr/ldpd -L
```

```
├─1540 /usr/lib/frr/ldpd -E
```

```
├─1541 /usr/lib/frr/ldpd --daemon -A 127.0.0.1
```

```
├─1548 /usr/lib/frr/nhrpd --daemon -A 127.0.0.1
```

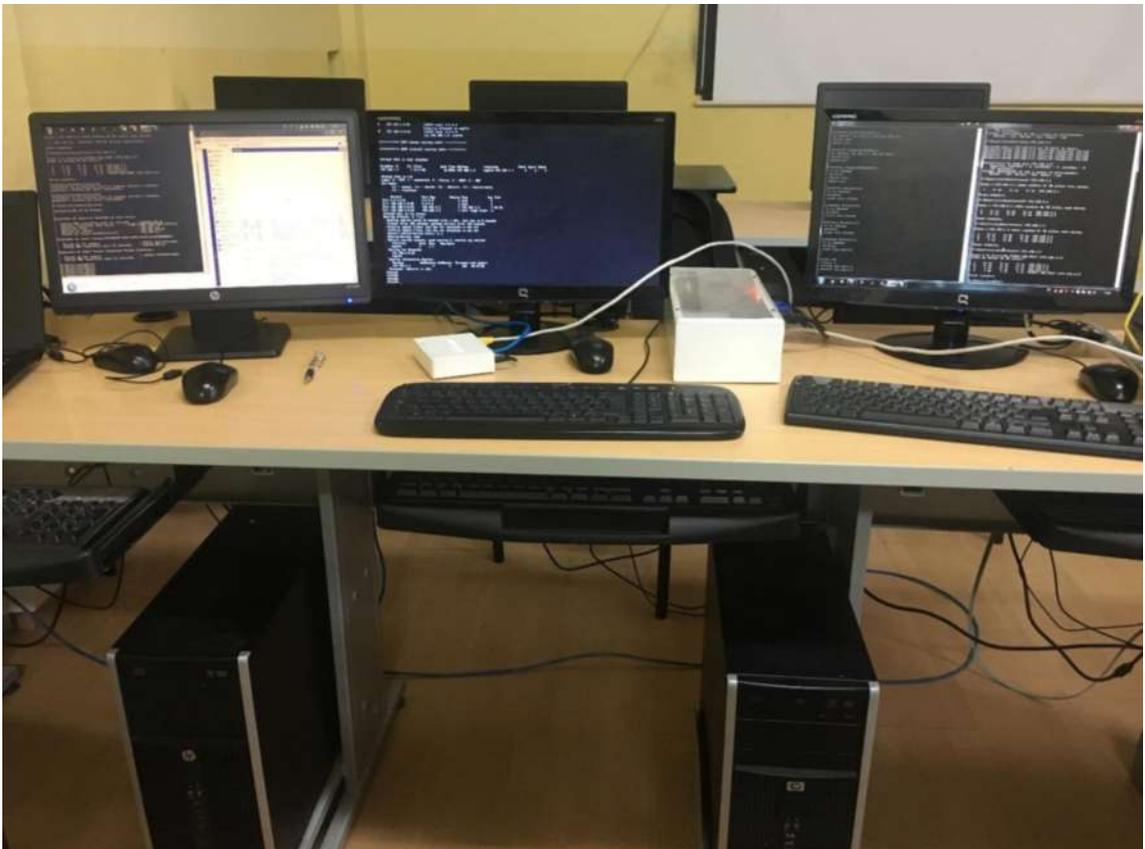
```
├─1555 /usr/lib/frr/eigrpd --daemon -A 127.0.0.1
```

```
└─1561 /usr/lib/frr/watchfrr -d -r /usr/sbin/servicebBfrrbBrestartbB:s -s /u
```

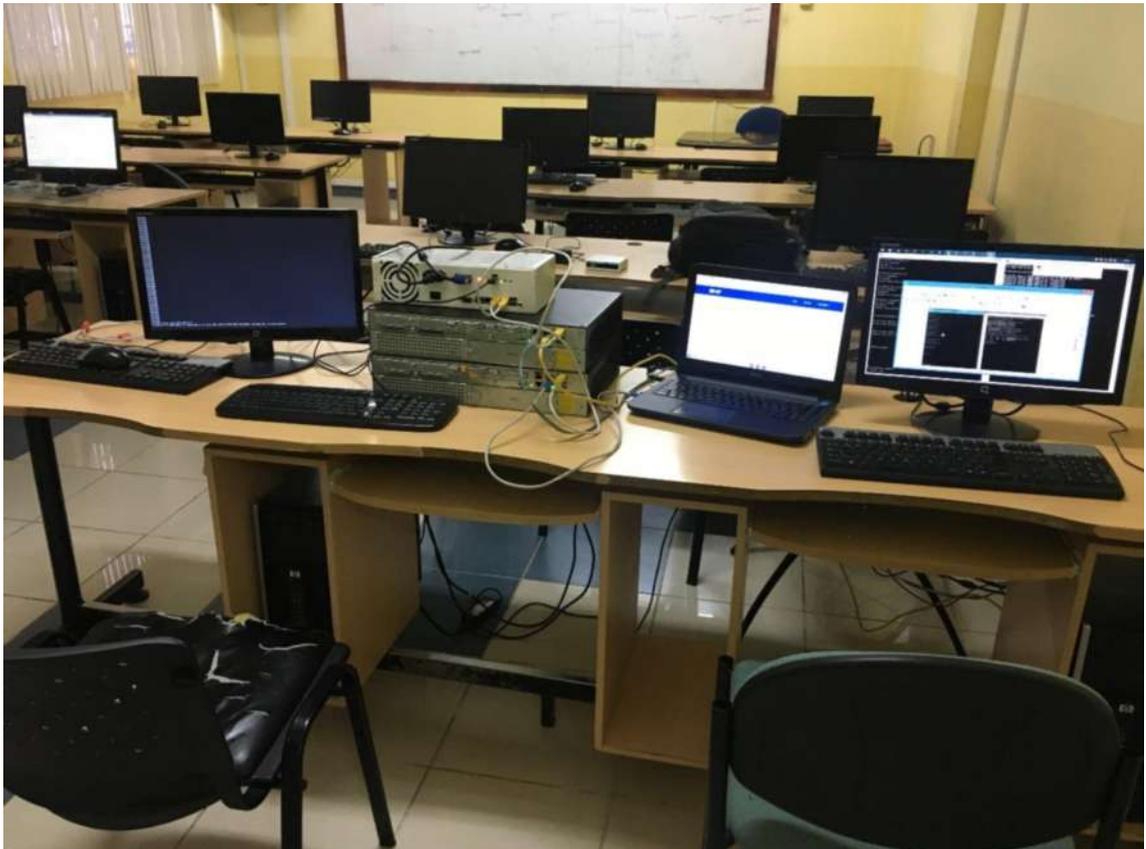
```
feb 11 11:46:06 Jetway watchfrr[1561]: bgpd state -> up : connect succeeded
feb 11 11:46:06 Jetway watchfrr[1561]: nhrpd state -> up : connect succeeded
feb 11 11:46:06 Jetway watchfrr[1561]: ospfd state -> up : connect succeeded
feb 11 11:46:06 Jetway watchfrr[1561]: zebra state -> up : connect succeeded
feb 11 11:46:06 Jetway watchfrr[1561]: ldpd state -> up : connect succeeded
feb 11 11:46:07 Jetway watchfrr[1561]: ripd state -> up : connect succeeded
feb 11 11:46:07 Jetway watchfrr[1561]: Watchfrr: Notifying Systemd we are up and running
feb 11 11:46:07 Jetway frr[1441]: Starting Frr monitor daemon: watchfrr.
feb 11 11:46:07 Jetway frr[1441]: Exiting from the script
feb 11 11:46:07 Jetway systemd[1]: Started Cumulus Linux FRR.
```

lines 1-34/34 (END)

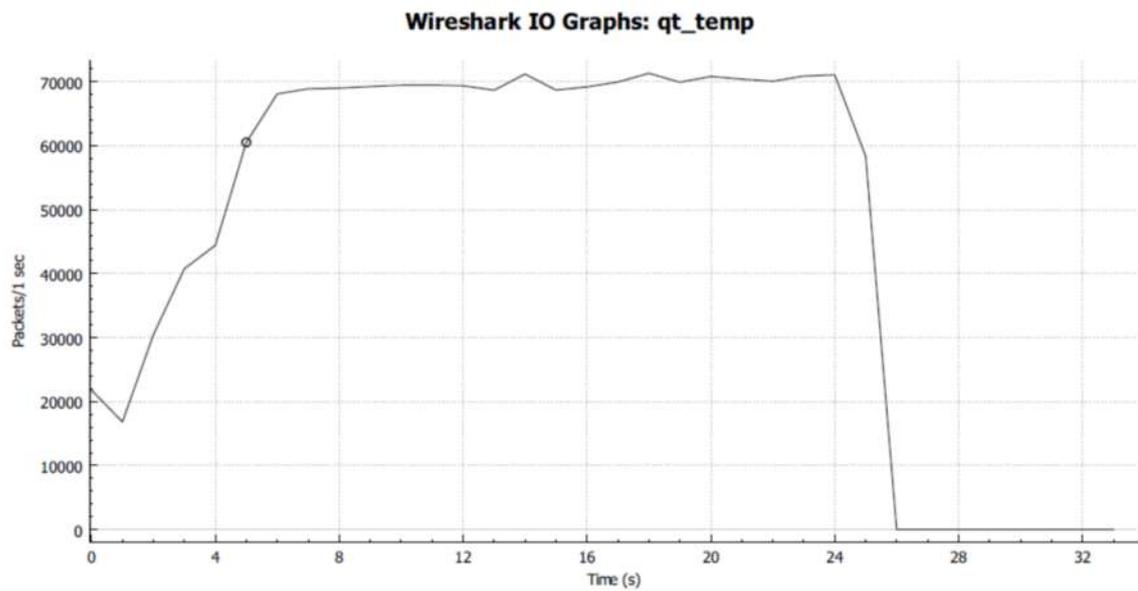
Anexo D: Interoperabilidad entre equipos







Anexo E: Datos relevantes



```
Use command at the base level
jan/02/1970 00:00:13 system,error,critical router was rebooted without proper shut
down
jan/02/1970 00:00:14 system,error,critical router was rebooted without proper shut
down
jan/02/1970 00:00:14 system,error,critical router was rebooted without proper shut
down
[admin@MikroTik] >
[admin@MikroTik] >
[admin@MikroTik] > ip address add address=192.168.1.2/24
interface: ether5
[admin@MikroTik] > system identity set name=mikrotik
syntax error (line 1 column 26)
[admin@mikrotik] > system identity set name=mikrotik
[admin@mikrotik] > routing ospf network add network=192.168.1.0/30 area=0
input does not match any value of area
[admin@mikrotik] > routing ospf network add network=192.168.1.0/30 area=backbone
[admin@mikrotik] >
```

t.tif

sesion: 03/10/2017 9:49

COMPAQ

```
root@Jetuay:/home/jetuay#  
root@Jetuay:/home/jetuay# cat /proc/meminfo  
MemTotal: 3923396 kB  
MemFree: 3362268 kB  
MemAvailable: 3357164 kB  
Buffers: 18232 kB  
Cached: 223636 kB  
SwapCached: 0 kB  
Active: 300756 kB  
Inactive: 173148 kB  
Active(anon): 235076 kB  
Inactive(anon): 48564 kB  
Active(file): 65680 kB  
Inactive(file): 124584 kB  
Unevictable: 3660 kB  
Mlocked: 3660 kB  
SwapTotal: 4070396 kB  
SwapFree: 4070396 kB  
Dirty: 380 kB  
Writeback: 0 kB  
AnonPages: 235772 kB  
Mapped: 70484 kB  
Shmem: 49124 kB  
Slab: 38264 kB  
SReclaimable: 18796 kB  
SUnreclaim: 19468 kB  
KernelStack: 3072 kB  
PageTables: 6880 kB  
NFS_Unstable: 0 kB  
Bounce: 0 kB  
WritebackTmp: 0 kB  
CommitLimit: 6032092 kB  
Committed_AS: 972528 kB  
UnallocTotal: 34359738367 kB  
UnallocUsed: 0 kB  
UnallocChunk: 0 kB  
HardwareCorrupted: 0 kB  
AnonHugePages: 157696 kB  
CmaTotal: 0 kB  
CmaFree: 0 kB  
HugePages_Total: 0  
HugePages_Free: 0  
HugePages_Rsvd: 0  
HugePages_Surp: 0  
Hugepagesize: 2048 kB  
DirectMap4k: 77812 kB  
DirectMap2M: 3993600 kB  
root@Jetuay:/home/jetuay#
```