

# **ESCUELA POLITÉCNICA NACIONAL**

## **ESCUELA DE FORMACIÓN DE TECNÓLOGOS**

### **IMPLEMENTACIÓN DE PRÁCTICAS DE LABORATORIO DE ROUTING MEDIANTE *LINUX* BASADO EN SOFTWARE DE ROUTING LIBRE ADICIONANDO UN MÓDULO RASPBERRY PI PARA LAS ASIGNATURAS DE REDES DE COMPUTADORAS (TRTR424) Y TCP/IP (TRTR314)**

#### **TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN ELECTRÓNICA Y TELECOMUNICACIONES**

**Angelo Salvatore Orellana Pinzón**

angelo.orellana@epn.edu.ec

**Pablo Andrés Puma Ushiña**

pablo.puma@epn.edu.ec

**DIRECTOR: ING. FERNANDO VINICIO BECERRA CAMACHO, MSc.**

fernando.becerrac@epn.edu.ec

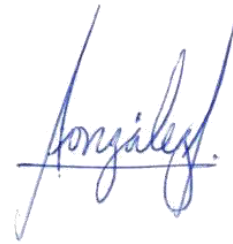
**CODIRECTOR: ING. FABIO MATÍAS GONZÁLEZ GONZÁLEZ, MSc.**

fabio.gonzalez@epn.edu.ec

**Quito, Enero 2022**

# CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por los Srs. Angelo Salvatore Orellana Pinzón y Pablo Andrés Puma Ushiña como requerimiento parcial a la obtención del título de TECNÓLOGOS EN ELECTRÓNICA Y TELECOMUNICACIONES, bajo nuestra supervisión:



---

**Fernando Vinicio Becerra  
Camacho**

DIRECTOR DEL PROYECTO

---

**Fabio Matías González  
González**

CODIRECTOR DEL PROYECTO

## DECLARACIÓN

Nosotros Orellana Pinzón Angelo Salvatore con CI: 1725279689 y Puma Ushiña Pablo Andrés con CI: 1721851648 declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que hemos consultado las referencias bibliográficas que se incluyen en este documento.

Sin perjuicio de los derechos reconocidos en el primer párrafo del artículo 144 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación – COESC-, somos titulares de la obra en mención y otorgamos una licencia gratuita, intransferible y no exclusiva de uso con fines académicos a la Escuela Politécnica Nacional.

Entregamos toda la información técnica pertinente, en caso de que hubiese una explotación comercial de la obra por parte de la EPN, se negociará los porcentajes de los beneficios conforme lo establece la normativa nacional vigente.



---

**Angelo Salvatore Orellana Pinzón**



---

**Pablo Andrés Puma Ushiña**

## **DEDICATORIA**

Dedico este trabajo de titulación para mi mamá y mi papá, quienes se esforzaron día con día para darme educación superior en la mejor universidad del país y que cuando vieron que las cosas no estaban en el sentido que debían me aconsejaron y guiaron para poder culminar mis estudios y tener un soporte para poder entrar al mundo laboral y sobre todo crecer como persona, les dedico cada palabra, cada signo, cada concepto ya que en este trabajo se refleja lo aprendido en la Escuela de Formación de Tecnólogos.

Aunque ha sido un reto muy largo y extenuante sepan que siempre di mi mejor esfuerzo todos estos años y no tome ningún atajo, gracias a ustedes que me demostraron que nada en la vida que valga la pena va a ser fácil.

Angelo

# **AGRADECIMIENTO**

Agradecer a todas las personas que han estado este largo camino, en especial a mis padres y hermanos que siempre supieron darme un consejo y ánimo en la duración de mis estudios en la Escuela de Formación de Tecnólogos, los recursos invertidos en mi servirán de ahora en adelante para poder afrontar el campo laboral que está lleno de retos y dificultades, que siempre serán una constante en esta vida, pero que son necesarios para formar el carácter y la personalidad.

Agradecer también a cada Ingeniero y colaborador de la Escuela Politécnica Nacional quienes fueron parte importantísima en el desarrollo de toda mi etapa universitaria ya que sin ellos no hubiera sido posible aprender y entender los conceptos y teoría para tener un panorama claro al aplicarlos en los laboratorios de ley.

Angelo

## **DEDICATORIA**

Se la dedico a mi familia ya que han sido un apoyo muy importante durante el transcurso de mi carrera, pero en especial se lo dedico a mis padres los cuales han sido el pilar fundamental para conseguir muchos de mis logros incluyendo este.

Pablo

# AGRADECIMIENTO

En primer lugar, agradezco a Dios por la salud y las bendiciones que ha hecho llegar a mi vida para así poder culminar esta etapa muy importante de mis estudios.

Agradezco a mis padres que con su apoyo moral e incondicional han aportado en un alto porcentaje a mis ganas de seguir adelante en mi carrera profesional y no decaer en ningún momento.

A la Escuela Politécnica Nacional por haberme aceptado ser parte de ella y brindarme la oportunidad de convertirme en un profesional, a cada docente que con su gran sabiduría y experiencia compartida hizo que todo este proceso de formación no tan sencillo valga la pena, además por ayudarme a llegar al punto en el cual me encuentro ahora.

Pablo

# ÍNDICE DE CONTENIDOS

1	INTRODUCCIÓN .....	1
1.1	Objetivo general.....	2
1.2	Objetivos específicos .....	2
1.3	Fundamentos .....	2
2	METODOLOGÍA.....	9
2.1	Descripción de la metodología usada .....	9
3	RESULTADOS Y DISCUSIÓN .....	11
3.1	Identificación de los procesos que sirven para la instalación y manejo de máquinas virtuales en <i>Linux</i> . .....	11
3.2	Análisis de los parámetros fundamentales del <i>software Quagga</i> y sus funcionalidades. ....	14
3.3	Determinación de los tipos de protocolos que maneja el <i>software</i> de <i>routing</i> virtual <i>Quagga</i> .....	17
3.4	Diseño de prácticas de laboratorio acorde al contenido teórico dictado en las asignaturas de Redes de Computadoras (TRTR424) y TCP/IP (TRTR314) de la Escuela de Formación de Tecnólogos aplicados a un <i>software</i> virtual y módulos físicos sobre <i>Raspberry Pi</i> . ....	20
3.5	Verificación del funcionamiento del sistema. ....	25
4	CONCLUSIONES Y RECOMENDACIONES .....	29
4.1	Conclusiones.....	29
4.2	Recomendaciones .....	30
5	REFERENCIAS BIBLIOGRAFICAS .....	32
	ANEXOS.....	35
	Anexo 1: Certificado de Funcionamiento .....	i
	Anexo 2: Prácticas de Laboratorio .....	iii



## ÍNDICE DE FIGURAS

<b>Figura 1.1</b> Partes de la <i>Raspberry Pi 4</i> .....	4
<b>Figura 1.2</b> Estructura de una dirección IPv4.....	5
<b>Figura 1.3</b> Estructura de una dirección IPv6.....	5
<b>Figura 3.1</b> Esquema de <i>hipervisor</i> de Tipo 1 .....	12
<b>Figura 3.2</b> Esquema de <i>hipervisor</i> de Tipo 2 .....	13
<b>Figura 3.3</b> Creación de Máquina Virtual.....	14
<b>Figura 3.4</b> Máquina Virtual sobre entorno <i>Windows</i> .....	14
<b>Figura 3.5</b> Arquitectura de <i>Quagga</i> .....	15
<b>Figura 3.6</b> Protocolos de enrutamiento soportado en <i>Quagga Routing Suite</i> .....	16
<b>Figura 3.7</b> Asignación de direcciones IPv4 en <i>router Quagga</i> .....	16
<b>Figura 3.8</b> Tabla de enrutamiento del <i>router Quagga</i> .....	17
<b>Figura 3.9</b> Interfaces operativas de <i>router Quagga</i> .....	17
<b>Figura 3.10</b> Demonio Central <i>Zebra</i> en estado activo.....	18
<b>Figura 3.11</b> Demonio <i>bgpd</i> en estado activo. ....	18
<b>Figura 3.12</b> Demonio <i>ospf6d</i> en estado activo. ....	19
<b>Figura 3.13</b> Demonio <i>ospf6d</i> en estado activo. ....	19
<b>Figura 3.14</b> Demonio <i>ripd</i> en estado activo. ....	19
<b>Figura 3.15</b> Demonio <i>ripngd</i> en estado activo. ....	20
<b>Figura 3.16</b> Demonio <i>isisd</i> en estado activo. ....	20
<b>Figura 3.17</b> <i>Ping</i> de <i>Router 1</i> a <i>Router 2</i> .....	25
<b>Figura 3.18</b> <i>Ping</i> de <i>Router 2</i> a <i>Router 1</i> .....	25
<b>Figura 3.19</b> Tabla de enrutamiento RIP del <i>Router 1</i> .....	25
<b>Figura 3.20</b> Tabla de enrutamiento RIP del <i>Router 2</i> .....	26
<b>Figura 3.21</b> <i>Ping</i> físico de módulo <i>Raspberry 1</i> a <i>Raspberry 2</i> .....	27
<b>Figura 3.22</b> <i>Ping</i> físico del módulo <i>Raspberry 1</i> a interfaz de salida del módulo <i>Raspberry 2</i> .....	27
<b>Figura 3.23</b> <i>Ping</i> físico de módulo <i>Raspberry 2</i> a <i>Raspberry 1</i> .....	28
<b>Figura 3.24</b> <i>Ping</i> físico del módulo <i>Raspberry 2</i> a interfaz de salida del módulo <i>Raspberry 1</i> .....	28

## RESUMEN

El presente proyecto de titulación se enfoca en el análisis del *software* de licencia libre para *routing*, denominado “*Quagga*”. Las bases de su implementación, principalmente en el entorno virtual (con la creación y configuración de máquinas virtuales), las características esenciales de su funcionamiento, los protocolos de enrutamiento que soporta tanto en direcciones IPv4 e IPv6 y el desarrollo de prácticas de laboratorio que apliquen dichos protocolos en topologías de red, son parte de todo este informe.

La primera sección del proyecto se enfoca en los parámetros técnicos para la comprensión en la aplicación de este, se tratan conceptos como fundamentos de *routing*, direcciones de red, protocolos de enrutamiento y demás temas que proporcionan un mejor contexto para la lectura del presente documento.

La segunda sección se centra en la metodología aplicada para la elaboración de las hojas guía de laboratorio, así como la temática de cada uno de los objetivos planteados en el proyecto de titulación.

La tercera sección trata sobre el análisis de los resultados, todo se explica en base a la convergencia de los objetivos del trabajo de titulación. El desarrollo de las prácticas de laboratorio con los protocolos disponibles en *Quagga*, así como su respectiva comprobación con una validación de extremo a extremo.

La cuarta sección abarca las conclusiones obtenidas durante la planeación y ejecución del proyecto y las recomendaciones para que tenga una mayor operatividad o ampliar su campo de acción.

La última sección contiene toda la información bibliográfica con la cual el desarrollo del proyecto de titulación fue posible.

**PALABRAS CLAVE:** *Quagga*, virtualización, protocolos, enrutamiento, *zebra*, *Ubuntu*.

## **ABSTRACT**

*This degree project focuses on the analysis of free license software for routing, called "Quagga", the bases of its implementation mainly in the virtual environment (with the creation and configuration of virtual machines), the essential characteristics of its operation, the routing protocols that supports both IPv4 and IPv6 addresses, the development of virtual laboratory practices that apply these protocols with network topologies are part of this whole study.*

*The first section of the project focuses on the technical parameters for compression in the application of this, concepts such as routing fundamentals, network addresses, routing protocols and other topics that provide a better context for reading this document are discussed.*

*The second section focuses on the methodology applied for the elaboration of the laboratory guide sheets, as well as the theme of each of the objectives set out in the degree project.*

*The third section deals with the analysis of the results, everything is explained based on the convergence of the objectives, the development of the laboratory practices, with the protocols available in Quagga, as well as their respective verification with an end-to-end validation.*

*The fourth section covers the conclusions obtained during the planning and execution of the project and the recommendations to make it more operational or expand its field of action.*

*The last section contains all the bibliographic information with which the development of the degree project was possible.*

**KEYWORDS:** *Quagga, virtualization, protocols, routing, zebra, Ubuntu.*

# 1 INTRODUCCIÓN

La implementación de redes de datos de forma virtual se ha convertido en una solución óptima, tanto para los ambientes ya sean educativos, profesionales o empresariales pequeños y medianos (PYMES). Sectores estratégicos en los cuales no se cuentan con los equipos físicos para desarrollar dichas inversiones y montaje de infraestructura de red.

El uso de *software* libre de *routing* es la base de esta solución, la cual tiene como objetivo permitir el manejo de equipos de *networking*. El presente proyecto de titulación se enfoca en la implementación de prácticas de laboratorio para las asignaturas de Redes de Computadoras (TRTR424) y TCP/IP (TRTR314) de la carrera de Tecnología Superior en Redes y Telecomunicaciones.

Dichas prácticas de laboratorio se realizan bajo una plataforma de *routing* en el Sistema Operativo *Linux*. Inicialmente, planteando el diseño de una red, con su respectiva topología y en la cual se apliquen los protocolos de enrutamiento, sean estos estáticos o dinámicos, generando dicha red, usando máquinas virtuales.

Las prácticas diseñadas se implementarán físicamente con módulos *Raspberry Pi 4* los cuales se les puede instalar cualquier distribución de licencia libre disponible en su librería. La distribución por utilizar puede ser la nativa de la tarjeta inteligente (*Raspbian OS*) o alguna versión del *Kernel* de *Linux*, como *Ubuntu* o *Debian*.

La implementación se completa con la respectiva configuración de los *routers*, el uso de *patchcords* para la conexión de dichas tarjetas, replicando la topología propuesta y la debida verificación de funcionamiento.

La falta de equipos físicos de laboratorio (*routers* y *switches*) en la ESFOT para realizar prácticas en base a redes de datos, se torna un problema, ya que no se puede hacer notar las diferencias entre la implementación virtual y física. A todo esto, es importante que el estudiante analice las diferencias en ambos escenarios para un mejor aprendizaje.

De esta manera se logrará dar un apoyo a los estudiantes de modo que tengan una mejor perspectiva del funcionamiento de las máquinas virtuales, así como de los tipos de enrutamiento que se pueden manejar dentro de las redes de computadoras.

Además, la implementación física será un incentivo en el aprendizaje de los módulos *Raspberry Pi* y la semejanza con un equipo de enrutamiento comercial.

## 1.1 Objetivo general

Implementar prácticas de laboratorio de *routing* mediante *Linux* basado en *software* de *routing* libre adicionando un módulo *Raspberry Pi* para las asignaturas de Redes de Computadoras (TRTR424) Y TCP/IP (TRTR314).

## 1.2 Objetivos específicos

- Identificar los procesos que sirven para la instalación y manejo de máquinas virtuales en *Linux*.
- Analizar los parámetros fundamentales del *software Quagga* y sus funcionalidades.
- Determinar los tipos de protocolos que maneja el *software* de *routing* virtual *Quagga*.
- Diseñar prácticas de laboratorio acorde al contenido teórico dictado en las asignaturas de Redes de Computadoras (TRTR424) y TCP/IP (TRTR314) de la Escuela de Formación de Tecnólogos aplicados a un *software* virtual y un módulo físico sobre *Raspberry Pi*.
- Verificar y probar el funcionamiento del sistema.

## 1.3 Fundamentos

### Imagen ISO

Es un archivo especial que incluye el contenido exacto de un CD, DVD y se utiliza en la mayoría de los casos para la creación de máquinas virtuales de sistemas operativos tales como *Windows*, *Mac* o *Linux*, aunque también se puede implementar físicamente en un equipo que sea compatible y tenga la capacidad de ejecución, los archivos ISO tienen la extensión (.iso) [1].

### RASPBERRY PI 4

La *Raspberry Pi 4* es el último producto de la familia *Raspberry Pi*, con un procesador *quad-core* de 64 bits con 1.5 (GHz), Red inalámbrica de banda dual 2,4/5 GHz, *Bluetooth 5 / BLE*, *Ethernet* más rápido (*Gigabit* y capacidad *PoE* a través de un *PoE HAT* separado) [2], [3].

La tarjeta inteligente presenta las siguientes características operativas y de funcionamiento:

- *Broadcom* BCM2711, *Cortex* núcleo cuádruple-A72 (ARM v8) 64-bit SoC @ 1,5 GHz
- SDRAM LPDDR4-2400 de 1 GB, 2 GB, 4 GB y 8 GB (según el modelo)
- IEEE 802.11ac inalámbrico 2.4 (GHz) y 5.0 (GHz), *Bluetooth* 5.0, BLE
- *Gigabit Ethernet*
- 2 puertos USB 3.0; 2 puertos USB 2.0.
- Cabezal GPIO estándar de 40 pines de *Raspberry Pi* (totalmente compatible con las placas anteriores)
- 2 × puertos micro-HDMI (soportados hasta 4kp60)
- Puerto de pantalla MIPI DSI de 2 vías
- Puerto de cámara MIPI CSI de 2 carriles
- Puerto de audio estéreo de 4 polos y de vídeo compuesto
- H.265 (decodificación 4kp60), H264 (decodificación 1080p60, decodificación 1080p30)
- Gráficos *OpenGL* ES 3.0
- Ranura para tarjetas Micro-SD para cargar el sistema operativo y el almacenamiento de datos
- 5 V DC a través de conector USB-C (mínimo 3 A)
- 5 V DC vía cabezal GPIO (mínimo 3 A)
- Alimentación a través de *Ethernet (PoE)* habilitada (requiere *PoE HAT* separado)
- Temperatura de funcionamiento: 0 °C – 50 °C ambiente
- LAN inalámbrica de doble banda que viene con certificación de cumplimiento modular, lo que permite que la placa se diseñe en productos finales con pruebas de cumplimiento de LAN inalámbrica significativamente reducidas, lo que mejora tanto el costo como el tiempo de comercialización [2], [3].

La distribución de los componentes de la tarjeta física se puede observar en la Figura 1.1.

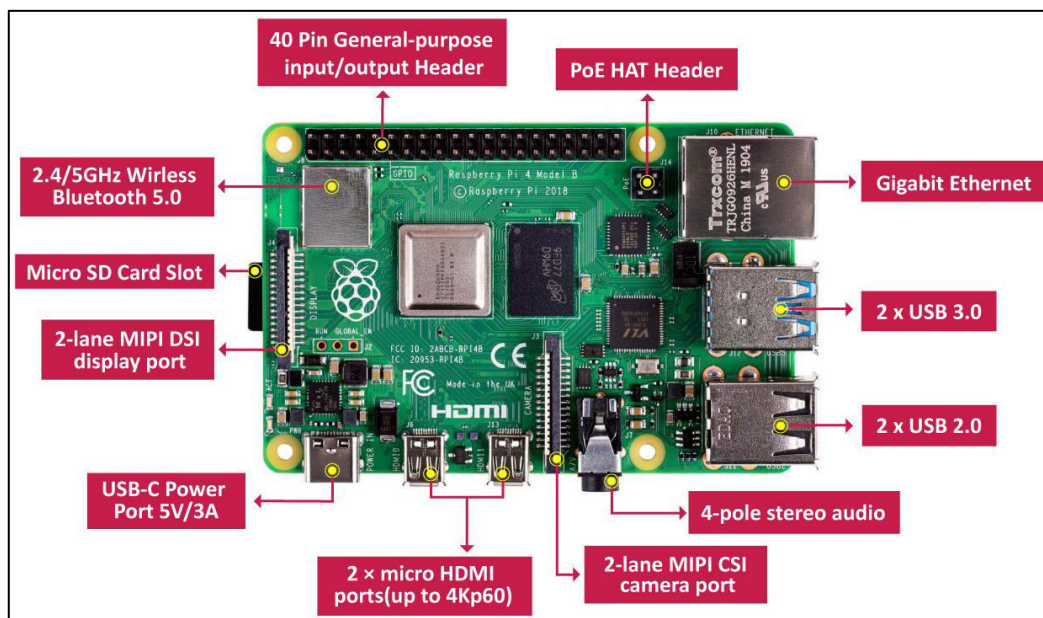


Figura 1.1 Partes de la *Raspberry Pi 4* [2], [3]

## DIRECCIONAMIENTO IP

El direccionamiento tiene una función clave en los protocolos de capa de red que permite la transmisión de datos entre hosts de la misma red o en redes diferentes. El diseño, implementación y administración de un plan de direccionamiento IPv4 efectivo asegura que las redes puedan operar de manera eficaz y eficiente [4].

Una dirección IP (Protocolo de Internet) es como una dirección de una calle o como un número de teléfono. Cuando un dispositivo se conecta a Internet se asigna una dirección IP y al ingresar a cualquier sitio este también tendrá una dirección IP [4].

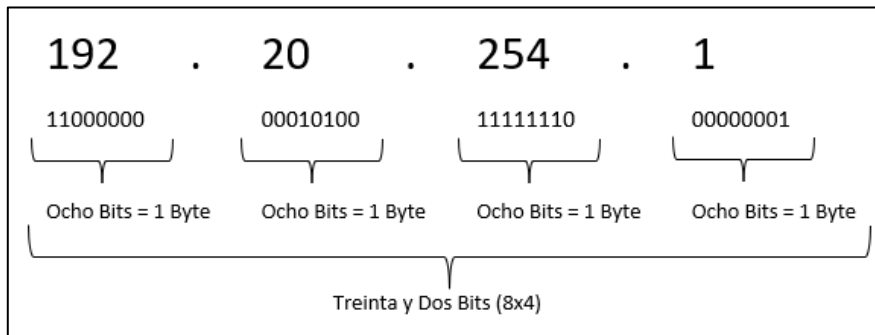
Las direcciones IPv4, sus siglas quieren decir Protocolo de Internet en versión 4, son direcciones de red es decir que nos van a ubicar dentro de un entramado de redes de computadoras, además ofrece direccionamiento jerárquico para paquetes que transportan datos [4].

Hoy en día las direcciones IPv4 se están reemplazando por IPv6, pero este reemplazo todavía no se lo integra completamente ya que faltan varios años para que se puedan incorporar del todo [5].

Ambas trabajan con lo que se denomina código binario esto implica que se van a emplear dos dígitos cero (0) y uno (1). El código en IPv4 se representa a través de una

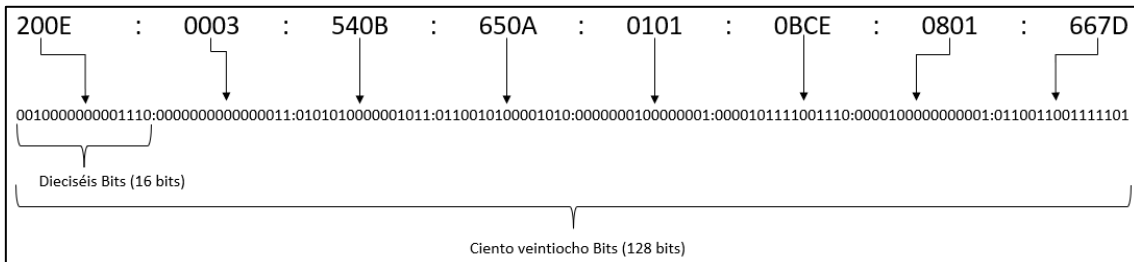
lista de 32 bits los cuales están subdivididos en 4 campos llamados octetos, es decir, toman 8 bits por grupo [4], [5].

Cada octeto no se representa con una notación binaria, sino que se van a transformar en notación decimal separadas por un punto entre cada grupo de 8 bits como se muestra en la Figura 1.2 [4].



**Figura 1.2** Estructura de una dirección IPv4

En tanto que las direcciones IPv6 (Protocolo de Internet en versión 6) se encuentran conformadas por una lista de 128 bits los cuales están subdivididos en grupos de 16 bits, en donde cada grupo se representa con una notación hexadecimal separadas por 2 puntos (:) como se muestra en la Figura 1.3 [6].



**Figura 1.3** Estructura de una dirección IPv6

## ENRUTAMIENTO DE REDES

El enrutamiento es el proceso de reenviar paquetes entre redes de datos, siempre buscando la mejor ruta (la más corta). Para encontrar esa ruta óptima, se debe tener en cuenta la tabla de enrutamiento y algunos otros parámetros como la métrica, la distancia administrativa, el ancho de banda entre otros parámetros fundamentales [7].



## MÉTRICAS EN PROTOCOLOS DE ENRUTAMIENTO

Un protocolo de enrutamiento tiene datos o información de más de una ruta para llegar al mismo destino y para ello tiene que elegir el mejor camino. Cada protocolo de enrutamiento tiene su propio algoritmo para resolver el problema de aprender todas las rutas, elegir la mejor ruta basado en la métrica y converger en reacción a los cambios ocurridos en la red [7].

Una métrica se define como un valor que el protocolo de enrutamiento utiliza para asignar a las distintas rutas con la finalidad de llegar a las redes remotas eligiendo el mejor camino [7].

### DISTANCIA ADMINISTRATIVA

Todos los protocolos de enrutamiento tienen un parámetro llamado distancia administrativa (AD) que es la confiabilidad del protocolo, este valor permite decidir cuál ruta es la más confiable y mientras menor sea este valor, más confiable es la ruta [8], [9].

La distancia administrativa se encuentra en el rango de 0 a 255, es decir, que el valor de 255 es la métrica menos confiable en la que el *router* no optará por establecer dicha ruta [8], [9].

## PROTOCOLOS DE ENRUTAMIENTO

Los protocolos de enrutamiento dinámico se clasifican en dos, los protocolos de *Gateway* interior (IGP) y los protocolos de *Gateway* exterior (EGP), de hecho, en los EGP solo hay uno que es BGP, el cual es compatible con IPv4 e IPv6 [8].

Dentro de los IGP hay dos grupos:

- Protocolos de vector distancia en donde se encuentra RIP.
- Protocolos de estado de enlace en donde se encuentra OSPF y IS-IS.

### RIP (*Routing Information Protocol*)

RIP es un protocolo de vector distancia lo que significa que solo le interesa cómo llegar a un destino basándose en la cantidad de saltos por donde atravesase menos equipos. El algoritmo que utiliza para determinar cuál es la ruta más cercana hacia un destino es el algoritmo *Bellman-Ford* [7], [8], [9], [10], [11].

Es un estándar libre lo que significa que puede correr en cualquier tipo de equipo, no solo en equipos *Cisco* sino en cualquier otro tipo de otro fabricante. Soporta el protocolo

IP, a nivel de capa de transporte tiene el protocolo UDP y se le asigna el puerto 520 [7], [8], [9], [10], [11].

La distancia administrativa de este protocolo es 120, en tanto que su métrica es el conteo de saltos en donde el máximo es de 15 saltos. RIP es un protocolo que no utiliza números de áreas que determinen algún tipo de unidad administrativa [7].

### **RIPng (*Routing Information Protocol new generation*)**

RIPng soporta este protocolo que está descrito en la RFC 2080, quiere decir que es una adaptación del protocolo RIP a IPv6. Este protocolo es el sucesor de RIP versión 2 pero sigue basándose en la técnica de vector distancia lo cual tiene problemas de convergencia [12], [13].

RIPng permite intercambiar datos sobre redes en IPv6, es decir, utiliza direccionamiento en IPv6 para el transporte de información. Para enviar las actualizaciones a este protocolo se le asigna el puerto 521 del protocolo UDP [12], [13].

Al igual que RIPv2 su métrica es el número de saltos el cual se fija un máximo de 15 y su distancia administrativa es de 120 [7], [8], [9], [12], [13].

### **OSPF (*Open Shortest Path First*)**

Es un protocolo de enrutamiento en el que los *routers* cambian entre si las subredes a las que están realizando la convergencia el cual cada *router* calcula el camino óptimo para llegar al resto de *routers* en la red [14], [15], [16].

Según [14] y [15], OSPF es un estándar libre eso quiere decir que es un protocolo que puede ser utilizado por cualquier fabricante diferente a *Cisco*. Hay dos versiones definidas para OSPF que son:

- RFC 2328 versión 2 para IPv4.
- RFC 5340 versión 3 para IPv6.

Este protocolo usa el algoritmo SPF (*Dijkstra*) el cual calcula el mejor camino para llegar a una red específica basándose en el ancho de banda que es la métrica de este protocolo y por otra parte su distancia administrativa es 110 [7], [8], [9].

También puede dar soporte de red escalable y tiempos de convergencia más rápidos el cual es usado considerablemente en redes grandes, como la red troncal de ISP [15].

## **BGP (*Border Gateway Protocol*)**

Es uno de los protocolos de puerta de enlace exterior y el estándar de hecho del protocolo de enrutamiento entre dominios [17], [18], [19], [20].

BGP es un protocolo externo que funciona en el Internet, lo que quiere decir que es un protocolo que ha sido construido para enviar información entre redes que pertenecen a diferentes ISP [17], [18], [19], [20].

Este protocolo es un vector de distancia (*Path Vector*) que no utiliza una métrica como RIP (saltos) y EIGRP (ancho de banda) sino que es un vector de ruta que utiliza atributos [7].

BGP tiene dos distancias administrativas, una interna y una externa ya que se pueden dar dos casos, la primera que dentro de un ISP se tiene implementado BGP entre sus *routers* bajo un mismo dominio, y la segunda es cuando se tiene sesiones externas contra otros proveedores de servicios [8], [9].

Sus distancias administrativas son de 20 para el externo y 200 para el interno en donde BGP suele confiar más en las sesiones externas ya que es menor, además estas sesiones siempre son punto a punto o están directamente conectadas [8], [9].

## **IS-IS (*Intermediate System to Intermediate System*)**

Es un protocolo de enrutamiento IGP que quiere decir Protocolo de puerta de enlace interior que puede dar soporte de red escalable además de tiempos de convergencia más rápidos como OSPF. Este protocolo se usa en grandes redes como ISP así también como en redes troncales de operadores [21].

La métrica de este protocolo de enrutamiento no tiene en cuenta el ancho de banda de las interfaces como lo hace OSPF, sino que tiene un valor por defecto que es el costo de cada enlace [7], [9].

Además de su valor por defecto (costo), IS-IS está basada en otros parámetros:

- *Delay* (Retardo).
- *Expense* (Gasto).
- Error (Confiabilidad).

La distancia administrativa de este protocolo es 115 [8].

## 2 METODOLOGÍA

### 2.1 Descripción de la metodología usada

En la primera sección del proyecto se pretendió analizar las diferentes variables que se presentaron al ejecutar una máquina virtual con el SO *Linux*. En este caso particular para la ejecución y funcionalidad del proyecto se usó la distribución *Ubuntu* 18.04.4 x64.

El contenido que abarcó este punto fueron las herramientas básicas para la instalación de la máquina virtual, la familiarización con su manipulación y las diferentes características que se deben configurar. Dependiendo de la aplicación o finalidad del proyecto, se puede reservar o aumentar recursos respecto a la memoria de la máquina virtual, así como también agregar, activar o habilitar variables funcionales, como interfaces de comunicación.

En la segunda fase, se trató todo el análisis de la plataforma *Quagga*, su arquitectura, y las funcionalidades que esta comprende. Esto puede resumirse en que el sistema donde esté ejecutándose pueda convertirse en un enrutador íntegro que realice enrutamiento de redes de datos, tanto de forma estática como dinámica. A todo esto, se tiene un factor positivo clave, ya que cada protocolo genera su propia tabla de enrutamiento, independientemente de los demás por lo que supone una ventaja importante ya que ningún proceso de enrutamiento cause conflicto sobre otro, tomando las consideraciones debidas como no utilizar las mismas direcciones de red en 2 protocolos distintos.

Una vez investigados los temas relacionados con el Sistema Operativo *Linux*, el respectivo manejo de las máquinas virtuales, su instalación y el manejo del *Quagga*, en la tercera fase se procedió a determinar cuáles son los tipos de protocolos con los que el *software* de *routing* virtual *Quagga* está relacionada (RIP, RIPng, OSPF, BGP, IS-IS). Estos protocolos nos permiten gestionar las tablas de enrutamiento del propio núcleo del sistema tomando en cuenta que se hizo más énfasis en los que tienen principal uso en el presente trabajo de titulación. Se tomó en consideración los siguientes protocolos que son RIP, RIPng, OSFP, BGP tanto para IPv4 e IPv6.

La cuarta fase de este proyecto se ejecutó en dos etapas, una de virtualización y otra de implementación. Para la virtualización se utilizó la distribución *Linux* denominada "*Ubuntu*" en tanto que para la implementación física se usó la distribución "*Raspbian OS*" que es propia para manejarla con las tarjetas inteligentes *Raspberry Pi* 4. El

enfoque de estos puntos se utilizó para la elaboración de 8 hojas guías de laboratorio sobre *routing* para ser aplicadas en las asignaturas de Redes de Computadoras y TCP/IP de la malla curricular de Redes y Telecomunicaciones de la Escuela de Formación de Tecnólogos. Las hojas guías están diseñadas tanto para el estudiante como para el instructor y la temática de las prácticas se dividió en varios puntos como, la instalación de una plataforma de *routing* en la distribución *Ubuntu* virtualizada, aplicación de enrutamiento estático y enrutamiento dinámico con los protocolos RIP, OSPF, BGP, con direcciones IPv4 e IPv6.

Esta fase también se centró en la implementación de las 8 prácticas planteadas, en la primera práctica se virtualizó la distribución *Ubuntu* en VirtualBox TM y principalmente la plataforma de *routing Quagga*, con sus principales paquetes y el manejo de archivos y concesión de permisos para que la ejecución del *router* se efectúe sin inconvenientes.

La segunda práctica esbozó todo el proceso para aplicar rutas estáticas en una red WAN (Red de Área Amplia), los comandos necesarios, la asignación adecuada de direcciones IPv4 para evitar conflictos de direccionamiento y la verificación de la operatividad del enrutamiento estático.

La tercera práctica comenzó con la aplicación de enrutamiento dinámico y en este caso particular con el protocolo RIP, el cual es factible para topologías de red pequeñas y se enfocó en la aplicación de comandos para la ejecución del protocolo, verificación de operatividad del protocolo y análisis de la tabla de enrutamiento general del *Kernel* y la tabla de enrutamiento RIP.

La cuarta práctica se centró de igual manera en los protocolos de enrutamiento dinámico, en este caso del protocolo OSPF, donde los cambios significativos fueron la introducción del concepto de "Área". Un área representa un conjunto de elementos comandados por una sola administración. Esto permite que este protocolo pueda excluir entre "Áreas" para poder segmentar una subred, donde se puede implementar servicios o aplicaciones diferentes e independientes unas de otras o simplemente identificándolas con un número determinado.

La quinta práctica se enfocó en el protocolo de enrutamiento dinámico BGP, los comandos de configuración del protocolo fueron importantes para que la aplicación de este sea correcta y adecuada. Un punto importante que recalcar en esta práctica son los sistemas autónomos (AS), la diferencia fundamental a un "Área" es que este concepto es usado mayormente por la mayoría de los Proveedores de servicios de Internet (ISP) en el mundo para redireccionar todo el tráfico hacia Internet.

La sexta, séptima y octava práctica se basaron en la aplicación de los protocolos de enrutamiento dinámico RIP, OSPF y BGP en la misma topología de red de datos planteada anteriormente. La diferencia fundamental fue que estas se ejecutaron con direcciones IPv6, tomando las consideraciones necesarias para la ejecución, como el cambio de comandos respecto a los efectuados con direccionamiento IPv4 entre otros cambios importantes.

La respectiva validación de las prácticas y del sistema en general se efectuó en la quinta fase y se basó en la comprobación de mensajes ICMP (*Internet Control Message Protocol*), así como también que el inicio de sesión *telnet* se efectúe correctamente con cada demonio de la plataforma, para verificar que la operatividad de los protocolos esté activa y la consulta en la tabla de enrutamiento de cada protocolo como del *Kernel* de *Ubuntu*.

El análisis de resultados del proyecto se llevó a cabo después de que se implementaron todas las prácticas planteadas. La respectiva validación se cotejó con el contenido teórico de los fundamentos de *routing* estático y dinámico, todo esto de forma clara para que el aprendizaje del estudiante sea óptimo. Cabe mencionar que las hojas guías del instructor fueron sumamente detalladas y explícitas para brindar soporte en cualquier duda que se genere en la respectiva aplicación.

### **3 RESULTADOS Y DISCUSIÓN**

Mediante la creación de una máquina virtual se realiza todo el proceso de instalación de la plataforma *Quagga* que es usada como un *router*. La activación de cada demonio respectivo se lo hace mediante una sesión *telnet* apuntada al puerto TCP correspondiente donde se le aplican los protocolos de enrutamiento con cada uno de sus comandos, tomando en cuenta que su sintaxis generalmente es similar a la de *Cisco*. Por último, realizando un *ping* se puede comprobar su conexión exitosa y operabilidad del protocolo sea estático o dinámico, en este caso particular entre dos dispositivos independientes. Los demonios son los protocolos de enrutamiento que maneja la plataforma.

#### **3.1 Identificación de los procesos que sirven para la instalación y manejo de máquinas virtuales en *Linux***

Se debe tener en cuenta los tipos de virtualización que hay y para ello se analizaron los mismos antes de desarrollar la primera fase.

Hay dos tipos de virtualización, de tipo 1 y de tipo 2. Para esto se utiliza un *hypervisor* el cual es un entorno mediante el que podemos ejecutar máquinas virtuales [22], [23].

### Hipervisor de tipo 1

Estos son también conocidos como “*bare-metal*” o nativos, los cuales tienen acceso directo al *hardware*, en otras palabras, es aquel en el que se instala el *hypervisor* directamente en el *hardware* y a partir de ello se puede implementar las máquinas virtuales. Las máquinas virtuales, así como el sistema operativo, son plenamente independientes, es decir si hay algún problema en un sistema esto no va a afectar a los demás ya que tienen independencia uno del otro [23], [24].

El *hypervisor* de tipo 1 está orientado a entornos profesionales para realizar pruebas de conectividad de sistemas en donde se tenga en cuenta que el rendimiento y la estabilidad son muy importantes [23], [24].

Como ejemplos del *hypervisor* de tipo 1 se tienen: *Hyper-V*, *VmWare ESXi*, *Citrix Xen Server*, etc. El esquema del *hypervisor* de tipo 1 se muestra en la Figura 3.1 [24].

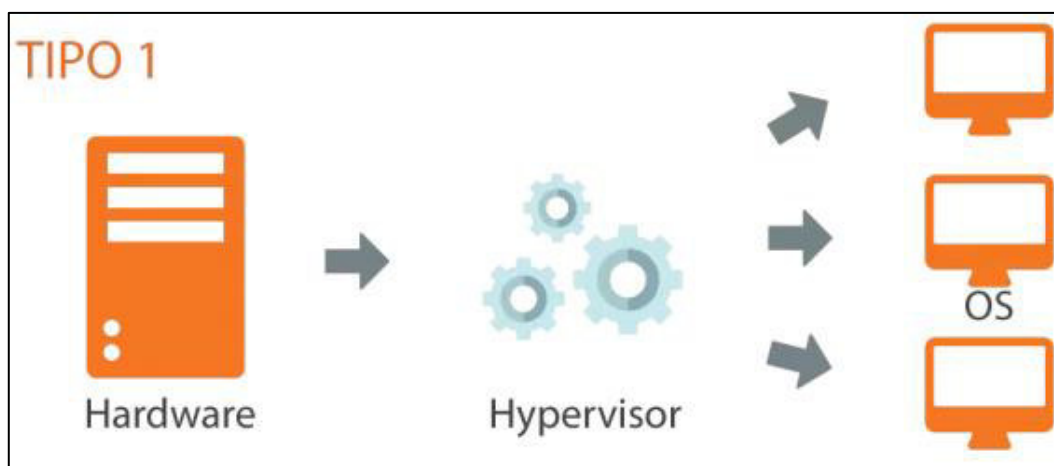


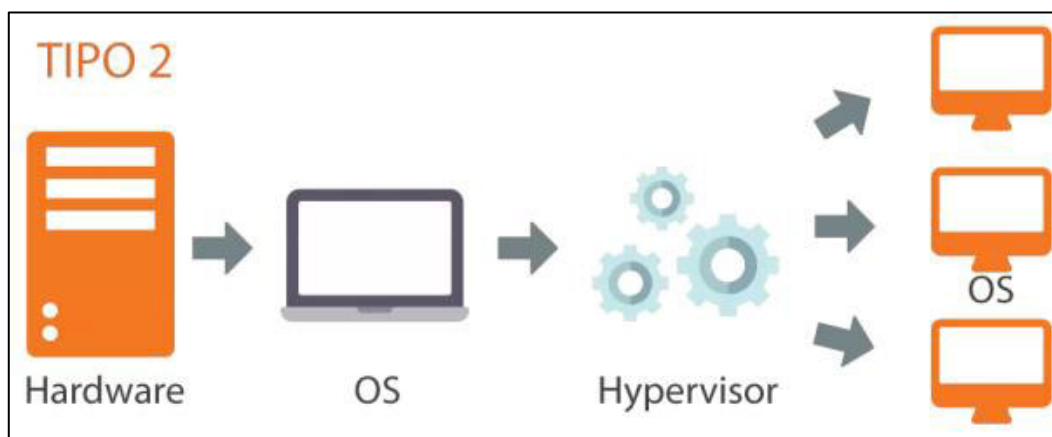
Figura 3.1 Esquema de *hypervisor* de Tipo 1 [24]

### Hipervisor de tipo 2

Se ejecuta sobre el sistema operativo el cual es conocido también como “*hosted*” que quiere decir hospedado. Este tiene una mayor compatibilidad de *hardware* y al contrario con lo que pasa con el de tipo 1, en este la estabilidad y el rendimiento son menores debido a que todo está soportado por un entorno de *software*, esto quiere decir que se ejecuta sobre un sistema operativo con la finalidad de dar la funcionalidad descrita [23], [24].

Dicho de otra manera, primero se instala el sistema operativo (*Windows, Linux, etc.*) y sobre este sistema operativo se instala el *hypervisor* para luego sobre este desplegar las máquinas virtuales [24].

Como ejemplos del *hypervisor* de tipo 2 se tienen: *VirtualPC, VmWare Workstation, Virtual Box, etc.* El esquema del *hypervisor* de tipo 2 se muestra en la Figura 3.2 [24].



**Figura 3.2** Esquema de *hypervisor* de Tipo 2 [24]

Una vez analizados los tipos de virtualización, se puede decir que para este caso se trabajó bajo las condiciones del *hypervisor* de tipo 2 ya que para este proyecto se utilizó el *software Virtual Box* para desarrollar una virtualización del sistema operativo *Linux* sobre el sistema operativo *Windows*.

Hay que destacar que el manejo de máquinas virtuales para administración y creación de una red, son de vital importancia en el mundo actual. Con la virtualización se pretende la maximización de recursos que es una gran salida para el ahorro y flexibilidad en cuanto a equipos físicos los cuales ocupan una gran cantidad de espacio siendo una desventaja considerable.

A continuación, se presenta la máquina virtual en *Linux* ejecutándose correctamente sobre el Sistema Operativo *Windows* tal como se observa en la **Figura 3.3** y **Figura 3.4**.



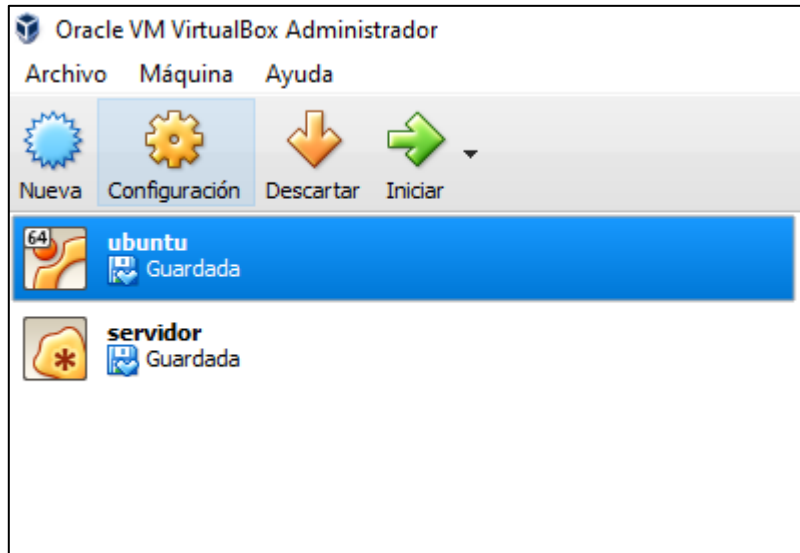


Figura 3.3 Creación de Máquina Virtual

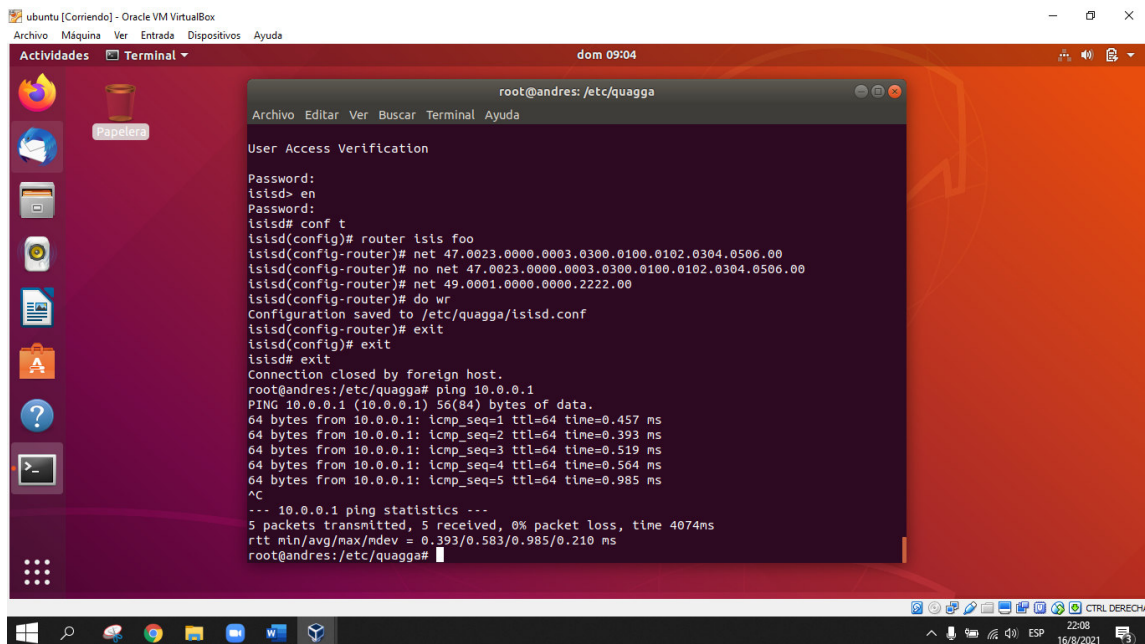


Figura 3.4 Máquina Virtual sobre entorno Windows

## 3.2 Análisis de los parámetros fundamentales del software Quagga y sus funcionalidades

### Quagga

Es una *suite* de *software* de *routing* con licencia GPL, que puede aplicar diversos protocolos de enrutamiento dinámico TCP/IP en la capa de red, exactamente como un *router*, tales como OSPFv2, OSPFv3, RIP v1 y v2, RIPng, BGP-4 y IS-IS. Para sistemas operativos basados en *Unix*, como lo son FreeBSD, *Linux*, Solaris y NetBSD. Todo el

*software* de enrutamiento está basado en un demonio llamado *Zebra*, que ejecuta todas las funciones principales del sistema, como administración, asignación de direcciones IP, enrutamiento estático y dinámico entre otros. Este *software* fue desarrollado por Kunihiro Ishiguro en 2005 [25].

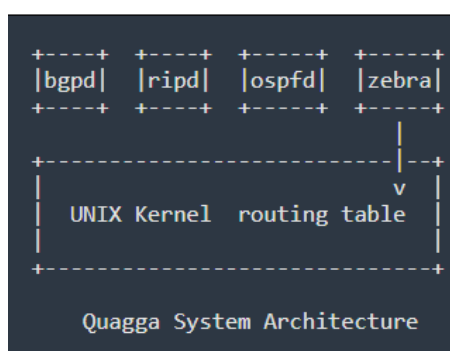
### Arquitectura

La arquitectura de *Quagga*, tiene un demonio central que gestiona todas las tablas de enrutamiento y funciona también como una capa de separación con el *Kernel Unix* del propio sistema operativo, teniendo en cuenta la compatibilidad de estos [25].

El sistema también tiene un API (*Application Programming Interface*) de servidor de conexión (*Zserv*) sobre *Unix* o TCP a los usuarios de *Quagga*. Un API es una interfaz que permite la comunicación entre aplicaciones, programas de *software*, mediante una serie de reglas para que puedan compartir datos entre ellos.

La implicación del *Zserv* es importante en el sistema, ya que este ejecuta los protocolos de enrutamiento que soporta el sistema y traslada las actualizaciones de enrutamiento al demonio principal *Zebra* [25], [26].

Los demonios de *routing*, tienen su propio sistema de archivos e interfaz de terminal, convirtiendo a la gestión del sistema multiproceso, ya que no hay necesidad de que un solo sistema realice cada “ruteo” sino que, un demonio individual realice su propio protocolo, reportando el informe de enrutamiento al demonio central “*Zebra*”, graficando la gestión del sistema total, sería semejante al de la **Figura 3.5** [25].



**Figura 3.5** Arquitectura de *Quagga* [25]

Teniendo en cuenta que los protocolos que soporta el *Zserv* son variados, se enlista los más importantes a continuación, en la Figura 3.6 se observa los protocolos analizados a profundidad posteriormente [25].

IPv4	IPv6
zebra	
ripd	ripngd
ospfd	ospf6d
bgpd	
isisd	

**Figura 3.6** Protocolos de enrutamiento soportado en *Quagga Routing Suite* [25]

La configuración de los demonios dentro del *software* se realiza a través de una terminal o CLI (*COMMAND LINE INTERFACE*), denominada “*vty*”. La ventaja particular del CLI de *Quagga* es que es muy similar al de cualquier otro *software* de *routing*, incluso a un *router* físico de cualquier marca comercial vigente. La interfaz como tal, es un punto de interacción y administración entre todos los demonios y se la puede llamar a través de la línea de comandos con la instrucción “*vtys*h” [25].

La plataforma *Quagga* se basa de forma general en la homologación de un *router* sobre un sistema operativo de licencia libre [25].

En las **Figura 3.7**, **Figura 3.8** y **Figura 3.9** demuestran que con el demonio central (*zebra*), se ejecuta a plenitud toda la operatividad de un *router*, como por ejemplo la asignación de direcciones IPv4 e IPv6, visualización de la tabla de enrutamiento del *Kernel*, comprobación de interfaces de comunicación operativas, entre otras funciones esenciales de cualquier otro *router* de marca comercial.

```
Router# configure terminal
Router(config)# inter
Router(config)# interface enp0s3
Router(config-if)# ip address 10.0.0.1/24
Router(config-if)# do wr
Configuration saved to /etc/quagga/zebra.conf
Router(config-if)# exit
Router(config)# interface enp0s8
Router(config-if)# ip address 192.168.2.1/24
Router(config-if)# do wr
Configuration saved to /etc/quagga/zebra.conf
Router(config-if)# exit
Router(config)# do wr
Configuration saved to /etc/quagga/zebra.conf
Router(config)# exit
```

**Figura 3.7** Asignación de direcciones IPv4 en *router Quagga*

```

Router# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, P - PIM, A - Babel, N - NHRP,
       > - selected route, * - FIB route

C>* 1.1.1.1/32 is directly connected, lo
B>* 2.2.2.2/32 [20/0] via 10.1.1.2, enp0s3, 06:45:12
C>* 10.0.0.0/24 is directly connected, enp0s3
C>* 10.1.1.0/24 is directly connected, enp0s3
O 33.1.1.0/24 [110/10] is directly connected, enp0s3, 00:49:13
C>* 33.1.1.0/24 is directly connected, enp0s3
O 122.1.1.0/24 [110/10] is directly connected, enp0s8, 00:48:55
C>* 122.1.1.0/24 is directly connected, enp0s8
C>* 127.0.0.0/8 is directly connected, lo
O 133.1.1.0/24 [110/20] via 33.1.1.2, enp0s3, 00:40:00
B>* 133.1.1.0/24 [20/0] via 10.1.1.2, enp0s3, 00:58:12
B>* 169.254.0.0/16 [20/0] via 10.1.1.2, enp0s3, 06:45:12
C>* 192.168.1.0/24 is directly connected, enp0s8
S 192.168.1.0/24 [1/0] via 10.0.0.2, enp0s3
R 192.168.2.0/24 [120/2] via 10.1.1.2, enp0s3, 05:30:31
B>* 192.168.2.0/24 [20/0] via 10.1.1.2, enp0s3, 06:29:02
C>* 192.168.101.0/24 is directly connected, enp0s8

```

Figura 3.8 Tabla de enrutamiento del *router Quagga*

```

router1-VirtualBox# show interface
Interface enp0s3 is up, line protocol is up
Link ups:      1 last: Sun, 09 May 2021 18:51:13 -0500
Link downs:    0 last: (never)
vrf: 0
index 2 metric 0 mtu 1500
flags: <UP,BROADCAST,RUNNING,MULTICAST>
Type: Ethernet
HWaddr: 08:00:27:72:d0:30
inet 10.1.1.1/24 broadcast 10.1.1.255
inet6 2001:db8:3::1/64
inet6 fe80::a00:27ff:fe72:d030/64
inet6 2001:db8:4::1/64
Interface enp0s8 is up, line protocol is up
Link ups:      1 last: Sun, 09 May 2021 18:51:13 -0500
Link downs:    0 last: (never)
vrf: 0
index 3 metric 0 mtu 1500
flags: <UP,BROADCAST,RUNNING,MULTICAST>
Type: Ethernet
HWaddr: 08:00:27:a7:43:d5
inet 192.168.101.1/24 broadcast 192.168.101.255
inet6 2001:db8:1::1/64
inet6 fe80::a00:27ff:fea7:43d5/64
inet6 2001:db8:4::1/64

```

Figura 3.9 Interfaces operativas de *router Quagga*

### 3.3 Determinación de los tipos de protocolos que maneja el software de *routing* virtual *Quagga*

Con los paquetes necesarios y sus respectivas configuraciones la plataforma *Quagga* puede manejar seis tipos de protocolos y un demonio principal que sirve para la gestión centralizada del sistema. Mediante el modo terminal (*telnet*) y su respectivo puerto TCP

o demonio de cada protocolo se puede ingresar para realizar el enrutamiento. Es importante aclarar que la plataforma discrimina entre cada demonio disponible en el sistema, es decir cada protocolo es independiente uno de otro.

A continuación, se muestran los estados de cada protocolo disponibles en la plataforma, mediante el comando “*sudo service zebra status*” se puede observar el estado activo del demonio central como se muestra en la Figura 3.10.

```
root@tesis-VirtualBox:/# sudo service zebra start
root@tesis-VirtualBox:/# sudo service zebra status
● zebra.service - GNU Zebra routing manager
   Loaded: loaded (/lib/systemd/system/zebra.service; enabled; vendor preset: e
   Active: active (running) since Wed 2020-11-11 00:43:29 -05; 6s ago
     Docs: man:zebra
   Process: 6853 ExecStart=/usr/sbin/zebra -d -A 127.0.0.1 -f /etc/quagga/zebra.
   Process: 6852 ExecStartPre=/bin/chown -f quagga:quaggavty /etc/quagga/vtysh.c
   Process: 6851 ExecStartPre=/bin/chown -f quagga:quagga /etc/quagga/zebra.conf
   Process: 6850 ExecStartPre=/bin/chmod -f 640 /etc/quagga/vtysh.conf /etc/quag
   Process: 6849 ExecStartPre=/sbin/ip route flush proto zebra (code=exited, sta
   Main PID: 6855 (zebra)
     Tasks: 1 (limit: 4667)
    CGroup: /system.slice/zebra.service
           └─6855 /usr/sbin/zebra -d -A 127.0.0.1 -f /etc/quagga/zebra.conf
```

**Figura 3.10** Demonio Central *Zebra* en estado activo

Mediante el comando “*sudo service bgpd status*” se puede observar el estado activo del protocolo BGP como se muestra en la Figura 3.11.

```
root@tesis-VirtualBox:/# sudo service bgpd start
root@tesis-VirtualBox:/# sudo service bgpd status
● bgpd.service - BGP routing daemon
   Loaded: loaded (/lib/systemd/system/bgpd.service; enabled; vendor preset: en
   Active: active (running) since Wed 2020-11-11 00:44:11 -05; 15s ago
     Docs: man:bgpd
   Process: 6891 ExecStart=/usr/sbin/bgpd -d -A 127.0.0.1 -f /etc/quagga/bgpd.co
   Process: 6890 ExecStartPre=/bin/chown -f quagga:quagga /etc/quagga/bgpd.conf
   Process: 6889 ExecStartPre=/bin/chmod -f 640 /etc/quagga/bgpd.conf (code=exit
   Main PID: 6893 (bgpd)
     Tasks: 1 (limit: 4667)
    CGroup: /system.slice/bgpd.service
           └─6893 /usr/sbin/bgpd -d -A 127.0.0.1 -f /etc/quagga/bgpd.conf

nov 11 00:44:11 tesis-VirtualBox systemd[1]: Starting BGP routing daemon...
nov 11 00:44:11 tesis-VirtualBox systemd[1]: Started BGP routing daemon.
lines 1-14/14 (END)
```

**Figura 3.11** Demonio *bgpd* en estado activo

Mediante el comando “*sudo service ospfd status*” se puede observar el estado activo del protocolo OSPF como se muestra en la Figura 3.12.

```
root@tesis-VirtualBox:/# sudo service ospfd start
root@tesis-VirtualBox:/# sudo service ospfd status
● ospfd.service - OSPF routing daemon
   Loaded: loaded (/lib/systemd/system/ospfd.service; enabled; vendor preset: e
   Active: active (running) since Wed 2020-11-11 01:06:18 -05; 2s ago
     Docs: man:ospfd
   Process: 7388 ExecStart=/usr/sbin/ospfd -d -A 127.0.0.1 -f /etc/quagga/ospfd.
   Process: 7387 ExecStartPre=/bin/chown -f quagga:quagga /etc/quagga/ospfd.conf
   Process: 7386 ExecStartPre=/bin/chmod -f 640 /etc/quagga/ospfd.conf (code=exl
   Main PID: 7389 (ospfd)
     Tasks: 1 (limit: 4667)
    CGroup: /system.slice/ospfd.service
           └─7389 /usr/sbin/ospfd -d -A 127.0.0.1 -f /etc/quagga/ospfd.conf

nov 11 01:06:18 tesis-VirtualBox systemd[1]: Starting OSPF routing daemon...
nov 11 01:06:18 tesis-VirtualBox systemd[1]: Started OSPF routing daemon.
```

**Figura 3.12** Demonio *ospfd* en estado activo

Mediante el comando “*sudo service ospf6d status*” se puede observar el estado activo del protocolo OSPF en su versión 3 con direccionamiento IPv6 como se muestra en la Figura 3.13.

```
root@tesis-VirtualBox:/# sudo service ospf6d start
root@tesis-VirtualBox:/# sudo service ospf6d status
● ospf6d.service - OSPF routing daemon for IPv6
   Loaded: loaded (/lib/systemd/system/ospf6d.service; enabled; vendor preset:
   Active: active (running) since Wed 2020-11-11 01:06:39 -05; 3s ago
     Docs: man:ospf6d
   Process: 7426 ExecStart=/usr/sbin/ospf6d -d -A ::1 -f /etc/quagga/ospf6d.conf
   Process: 7425 ExecStartPre=/bin/chown -f quagga:quagga /etc/quagga/ospf6d.conf
   Process: 7424 ExecStartPre=/bin/chmod -f 640 /etc/quagga/ospf6d.conf (code=ex
   Main PID: 7427 (ospf6d)
     Tasks: 1 (limit: 4667)
    CGroup: /system.slice/ospf6d.service
           └─7427 /usr/sbin/ospf6d -d -A ::1 -f /etc/quagga/ospf6d.conf

nov 11 01:06:39 tesis-VirtualBox systemd[1]: Starting OSPF routing daemon for I
nov 11 01:06:39 tesis-VirtualBox systemd[1]: Started OSPF routing daemon for IP
```

**Figura 3.13** Demonio *ospf6d* en estado activo

Mediante el comando “*sudo service ripd status*” se puede observar el estado activo del protocolo RIP como se muestra en la Figura 3.14.

```
root@tesis-VirtualBox:/# sudo service ripd start
root@tesis-VirtualBox:/# sudo service ripd status
● ripd.service - RIP routing daemon
   Loaded: loaded (/lib/systemd/system/ripd.service; enabled; vendor preset: en
   Active: active (running) since Wed 2020-11-11 01:07:02 -05; 4s ago
     Docs: man:ripd
   Process: 7464 ExecStart=/usr/sbin/ripd -d -A 127.0.0.1 -f /etc/quagga/ripd.co
   Process: 7463 ExecStartPre=/bin/chown -f quagga:quagga /etc/quagga/ripd.conf
   Process: 7462 ExecStartPre=/bin/chmod -f 640 /etc/quagga/ripd.conf (code=exit
   Main PID: 7465 (ripd)
     Tasks: 1 (limit: 4667)
    CGroup: /system.slice/ripd.service
           └─7465 /usr/sbin/ripd -d -A 127.0.0.1 -f /etc/quagga/ripd.conf

nov 11 01:07:02 tesis-VirtualBox systemd[1]: Starting RIP routing daemon...
nov 11 01:07:02 tesis-VirtualBox systemd[1]: Started RIP routing daemon.
```

**Figura 3.14** Demonio *ripd* en estado activo

Mediante el comando “*sudo service ripngd status*” se puede observar el estado activo del protocolo RIP en su nueva versión con direccionamiento IPv6 como se muestra en la Figura 3.15.

```
root@tesis-VirtualBox:/# sudo service ripngd start
root@tesis-VirtualBox:/# sudo service ripngd status
● ripngd.service - RIP routing daemon for IPv6
   Loaded: loaded (/lib/systemd/system/ripngd.service; enabled; vendor preset:
   Active: active (running) since Wed 2020-11-11 01:07:26 -05; 2s ago
     Docs: man:ripngd
   Process: 7508 ExecStart=/usr/sbin/ripngd -d -A :::1 -f /etc/quagga/ripngd.conf
   Process: 7507 ExecStartPre=/bin/chown -f quagga:quagga /etc/quagga/ripngd.conf
   Process: 7506 ExecStartPre=/bin/chmod -f 640 /etc/quagga/ripngd.conf (code=ex
 Main PID: 7509 (ripngd)
    Tasks: 1 (limit: 4667)
   CGroup: /system.slice/ripngd.service
           └─7509 /usr/sbin/ripngd -d -A :::1 -f /etc/quagga/ripngd.conf

nov 11 01:07:26 tesis-VirtualBox systemd[1]: Starting RIP routing daemon for IP
nov 11 01:07:26 tesis-VirtualBox systemd[1]: Started RIP routing daemon for IPv
```

**Figura 3.15** Demonio *ripngd* en estado activo

Mediante el comando “*sudo service isisd status*” se puede observar el estado activo del protocolo ISIS como se muestra en la Figura 3.16.

```
root@tesis-VirtualBox:/# sudo service isisd start
root@tesis-VirtualBox:/# sudo service isisd status
● isisd.service - IS-IS routing daemon
   Loaded: loaded (/lib/systemd/system/isisd.service; enabled; vendor preset: e
   Active: active (running) since Wed 2020-11-11 01:07:52 -05; 3s ago
     Docs: man:isisd
   Process: 7545 ExecStart=/usr/sbin/isisd -d -A 127.0.0.1 -f /etc/quagga/isisd.
   Process: 7544 ExecStartPre=/bin/chown -f quagga:quagga /etc/quagga/isisd.conf
   Process: 7543 ExecStartPre=/bin/chmod -f 640 /etc/quagga/isisd.conf (code=ex
 Main PID: 7546 (isisd)
    Tasks: 1 (limit: 4667)
   CGroup: /system.slice/isisd.service
           └─7546 /usr/sbin/isisd -d -A 127.0.0.1 -f /etc/quagga/isisd.conf
```

**Figura 3.16** Demonio *isisd* en estado activo

### 3.4 Diseño de prácticas de laboratorio

El diseño y desarrollo de las prácticas de laboratorio cumple con el objetivo de reforzar el contenido teórico visto en clase y que el estudiante aplique los conocimientos adquiridos en cuanto a redes y máquinas virtuales, visualizando de forma más amplia y ayudando a comprender de forma más clara todo lo relacionado a enrutamiento y direccionamiento.

Establecer la guía de manejo del estudiante e instructor está directamente cubierto en el desarrollo de la práctica 1, ya que la misma explica como configurar la máquina virtual y la manipulación de los demonios para realizar los respectivos enrutamientos con los protocolos.

El diseño correspondiente de las prácticas de laboratorio para el alumno y el docente se encuentra disponible en el Anexo 2.

### **Práctica 1: Instalación y configuración de máquinas virtuales y *software* de *Routing Quagga***

#### **Objetivos**

- Detallar el procedimiento para la creación y configuración de una máquina virtual en *Virtual Box TM*.
- Conocer los fundamentos sobre *software* virtual basado en licencia libre.
- Seguir el procedimiento y manejo de archivos para la instalación de *Quagga* en *Ubuntu 18.04.4*.

**Descripción:** La primera práctica de laboratorio se enfoca en el aprendizaje de máquinas virtuales, su configuración y manipulación en este caso particular con la distribución *Ubuntu*. Se destaca principalmente la instalación de la plataforma *Quagga*, los paquetes necesarios, el manejo de archivos para que el funcionamiento sea óptimo y poder ejecutar un *router* sin la necesidad de perder las características propias del sistema operativo.

El objetivo de esta práctica es la familiarización del estudiante y el instructor con la plataforma, como punto primordial es el inicio de sesión *telnet* con todos los demonios disponibles en el sistema para la correcta aplicación de estos.

### **Práctica 2: Práctica de *Routing* con rutas estáticas en *Quagga*, simulando una RED WAN**

#### **Objetivos**

- Aplicar conceptos sobre rutas estáticas en direccionamiento y enrutamiento en redes WAN.
- Verificar los cambios en tabla de enrutamiento del router *Quagga* antes y después del ruteo.
- Analizar el beneficio de la aplicación de rutas estáticas en redes WAN.

**Descripción:** El desarrollo de la práctica de rutas estáticas engloba todo el contenido teórico y práctico de direccionamiento y enrutamiento de redes, con la particularidad que, para esta práctica, el demonio central del sistema (*zebra*) será el encargado de asignar las direcciones IPv4 y también la respectiva configuración en el *Router Quagga* para que se pueda enviar tráfico de un extremo a otro de la red.



### **Práctica 3: Práctica de *Routing* con el protocolo de enrutamiento dinámico RIP en *Quagga*, simulando una RED WAN**

#### **Objetivos**

- Comparar el contenido teórico y práctico del protocolo de enrutamiento RIP con direcciones IPV4
- Reconocer el tiempo que tarda la compartición de la tabla de enrutamiento entre routers con el protocolo RIP
- Determinar si el protocolo RIP tiene mayores beneficios en redes pequeñas o redes más grandes.

**Descripción:** La tercera práctica se enfoca en la aplicación de los conocimientos de rutas dinámicas, en este caso RIP, asignando direcciones IPv4 mediante el demonio central (*zebra*) al cual se ingresa con el modo terminal (*telnet*). Después se realiza el enrutamiento dinámico RIP ingresando al *router*, al cual también se accede con *telnet* y su respectivo puerto TCP 2602. Finalmente se comprueba su conexión realizando *ping* entre los dos dispositivos configurados, verificando que el protocolo esté operativo en la tabla de enrutamiento del sistema.

### **Práctica 4: Práctica de *Routing* con el protocolo de enrutamiento dinámico OSPF en *Quagga*, simulando una RED WAN**

#### **Objetivos**

- Comparar el contenido teórico y práctico del protocolo de enrutamiento OSPF con direcciones IPV4
- Identificar el concepto de Sistema Autónomo (AS) en redes con el protocolo OSPF
- Determinar el nombre de la división de un sistema autónomo.

**Descripción:** La cuarta práctica se enfoca en la aplicación de los conocimientos de rutas dinámicas, en este caso OSPF, asignando direcciones IPv4 mediante el demonio central (*zebra*) al cual se ingresa con el modo terminal (*telnet*). Después se realiza el enrutamiento dinámico OSPF ingresando al *router*, al cual también se accede con *telnet* y su respectivo puerto TCP 2606. Finalmente se comprueba su conexión realizando *ping* entre los dos dispositivos configurados, verificando que el protocolo esté operativo en la tabla de enrutamiento del sistema.

## **Práctica 5: Práctica de *Routing* con el protocolo de enrutamiento dinámico BGP en *Quagga*, simulando una RED WAN**

### **Objetivos**

- Analizar el contenido teórico para identificar las aplicaciones y ventajas basado en el protocolo BGP.
- Implementar el protocolo BGP para visualizar de mejor manera sus funciones y comportamiento.
- Especificar el concepto de sistema autónomo (AS) en una red de datos y la relación con el protocolo de enrutamiento BGP.

**Descripción:** La quinta práctica se enfoca en la aplicación de los conocimientos de rutas dinámicas, en este caso BGP, asignando direcciones IPv4 mediante el demonio central (*zebra*) al cual se ingresa con el modo terminal (*telnet*). Luego se realiza el enrutamiento dinámico BGP ingresando al *router*, al cual también se accede con *telnet* y su respectivo puerto TCP 2605. Finalmente se comprueba su conexión realizando *ping* entre los dos dispositivos configurados, verificando que el protocolo esté operativo en la tabla de enrutamiento del sistema.

## **Práctica 6: Práctica de *Routing* con el protocolo de enrutamiento dinámico RIPng en *Quagga*, simulando una RED WAN**

### **Objetivos**

- Comparar la actualización del protocolo RIP en IPv4 con su versión en IPv6 (RIPng)
- Establecer los parámetros teóricos como actualización y expiración de rutas en la aplicación del protocolo.
- Comprobar la comunicación de extremo a extremo mediante mensajes simples (*ping*)
- Considerar las semejanzas en rutas RIP tanto para direcciones IPv4 como IPv6.

**Descripción:** La sexta práctica se enfoca en la aplicación de los conocimientos de rutas dinámicas, en este caso RIPng, asignando direcciones IPv6 mediante el demonio central (*zebra*) al cual se ingresa con el modo terminal (*telnet*). Luego se realiza el enrutamiento dinámico RIPng ingresando al *router*, al cual también se accede con *telnet* seguido de `::` y su respectivo puerto TCP 2603. Finalmente se comprueba su conexión realizando *ping* entre los dos dispositivos configurados, verificando que el protocolo esté operativo en la tabla de enrutamiento del sistema.

## **Práctica 7: Práctica de *Routing* con el protocolo de enrutamiento dinámico OSPF6d en *Quagga*, simulando una RED WAN**

### **Objetivos**

- Aplicar los conocimientos adquiridos en clase que se basan en el direccionamiento y enrutamiento básico de redes enfocado a OSPF con direcciones IPv6.
- Configurar los routers Quagga con las direcciones IPv6 y comandos propuestos por el instructor, siguiendo la misma sintaxis de la marca Cisco.
- Analizar las semejanzas y diferencias entre las versiones disponibles del protocolo OSPF con la correlación de direcciones IPv4 e IPv6.

**Descripción:** La séptima práctica se enfoca en la aplicación de los conocimientos de rutas dinámicas, en este caso OSPF6d, asignando direcciones IPv6 mediante el demonio central (*zebra*) al cual se ingresa con el modo terminal (*telnet*). Después se realiza el enrutamiento dinámico OSPF6d ingresando al *router*, al cual también se accede con *telnet* seguido de `::` y su respectivo puerto TCP 2607. Por último, se comprueba su conexión realizando *ping* entre los dos dispositivos configurados, verificando que el protocolo esté operativo en la tabla de enrutamiento del sistema.

## **Práctica 8: Práctica de *Routing* con el protocolo de enrutamiento dinámico BGP con direcciones IPv6 en *Quagga*, simulando una RED WAN**

### **Objetivos**

- Emplear el contenido teórico para el protocolo BGP con direcciones IPv6.
- Analizar cual distancia administrativa es la óptima tanto para redes pequeñas como para redes grandes.
- Comprobar la comunicación de extremo a extremo.
- Comparar la diferencia o similitud del direccionamiento BGP con direcciones IPv4 e IPv6.

**Descripción:** La octava práctica se enfoca en la aplicación de los conocimientos de rutas dinámicas, en este caso BGP, asignando direcciones IPv6 mediante el demonio central (*zebra*) al cual se ingresa con el modo terminal (*telnet*). Después se realiza el enrutamiento dinámico BGP ingresando al *router*, al cual también se accede con *telnet* y su respectivo puerto TCP 2605. Finalmente se comprueba su conexión realizando *ping* entre los dos dispositivos configurados, verificando que el protocolo esté operativo en la tabla de enrutamiento del sistema.

### 3.5 Verificación del funcionamiento del sistema

En este punto se puso a prueba la comunicación de los protocolos en los dos sistemas independientes tanto de manera virtual (máquinas virtuales) como de forma física (módulos *Raspberry Pi*).

Con los resultados que se presentan en las figuras **Figura 3.17** y **Figura 3.18** se comprobó la operatividad del protocolo de enrutamiento RIP y en las figuras **Figura 3.19** y **Figura 3.20**, la actualización de la tabla de enrutamiento del sistema con las direcciones IP asignadas en dicho protocolo.

```
root@router1-VirtualBox:/home/router1# ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
64 bytes from 192.168.2.1: icmp_seq=1 ttl=64 time=0.367 ms
64 bytes from 192.168.2.1: icmp_seq=2 ttl=64 time=0.475 ms
64 bytes from 192.168.2.1: icmp_seq=3 ttl=64 time=0.475 ms
64 bytes from 192.168.2.1: icmp_seq=4 ttl=64 time=0.434 ms
^C
--- 192.168.2.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3060ms
rtt min/avg/max/mdev = 0.367/0.437/0.475/0.051 ms
root@router1-VirtualBox:/home/router1#
```

**Figura 3.17** Ping de Router 1 a Router 2

```
root@router2-VirtualBox:~# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0.024 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.040 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=0.095 ms
^C
--- 192.168.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2025ms
rtt min/avg/max/mdev = 0.024/0.053/0.095/0.030 ms
root@router2-VirtualBox:~#
```

**Figura 3.18** Ping de Router 2 a Router 1

```
ripd# conf t
ripd(config)# show ip rip
% [RIP] Unknown command: show ip rip
ripd(config)# exit
ripd# show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface

      Network          Next Hop          Metric From      Tag Time
C(i) 10.0.0.0/24      0.0.0.0           1 self           0
C(i) 192.168.1.0/24  0.0.0.0           1 self           0
R(n) 192.168.2.0/24  10.1.1.2          2 10.1.1.2       0 02:49
ripd#
```

**Figura 3.19** Tabla de enrutamiento RIP del Router 1

```
ripd# show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
  (n) - normal, (s) - static, (d) - default, (r) - redistribute,
  (i) - interface

Network      Next Hop      Metric From      Tag Time
C(i) 10.0.0.0/24 0.0.0.0        1 self           0
R(n) 192.168.1.0/24 10.1.1.1      2 10.1.1.1       0 02:42
C(i) 192.168.2.0/24 0.0.0.0        1 self           0
ripd#
```

**Figura 3.20** Tabla de enrutamiento RIP del Router 2

Las pruebas de conexión y operatividad de los protocolos estáticos y dinámicos que admite *Quagga* se realizaron en todas las prácticas planteadas en el proyecto.

Para comprobar que las prácticas desarrolladas en el entorno virtual son totalmente funcionales en el entorno físico se utilizaron los módulos *Raspberry Pi* para diseñar la topología planteada y las configuraciones necesarias para la respectiva verificación de comunicación, con sus respectivas tarjetas de red (NIC con conexión a USB) adaptadas que sirvieron para compensar la falta de puertos RJ45 que se ilustra en la Figura 3.21 y las pruebas correspondientes de extremo a extremo tal como se aprecia en las figuras Figura 3.22, Figura 3.23, Figura 3.24 y Figura 3.25



**Figura 3.21** NIC con conexión a USB

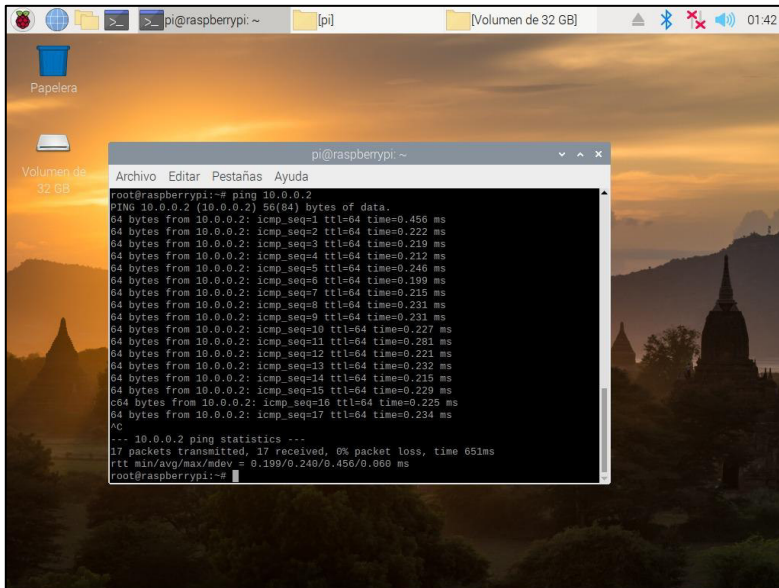


Figura 3.22 Ping físico de módulo *Raspberry 1* a *Raspberry 2*

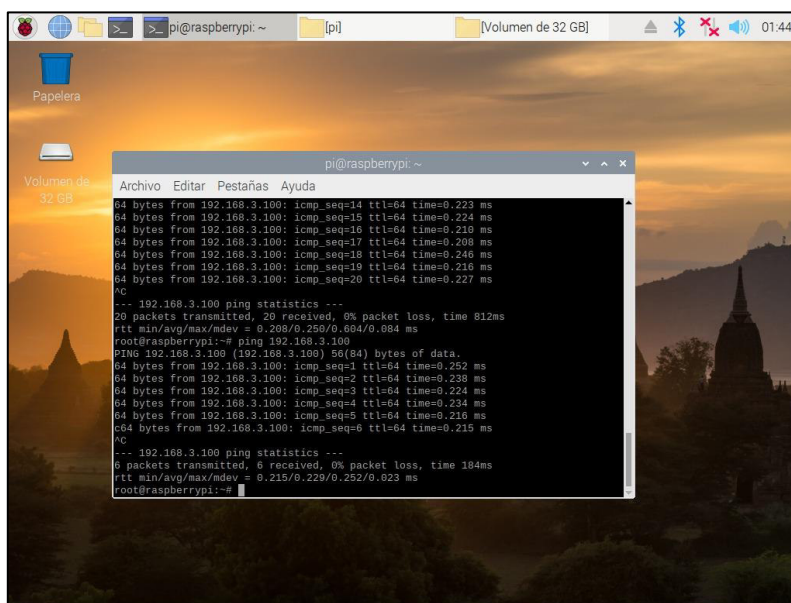


Figura 3.23 Ping físico del módulo *Raspberry 1* a interfaz de salida del módulo *Raspberry 2*

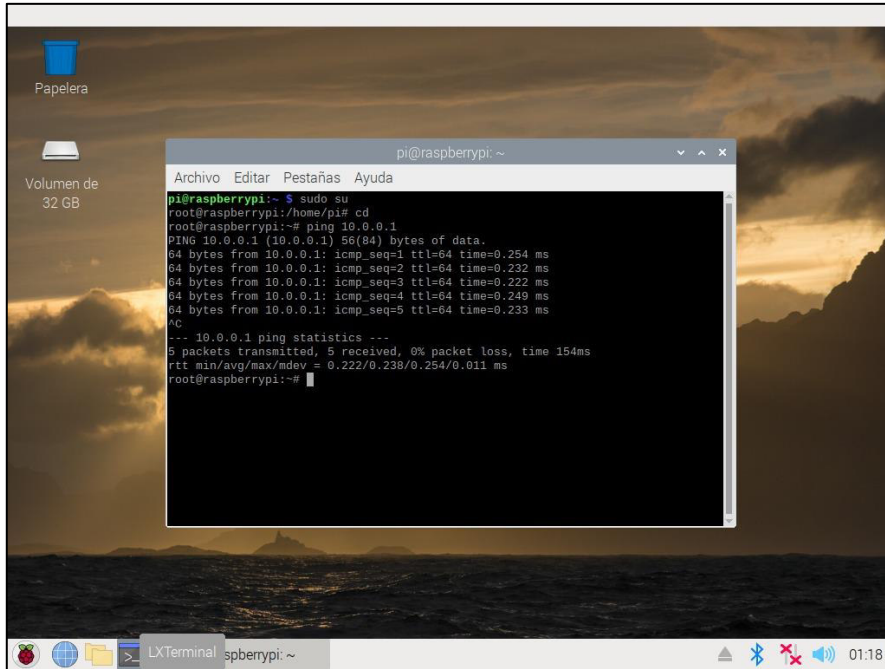


Figura 3.24 Ping físico de módulo Raspberry 2 a Raspberry 1

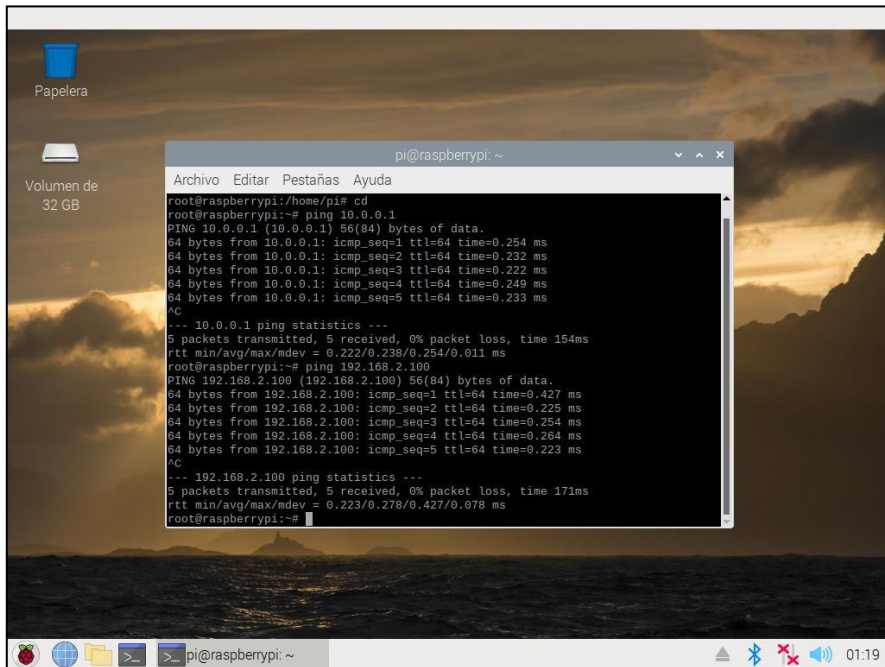


Figura 3.25 Ping físico del módulo Raspberry 2 a interfaz de salida del módulo Raspberry 1

## 4 CONCLUSIONES Y RECOMENDACIONES

### 4.1 Conclusiones

- La implementación virtual reduce en gran medida el espacio físico que se utilizaría para la creación de una red con varios dispositivos como *routers* o *switches*.
- El manejo y ejecución de los comandos que se utilizan para realizar los diferentes enrutamientos es similar en cuanto a su sintaxis a la marca *Cisco* y demás marcas reconocidas.
- Tomando en cuenta las métricas de los diferentes protocolos de enrutamiento dinámico se puede decir que el mejor es OSPF, ya que este maneja anchos de banda lo que significa que toma el camino más fácil para que el paquete llegue a su destino, mientras que RIP encuentra caminos cortos mediante saltos.
- OSPF al usar anchos de banda se puede manejar en redes grandes sin mayores inconvenientes al contrario de lo que hace RIP.
- RIP es el protocolo que se puede configurar de un modo fácil y por su limitación se usa en redes pequeñas locales.
- Para poder ingresar al *router* mediante *telnet* existen dos escenarios, los que utilizan IPv4 mediante el comando *telnet localhost* y el puerto TCP correspondiente y los que utilizan IPv6 mediante el comando *telnet ::1* y el puerto TCP correspondiente.
- La elaboración de las prácticas de laboratorio influyó en el entendimiento del manejo de máquinas virtuales, así como en el tema de redes, analizando los diferentes protocolos permitiendo una mejor destreza en la ejecución de estos mediante comandos sencillos.
- El uso del *software Quagga* permitió en gran medida ampliar el conocimiento de direccionamiento y enrutamiento de redes ya que la plataforma permite la misma ejecución de un *router* convencional con la diferencia que está sobre un sistema operativo de licencia libre que deriva en ahorro de recursos económicos.
- Los resultados obtenidos mediante el envío de paquetes fueron exitosos en cuanto a la conexión entre los dispositivos que fueron configurados tanto en la virtualización como en la implementación física.
- En la implementación física, el uso de interfaces de comunicación difiere de los puertos disponibles en la *Raspberry Pi* por lo que es necesario emplear NIC's con conexión a USB para que el sistema reconozca una nueva interfaz de



comunicación diferente a la que está por defecto en el sistema, lo que no ocurre de forma virtual ya que mediante configuraciones simples se pueden adaptar las interfaces necesarias para su respectiva conexión.

- La implementación física en el sistema operativo Raspbian, no se diferencia de la instalación virtual, solo tiene un único aspecto que es el nombre de las interfaces de comunicaciones, *enp0s* para *Ubuntu* y *eth* para Raspbian.
- El enrutamiento estático usa siempre la misma ruta o camino para el envío de paquetes lo que reduce el tiempo de procesamiento del *router* lo que conlleva a reducir el tamaño de la tabla de enrutamiento.
- En el protocolo BGP todo el conjunto de enrutadores que en este caso conformaría un sistema autónomo es una clave muy importante ya que distingue a un conjunto de redes que son comandados bajo la misma administración técnica.
- El protocolo de enrutamiento dinámico BGP por su flexibilidad de funcionamiento es la opción por excelencia para comunicar todo el tráfico hacia Internet, así que es el más usado por la mayoría de los proveedores de Internet en el mundo.
- El protocolo que soporta direcciones en IPv6 para el demonio de *Quagga* OSPF6d es la versión OSPFv3.
- Las direcciones IPv4 son las más usadas en el mundo, pero debido a sus limitaciones se creó IPv6 que ofrecen más direcciones que IPv4.

## 4.2 Recomendaciones

- Al momento de la creación de la máquina virtual se debe tomar en cuenta que el tamaño de la memoria RAM no puede ser muy pequeña (mínimo sugerido 1Gb) ya que se va a trabajar en la parte de enrutamiento por lo que el tiempo de respuesta tiene que ser óptimo.
- Considerar las interfaces necesarias para la cantidad de dispositivos a utilizarse tanto para la virtualización mediante configuraciones como para implementación mediante NIC's con conexión a USB.
- Antes de comenzar con la ejecución del enrutamiento se debe tener en cuenta que la configuración sea la necesaria para los adaptadores en las máquinas virtuales creadas ya que de eso depende que la conexión sea exitosa.
- Se debe tomar en cuenta que al momento del diseño de la topología de red se segmente adecuadamente dicha red para prevenir conflictos de enrutamiento y direccionamiento.

- Se recomienda realizar las prácticas no solo de manera virtual sino también en la parte física ya que de ese modo se tendrá una perspectiva más amplia y se podrá visualizar los puntos más importantes del manejo de los protocolos y los elementos de conexión como *patchcord* e interfaces de comunicación.
- Al momento de la configuración del programa *Quagga* se sugiere instalar el paquete de *nmap* ya que con esto se podrá verificar el estado de cada protocolo al momento del enrutamiento.
- Utilizar la *Raspberry Pi* a plenitud (programación, proyectos IoT) ya que la configuración para utilizarlo como *router*, no cancela sus características iniciales del sistema.
- Para la implementación física se recomienda el uso de la tarjeta *Raspberry Pi 4* de 2 Gb de RAM la cual sería suficiente para aportar en la efectividad de los tiempos de respuesta.
- Al momento de adquirir el adaptador USB a RJ45 se recomienda usar el de tipo 3.0 ya que las especificaciones aclaran que es de alta velocidad para que las conexiones entre las tarjetas sean más eficaces.
- Para la implementación física se recomienda revisar el datasheet de las tarjetas *Raspberry Pi* a utilizarse para tomar en cuenta las características y especificaciones de los periféricos disponibles y las conexiones respectivas que deben realizarse.
- En cuanto a lo físico se recomienda armar la carcasa de la tarjeta *Raspberry Pi* para una manipulación más adecuada y evitar daños a los elementos de dicha tarjeta.
- Se recomienda tener equipos individuales (mouse, pantalla, teclado) para cada tarjeta de modo que se pueda tener una mejor visualización, distinción y comprobación de resultados.
- Para el desarrollo de la implementación física se recomienda usar la distribución Raspbian debido a que este es el Sistema Operativo por defecto para la tarjeta *Raspberry Pi*.
- Al momento de realizar las configuraciones de la máquina virtual se recomienda seleccionar el idioma inglés ya que todos los comandos a ejecutarse están en inglés.
- En caso de crear los *patchcords* para la conexión de las tarjetas *Raspberry Pi*, se recomienda realizar las pruebas respectivas de conexión de todos los pines ya que esto implicaría una falla de conexión al momento de realizar un *ping*.

## 5 REFERENCIAS BIBLIOGRÁFICAS

- [1] A. Gutierrez, «Qué es una imagen ISO,» Dotdash, 01 Noviembre 2019. [En línea]. Available: <https://www.aboutspanol.com/que-es-una-imagen-iso-3507896>. [Último acceso: 07 Julio 2021].
- [2] *Raspberry Pi*, «*Raspberry Pi 4*,» MCI Electronics, 23 octubre 2017. [En línea]. Available: <https://raspberrypi.cl/raspberry-pi-4/>. [Último acceso: 15 septiembre 2021].
- [3] C. Rus, «*Raspberry Pi 4* es oficial,» XATAKA, 25 julio 2019. [En línea]. Available: <https://www.xataka.com/ordenadores/raspberry-pi-4-caracteristicas-precio-ficha-tecnica>. [Último acceso: 15 septiembre 2021].
- [4] M. Gonzalez, «Direccionamiento IPv4,» Redes Telematicas, 17 diciembre 2012. [En línea]. Available: <https://redestelematicas.com/direccionamiento-ipv4/>. [Último acceso: 24 septiembre 2021].
- [5] IPv6 MX, «IPv4 vs IPv6,» Network Information Center México S.C., 14 mayo 2018. [En línea]. Available: <http://www.ipv6.mx/index.php/component/content/article/189-ipv4-vs-ipv6-icual-es-la-diferencia>. [Último acceso: 24 septiembre 2021].
- [6] Oracle Homeage, «Descripción general de las direcciones IPv6,» Oracle Corporation and/or its affiliates, 10 agosto 2010. [En línea]. Available: <https://docs.oracle.com/cd/E19957-01/820-2981/ipv6-overview-10/index.html>. [Último acceso: 24 septiembre 2021].
- [7] Marcelo, «Distancia Administrativa y Métrica,» CCNA desde cero, 10 agosto 2020. [En línea]. Available: <https://ccnadesdecero.com/curso/distancia-administrativa-y-metrica/>. [Último acceso: 24 septiembre 2021].
- [8] CCNA2, «La evolución de los protocolos de enrutamiento dinámico,» reuter.com, 15 abril 2015. [En línea]. Available: [https://www.reuter.com.ar/CCNA/CCNA2/mod7\\_ccna2/](https://www.reuter.com.ar/CCNA/CCNA2/mod7_ccna2/). [Último acceso: 24 septiembre 2021].

- [9] CCNA2, «Metricas,» Google Sites, 12 noviembre 2015. [En línea]. Available: <https://sites.google.com/site/cursoccna22015/unidad-3-protocolos-de-enrutamiento-dinamico/3-3-metricas>. [Último acceso: 24 septiembre 2021].
- [10] CCNA2 REDES\_2, «RIP versión 2,» Sites Google, 7 Agosto 2016. [En línea]. Available: <https://sites.google.com/site/asmccna2redes2/7-rip-version-2>. [Último acceso: 15 Junio 2021].
- [11] R. Morales, «*Quagga* - Enrutamiento RIP,» ticARTE, 03 Marzo 2014. [En línea]. Available: <https://www.ticarte.com/contenido/quagga-enrutamiento-rip>. [Último acceso: 15 Junio 2021].
- [12] e. a. Kunihiro Ishiguro, «RIPng,» *QUAGGA*, 19 Abril 2005. [En línea]. Available: <https://www.quagga.net/docs/docs-multi/RIPng.html>. [Último acceso: 15 Junio 2021].
- [13] e. a. Kunihiro Ishiguro, «*ripngd* Configuration,» *QUAGGA*, 19 Abril 2005. [En línea]. Available: <https://www.quagga.net/docs/docs-multi/ripngd-Configuration.html#ripngd-Configuration>. [Último acceso: 15 Junio 2021].
- [14] EvaCastro, «OSPF en *Quagga*,» GitBook, 25 Mayo 2013. [En línea]. Available: [https://evacastro.gitbooks.io/internet/content/ospf\\_en\\_quagga.html](https://evacastro.gitbooks.io/internet/content/ospf_en_quagga.html). [Último acceso: 16 Junio 2021].
- [15] e. a. Kunihiro Ishiguro, «OSPFv2,» *QUAGGA*, 19 Abril 2005. [En línea]. Available: <https://www.quagga.net/docs/docs-multi/OSPFv2.html#OSPFv2>. [Último acceso: 15 Junio 2021].
- [16] R. Morales, «*Quagga* - Enrutamiento OSPF,» ticARTE, 13 Febrero 2014. [En línea]. Available: <https://www.ticarte.com/contenido/quagga-enrutamiento-ospf>. [Último acceso: 16 Junio 2021].
- [17] EvaCastro, «BGP en *Quagga*,» GitBook, 20 Marzo 2013. [En línea]. Available: [https://evacastro.gitbooks.io/internet/content/bgp\\_en\\_quagga.html](https://evacastro.gitbooks.io/internet/content/bgp_en_quagga.html). [Último acceso: 16 Junio 2021].
- [18] V. Gapon, «Setting up BGP in *Quagga*,» IT Blog, 10 Noviembre 2018. [En línea]. Available: <https://ixnfo.com/en/setting-up-bgp-in-quagga.html>. [Último acceso: 16 Junio 2021].

- [19] e. a. Kunihiro Ishiguro, «BGP router,» QUAGGA, 19 Abril 2005. [En línea]. Available: <https://www.quagga.net/docs/docs-multi/BGP-router.html>. [Último acceso: 16 Junio 2021].
- [20] J. Scaife, «Introduction to BGP Routing and Security,» JamieWeb, 01 Junio 2020. [En línea]. Available: <https://www.jamieweb.net/blog/bgp-routing-security-part-1-bgp-peering-with-quagga/>. [Último acceso: 16 Junio 2021].
- [21] e. a. Kunihiro Ishiguro, «ISIS router,» [En línea]. Available: <https://www.nongnu.org/quagga/docs/docs-multi/ISIS.html>. [Último acceso: 17 Junio 2021].
- [22] DIGITAL GUIDE, «Hypervisor: el medio para la virtualización,» IONOS, 06 agosto 2020. [En línea]. Available: <https://www.ionos.es/digitalguide/servidores/know-how/que-es-un-hypervisor/>. [Último acceso: 27 septiembre 2021].
- [23] F. P. Gomez, «¿Qué es la virtualización?,» fpngenred.es, 12 julio 2020. [En línea]. Available: [https://www.fpgenred.es/VirtualBox/qu\\_es\\_la\\_virtualizacin.html](https://www.fpgenred.es/VirtualBox/qu_es_la_virtualizacin.html). [Último acceso: 27 septiembre 2021].
- [24] A. E. Ortiz, «¿Cual es el hipervisor bare metal?,» HostDimeBlog, 22 diciembre 2017. [En línea]. Available: <https://blog.hostdime.com.co/cual-es-el-hipervisor-bare-metal/>. [Último acceso: 27 septiembre 2021].
- [25] e. a. Kunihiro Ishiguro, «Quagga Routing Suite,» 28 Enero 2018. [En línea]. Available: <https://www.nongnu.org/quagga/>. [Último acceso: 7 Julio 2021].
- [26] Y. Fernández, «API: qué es y para qué sirve,» XATAKA, 23 agosto 2019. [En línea]. Available: <https://www.xataka.com/basics/api-que-sirve>. [Último acceso: 19 octubre 2021].
- [27] Conpilar News, «Cómo configurar Quagga Routing Suite en Linux,» conpilar.es, 4 Marzo 2021. [En línea]. Available: <https://conpilar.es/como-configurar-quagga-routing-suite-en-linux/>. [Último acceso: 7 Julio 2021].

## **ANEXOS**

**ANEXO 1: CERTIFICADO DE FUNCIONAMIENTO**



# ESCUELA POLITECNICA NACIONAL

Campus Politécnico "J. Rubén Orellana R

Quito, 10 de febrero de 2022

## CERTIFICADO DE FUNCIONAMIENTO DE PROYECTO DE TITULACIÓN

Yo, *nombres y apellidos del director*, docente a tiempo completo de la Escuela Politécnica Nacional y como director de este trabajo de titulación, certifico que he constatado el correcto funcionamiento de los puntos de energía seguros para protección de equipo electrónico en la ESFOT – EPN de la Zona 2, los cuales fueron implementados por los estudiantes Angelo Orellana y Pablo Puma.

El proyecto cumple con los requerimientos de diseño y parámetros necesarios para que los usuarios de la ESFOT puedan usar las instalaciones con seguridad para los equipos y las personas.

---

**DIRECTOR**

Ing. Nombres y apellidos., Msc.

---

Ladrón de Guevara E11-253, Escuela de Formación de Tecnólogos, Oficina 28. EXT: 2729  
email: pablo.proano@epn.edu.ec

Quito-Ecuador



## **ANEXO 2: PRÁCTICAS DE LABORATORIO**



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA ESTUDIANTE

#### PRÁCTICA 1

1. **TEMA:** Instalación y configuración de máquinas virtuales y *software* de Routing *Quagga*.

#### 2. OBJETIVOS

##### 2.1. OBJETIVO GENERAL

- Identificar los procesos adecuados para el correcto funcionamiento virtual de las máquinas virtuales y *Quagga*.

##### 2.2. OBJETIVOS ESPECÍFICOS

- Detallar el procedimiento para la creación y configuración de una máquina virtual en *Virtual Box TM*.

- Conocer los fundamentos sobre *Software* Virtual basados en licencia libre.

- Seguir el procedimiento y manejo de archivos para la instalación de *Quagga* en *Ubuntu* 18.04.4.

**Nota:** Referenciar todo acerca de la práctica 1 ya que ésta servirá como guía de funcionamiento tanto para la configuración de las máquinas virtuales como el desarrollo de las posteriores prácticas.

#### 3. TRABAJO PREPARATORIO

##### 3.1. CUESTIONARIO

- Consultar los requerimientos mínimos para la ejecución óptima de un sistema operativo de licencia libre en *Virtual Box TM*.

- Consultar sobre qué tratan los siguientes términos:

- *Software Quagga*
- SDN

- Consultar los requerimientos mínimos para la instalación del *Software Quagga* en un sistema operativo de licencia libre.

#### 4. DESCRIPCIÓN DE ACTIVIDADES Y PROCEDIMIENTO DE LA PRÁCTICA

4.1 Descargar la imagen ISO de *Ubuntu* 18.04.4 (preferentemente de su repositorio oficial)

4.2. Seguir el proceso de creación y configuración de la máquina virtual con los requerimientos necesarios para el correcto funcionamiento de la distribución seleccionada en 2 máquinas virtuales independientes.



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA ESTUDIANTE

**4.3.** Instalación de los paquetes necesarios y manejo de archivos para que el *software* de *routing* se ejecute en toda su plenitud (simultáneamente seguir el proceso en las 2 máquinas virtuales creadas).

**4.4.** Iniciar sesión con *telnet* con todos los protocolos de enrutamiento del sistema y su demonio central (*zebra*).

#### **5. INFORME**

**5.1.** Enumerar las ventajas de usar un *software* de *routing* sobre un sistema operativo de licencia libre.

**5.2.** Investigar si existen más *software* de *routing* de licencia libre, de ser así, describa sintetizada mente en qué consisten.

**5.3** ¿Qué es el Hipervisor de tipo 1 y 2?

**5.4** ¿Consultar cómo se asigna una dirección IP en un *router* de marca *Cisco*?



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA ESTUDIANTE

#### PRÁCTICA 2

1. **TEMA:** Práctica de Routing con rutas estáticas en *Quagga*, simulando una Red WAN.

#### 2. OBJETIVOS

##### 2.1. OBJETIVO GENERAL

- Realizar un enrutamiento en la siguiente red, aplicando rutas estáticas entre 2 sistemas independientes comandados por *Quagga*.

##### 2.2. OBJETIVOS ESPECÍFICOS

- Aplicar conceptos sobre rutas estáticas en direccionamiento y enrutamiento en redes WAN.
- Verificar los cambios en tabla de enrutamiento del router *Quagga* antes y después del ruteo.
- Analizar el beneficio de la aplicación de rutas estáticas en redes WAN.

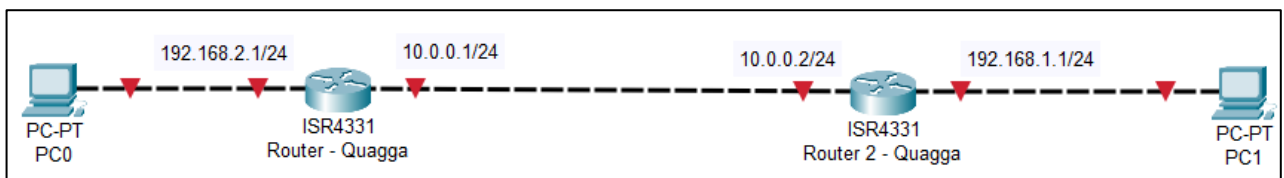
#### 3. TRABAJO PREPARATORIO

##### 3.1. CUESTIONARIO

- Consultar sobre los tipos de direcciones de red.
- Consultar las ventajas de la aplicación de rutas estáticas en redes WAN.
- La interfaz “vtysh”, ¿Qué representa en el *Software Quagga*?
- Consultar el puerto TCP con el que se hace sesión *Telnet* con el demonio principal “zebra”.

#### 4. DESCRIPCIÓN DE ACTIVIDADES Y PROCEDIMIENTO DE LA PRÁCTICA

4.1 Seguir la topología de red propuesta mostrada en la Figura 1.



Figura

4.2. Utilizar los adaptadores de red enp0s3 y enp0s8 (preferentemente en RED 1) en cada *router Quagga* y asignar las direcciones IP propuestas en dichas interfaces de comunicación.

4.3. Aplicar rutas estáticas para la comunicación entre los 2 *routers Quagga* en la interfaz principal (*zebra*).

4.4. Comprobar la comunicación entre *routers* y las interfaces de salida con mensajes PING.

4.5 (OPCIONAL) Replicar el ejercicio en los módulos de *Raspberry Pi 4* disponibles en el laboratorio para verificar la aplicación de conocimientos en equipos reales.

#### 5. INFORME



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA ESTUDIANTE

- 5.1. ¿Consultar los comandos utilizados con rutas estáticas en *routers* CISCO para la misma topología de red?
- 5.2. ¿Cuál es la diferencia de realizar el enrutamiento estático con *zebra* y con el *vtys*?
- 5.3 El enrutamiento dinámico según sus propias palabras es...
- 5.4 Realice el mismo enrutamiento, con diferentes direcciones IP y capture las tablas de enrutamiento después de efectuado el ejercicio.



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA ESTUDIANTE

#### PRÁCTICA 3

1. **TEMA:** Práctica de Routing con el protocolo de enrutamiento dinámico RIP en *Quagga*, simulando una Red WAN.

#### 2. OBJETIVOS

##### 2.1. OBJETIVO GENERAL

- Realizar un ruteo aplicando el protocolo de enrutamiento dinámico RIP entre 2 sistemas independientes comandados por *Quagga*.

##### 2.2. OBJETIVOS ESPECÍFICOS

- Comparar el contenido teórico y práctico del protocolo de enrutamiento RIP con direcciones IPV4

- Reconocer el tiempo que tarda la compartición de la tabla de enrutamiento entre routers con el protocolo RIP

- Determinar si el protocolo RIP tiene mayores beneficios en redes pequeñas o redes más grandes.

#### 3. TRABAJO PREPARATORIO

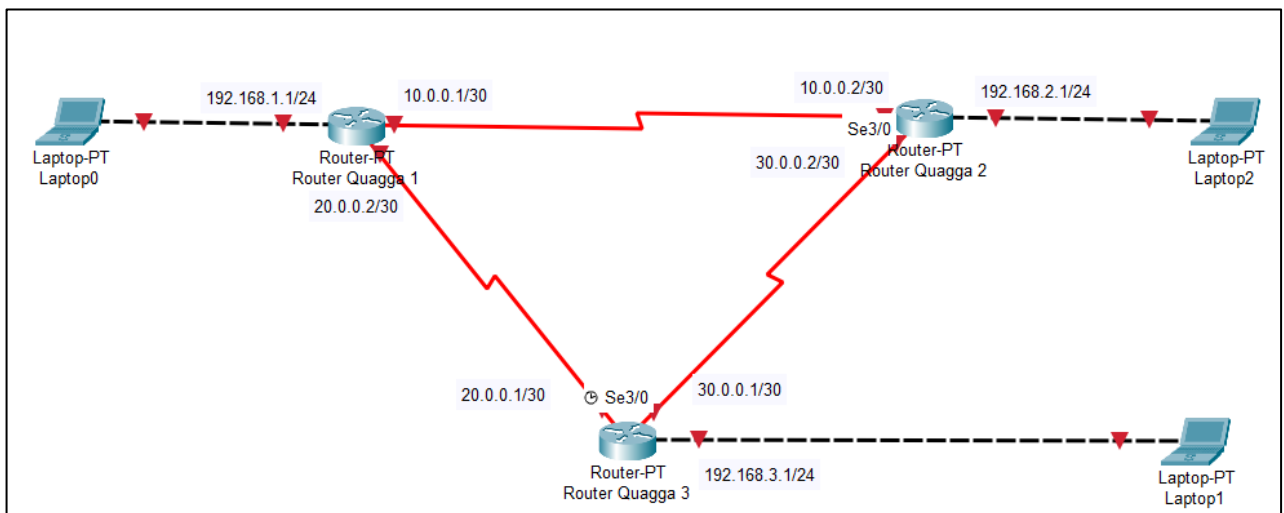
##### 3.1. CUESTIONARIO

- Consultar la métrica del protocolo de enrutamiento RIP.

- Consultar el puerto TCP con el que se hace sesión *Telnet* con el demonio *ripd*.

#### 4. DESCRIPCIÓN DE ACTIVIDADES Y PROCEDIMIENTO DE LA PRÁCTICA

4.1 Seguir la topología de red propuesta mostrada en la Figura 1.



4.2. Utilizar los adaptadores de red enp0s3, enp0s8 y enp0s9 (preferentemente en RED 1) en cada *router Quagga* y asignar las direcciones IP propuestas en dichas interfaces de comunicación.



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA ESTUDIANTE

**4.3.** Aplicar el protocolo RIP para la comunicación entre los 2 *routers Quagga* en la interfaz principal (*zebra*) para la asignación de direcciones IP, y en el respectivo demonio *ripd*.

**4.4.** Comprobar la comunicación entre *routers* y las interfaces de salida con mensajes PING.

**4.5 (OPCIONAL)** Replicar el ejercicio en los módulos de *Raspberry Pi 4* disponibles en el laboratorio para verificar la aplicación de conocimientos en equipos reales.

#### **5. INFORME**

**5.1.** Escribir 2 diferencias del enrutamiento entre rutas estáticas y RIP

**5.2.** Consultar la métrica y la distancia administrativa de RIP.

**5.3.** ¿Entre el enrutamiento estático y enrutamiento dinámico cuál es mejor? ¿Por qué?



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA ESTUDIANTE

#### PRÁCTICA 4

1. **TEMA:** Práctica de Routing con el protocolo de enrutamiento dinámico OSPF en *Quagga*, simulando una Red WAN.

#### 2. OBJETIVOS

##### 2.1. OBJETIVO GENERAL

- Realizar un ruteo sencillo aplicando el protocolo de enrutamiento dinámico OSPF entre 2 sistemas independientes comandados por *Quagga*.

##### 2.2. OBJETIVOS ESPECÍFICOS

- Comparar el contenido teórico y práctico del protocolo de enrutamiento OSPF con direcciones IPV4
- Identificar el concepto de Sistema Autónomo (AS) en redes con el protocolo OSPF
- Determinar el nombre de la división de un sistema autónomo.

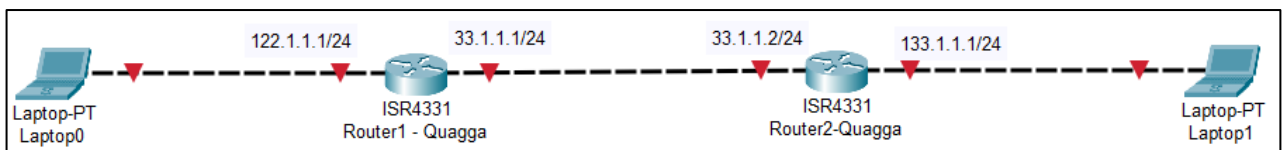
#### 3. TRABAJO PREPARATORIO

##### 3.1. CUESTIONARIO

- Consultar la métrica del protocolo de enrutamiento OSPF.
- ¿Cuál es el algoritmo con el cual OSPF calcula el camino más corto para llegar a los demás *routers* de la red? proponga un ejemplo gráfico.
- Consultar el puerto TCP con el que se hace sesión *Telnet* con el demonio *ospf6d*.

#### 4. DESCRIPCIÓN DE ACTIVIDADES Y PROCEDIMIENTO DE LA PRÁCTICA

4.1 Seguir la topología de red propuesta mostrada en la Figura 1.



4.2. Utilizar los adaptadores de red enp0s3 (Adaptador 1) y enp0s8 (Adaptador 2) (preferentemente en RED 1) en cada *router Quagga* y asignar las direcciones IP propuestas en dichas interfaces de comunicación.

4.3. Aplicar el protocolo OSPF para la comunicación entre los 2 *routers Quagga* en la interfaz principal (*zebra*) para la asignación de direcciones IP, y en el respectivo demonio *ospf6d*.

4.4. Comprobar la comunicación entre *routers* y las interfaces de salida con mensajes PING.

4.5 (OPCIONAL) Replicar el ejercicio en los módulos de *Raspberry Pi 4* disponibles en el laboratorio para verificar la aplicación de conocimientos en equipos reales.





**ESCUELA POLITÉCNICA NACIONAL**  
**ESCUELA DE FORMACIÓN DE TECNÓLOGOS**  
**HOJA GUÍA PARA ESTUDIANTE**

**5. INFORME**

- 5.1. Realizar un cuadro comparativo entre los protocolos OSPF y RIP
- 5.2. ¿Por qué OSPF es mejor que RIP?
- 5.3. ¿Cuál es la métrica y la distancia administrativa de OSPF?



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA ESTUDIANTE

#### PRÁCTICA 5

1. **TEMA:** Práctica de Routing con el protocolo de enrutamiento dinámico BGP en *Quagga*, simulando una Red WAN.

#### 2. OBJETIVOS

##### 2.1. OBJETIVO GENERAL

- Realizar un ruteo sencillo aplicando el protocolo de enrutamiento dinámico BGP entre 2 sistemas independientes comandados por *Quagga*.

##### 2.2. OBJETIVOS ESPECÍFICOS

- Analizar el contenido teórico para identificar las aplicaciones y ventajas basado en el protocolo BGP.
- Implementar el protocolo BGP para visualizar de mejor manera sus funciones y comportamiento.
- Especificar el concepto de sistema autónomo (AS) en una red de datos y la relación con el protocolo de enrutamiento BGP.

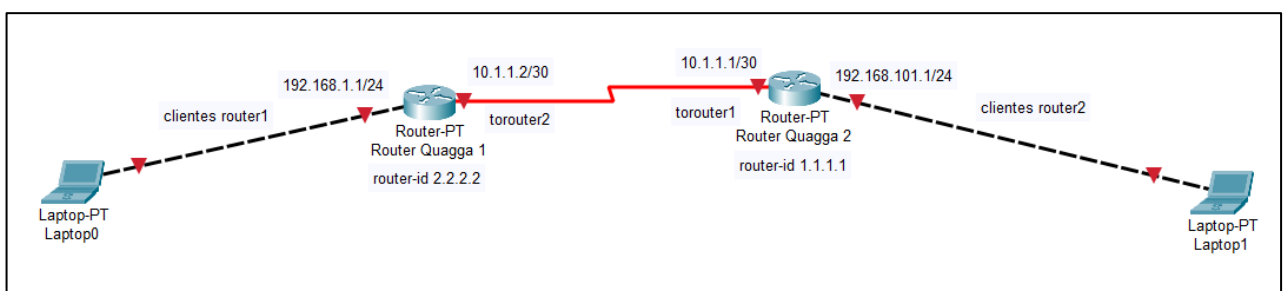
#### 3. TRABAJO PREPARATORIO

##### 3.1. CUESTIONARIO

- Consultar la métrica del protocolo de enrutamiento BGP.
- Consultar el puerto TCP con el que se hace sesión *Telnet* con el demonio *bgpd*.

#### 4. DESCRIPCIÓN DE ACTIVIDADES Y PROCEDIMIENTO DE LA PRÁCTICA

4.1 Seguir la topología de red propuesta mostrada en la Figura 1.



4.2. Utilizar los adaptadores de red enp0s3 y enp0s8 (preferentemente en RED 1) en cada *router Quagga* y asignar las direcciones IP propuestas en dichas interfaces de comunicación.

4.3. Aplicar el protocolo BGP para la comunicación entre los 2 *routers Quagga* en la interfaz principal (*zebra*) para la asignación de direcciones IP, y en el respectivo demonio *bgpd*.

4.4. Comprobar la comunicación entre *routers* y las interfaces de salida con mensajes PING.



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA ESTUDIANTE

**4.5 (OPCIONAL)** Replicar el ejercicio en los módulos de *Raspberry Pi 4* disponibles en el laboratorio para verificar la aplicación de conocimientos en equipos reales.

#### **5. INFORME**

**5.1.** Realizar un cuadro comparativo entre los protocolos BGP y OSPF

**5.2.** Consultar las características principales de BGP.

**5.3.** Consultar la métrica y la distancia administrativa de BGP.



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA ESTUDIANTE

#### PRÁCTICA 6

1. **TEMA:** Práctica de Routing con el protocolo de enrutamiento dinámico RIPng en *Quagga*, simulando una Red WAN.

#### 2. OBJETIVOS

##### 2.1. OBJETIVO GENERAL

- Realizar un ruteo sencillo aplicando el protocolo de enrutamiento dinámico RIP entre 2 sistemas independientes comandados por *Quagga*.

##### 2.2. OBJETIVOS ESPECÍFICOS

- Comparar la actualización del protocolo RIP en IPv4 con su versión en IPv6 (RIPng)
- Establecer los parámetros teóricos como actualización y expiración de rutas en la aplicación del protocolo.
- Comprobar la comunicación de extremo a extremo mediante mensajes simples (*ping*)
- Considerar las semejanzas en rutas RIP tanto para direcciones IPv4 como IPv6.

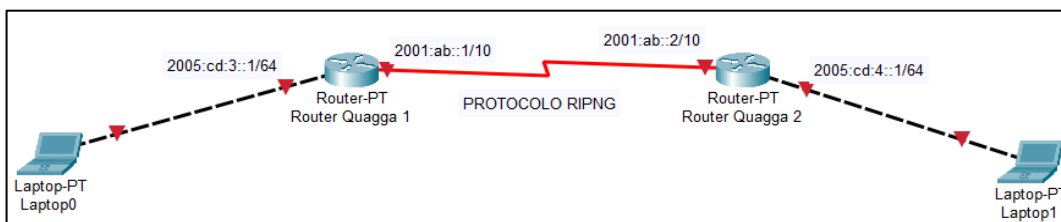
#### 3. TRABAJO PREPARATORIO

##### 3.1. CUESTIONARIO

- Consultar la métrica del protocolo de enrutamiento RIPng.
- Consultar el puerto TCP con el que se hace sesión *Telnet* con el demonio *ripngd*.

#### 4. DESCRIPCIÓN DE ACTIVIDADES Y PROCEDIMIENTO DE LA PRÁCTICA

4.1 Seguir la topología de red propuesta mostrada en la Figura 1.



4.2. Utilizar los adaptadores de red enp0s3 y enp0s8 (preferentemente en RED 1) en cada *router Quagga* y asignar las direcciones IP propuestas en dichas interfaces de comunicación.

4.3. Aplicar el protocolo RIP para la comunicación entre los 2 *routers Quagga* en la interfaz principal (*zebra*) para la asignación de direcciones IP, y en el respectivo demonio *ripngd*. (Nota: Realizarlo en el vtysh evitará realizar el cambio de sesión *telnet* entre *zebra* y *ripngd*, proceso que lo realizará el instructor)

4.4. Comprobar la comunicación entre *routers* y las interfaces de salida con mensajes PING.



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA ESTUDIANTE

**4.5 (OPCIONAL)** Replicar el ejercicio en los módulos de *Raspberry Pi 4* disponibles en el laboratorio para verificar la aplicación de conocimientos en equipos reales.

#### **5. INFORME**

**5.1.** ¿Cuáles son los puertos UDP por los que escuchan RIPng y RIP?

**5.2.** Consultar las características de RIPng y por qué se diferencia de RIP.

**5.3.** Consultar la métrica y la distancia administrativa de RIPng.



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA ESTUDIANTE

#### PRÁCTICA 7

1. **TEMA:** Práctica de Routing con el protocolo de enrutamiento dinámico OSPF6d en *Quagga*, simulando una Red WAN.

#### 2. OBJETIVOS

##### 2.1. OBJETIVO GENERAL

- Realizar un ruteo sencillo aplicando el protocolo de enrutamiento dinámico OSPF entre 2 sistemas independientes comandados por *Quagga*.

##### 2.2. OBJETIVOS ESPECÍFICOS

- Aplicar los conocimientos adquiridos en clase que se basan en el direccionamiento y enrutamiento básico de redes enfocado a OSPF con direcciones IPv6.

- Configurar los *routers Quagga* con las direcciones IPv6 y comandos propuestos por el instructor, siguiendo la misma sintaxis de la marca *Cisco*.

- Analizar las semejanzas y diferencias entre las versiones disponibles del protocolo OSPF con la correlación de direcciones IPv4 e IPv6.

#### 3. TRABAJO PREPARATORIO

##### 3.1. CUESTIONARIO

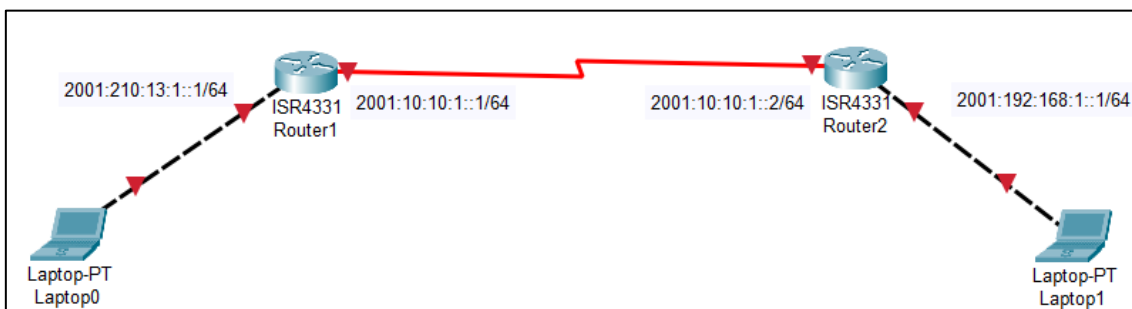
- Consultar la métrica del protocolo de enrutamiento OSPFv3.

- ¿Cuál es el algoritmo con el cual OSPF calcula el camino más corto para llegar a los demás *routers* de la red? proponga un ejemplo gráfico.

- Consultar el puerto TCP con el que se hace sesión *Telnet* con el demonio *ospf6d*.

#### 4. DESCRIPCIÓN DE ACTIVIDADES Y PROCEDIMIENTO DE LA PRÁCTICA

4.1 Seguir la topología de red propuesta mostrada en la Figura 1.



4.2. Utilizar los adaptadores de red *enp0s3* (Adaptador 1) y *enp0s8* (Adaptador 2) (preferentemente en RED 1) en cada *router Quagga* y asignar las direcciones IPv6 propuestas en dichas interfaces de comunicación.



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA ESTUDIANTE

**4.3.** Aplicar el protocolo OSPF para la comunicación entre los 2 *routers Quagga* en la interfaz principal (*zebra*) para la asignación de direcciones IPv6, y en el respectivo demonio *ospf6d*.

**4.4.** Comprobar la comunicación entre *routers* y las interfaces de salida con mensajes PING.

**4.5 (OPCIONAL)** Replicar el ejercicio en los módulos de *Raspberry Pi 4* disponibles en el laboratorio para verificar la aplicación de conocimientos en equipos reales.

#### **5. INFORME**

**5.1.** Realizar un cuadro comparativo entre los protocolos OSPFv2 y OSPFv3

**5.2.** Consultar las características esenciales de OSPFv3.

**5.3.** ¿Cuál es la métrica y la distancia administrativa de OSPFv3?



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA ESTUDIANTE

#### PRÁCTICA 8

1. **TEMA:** Práctica de Routing con el protocolo de enrutamiento dinámico BGP IPv6 en *Quagga*, simulando una Red WAN.

#### 2. OBJETIVOS

##### 2.1. OBJETIVO GENERAL

- Realizar un ruteo sencillo aplicando el protocolo de enrutamiento dinámico BGP con direcciones IPv6 entre 2 sistemas independientes comandados por *Quagga*.

##### 2.2. OBJETIVOS ESPECÍFICOS

- Emplear el contenido teórico para el protocolo BGP con direcciones IPv6.
- Analizar cual distancia administrativa es la óptima tanto para redes pequeñas como para redes grandes.
- Comprobar la comunicación de extremo a extremo.
- Comparar la diferencia o similitud del direccionamiento BGP con direcciones IPv4 e IPv6.

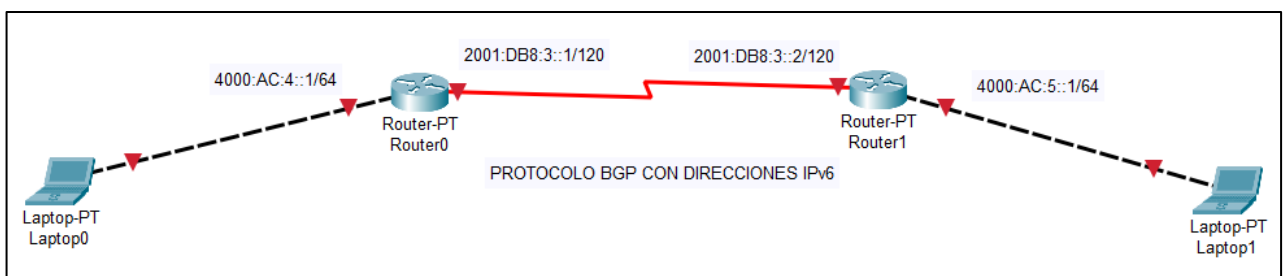
#### 3. TRABAJO PREPARATORIO

##### 3.1. CUESTIONARIO

- Consultar la métrica del protocolo de enrutamiento OSPFv3.
- ¿Cuál es el algoritmo con el cual OSPF calcula el camino más corto para llegar a los demás *routers* de la red? proponga un ejemplo gráfico.
- Consultar el puerto TCP con el que se hace sesión *Telnet* con el demonio *ospf6d*.

#### 4. DESCRIPCIÓN DE ACTIVIDADES Y PROCEDIMIENTO DE LA PRÁCTICA

4.1 Seguir la topología de red propuesta mostrada en la Figura 1.



4.2. Utilizar los adaptadores de red *enp0s3* (Adaptador 1) y *enp0s8* (Adaptador 2) (preferentemente en RED 1) en cada *router Quagga* y asignar las direcciones IPv6 propuestas en dichas interfaces de comunicación.

4.3. Aplicar el protocolo BGP para la comunicación entre los 2 *routers Quagga* en la interfaz principal (*zebra*) para la asignación de direcciones IPv6, y en el respectivo demonio *ospf6d*.





# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA ESTUDIANTE

**4.4.** Comprobar la comunicación entre *routers* y las interfaces de salida con mensajes PING.

**4.5 (OPCIONAL)** Replicar el ejercicio en los módulos de *Raspberry Pi 4* disponibles en el laboratorio para verificar la aplicación de conocimientos en equipos reales.

#### **5. INFORME**

**5.1.** Realizar un cuadro comparativo entre IPv4 BGP vs IPv6 BGP.

**5.2.** ¿Cuáles son los escenarios en los que se puede encontrar un enrutamiento BGP?

**5.2.** Diseñar una topología de red con 3 *routers* y sus respectivas direcciones IPv6 y realizar el enrutamiento BGP.



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

#### PRÁCTICA 1

1. **TEMA:** Instalación y configuración de máquinas virtuales y *software* de Routing *Quagga*

#### 2. DESARROLLO DE LA PRÁCTICA

**NOTA:** El tiempo para realizar la práctica es de 80 minutos. Es necesario que el alumno tome capturas de todo el proceso realizado en la práctica para la posterior evidencia en el informe ya que ésta servirá como guía de manejo para las demás prácticas tomando en cuenta que en esta se manejarán todos los archivos de configuración del *software Quagga* la cual funcionará como *router* para el enrutamiento de los protocolos.

#### CREACIÓN DE MAQUINA VIRTUAL

**NOTA:** Tener lista la imagen ISO de la distribución *UBUNTU* 18.04.4 en una memoria externa o descargarla inmediatamente del repositorio oficial.

**Paso 1:** Instalación de la distro *Ubuntu*. Crear una máquina virtual dando clic en nueva como se muestra en la Figura 1.

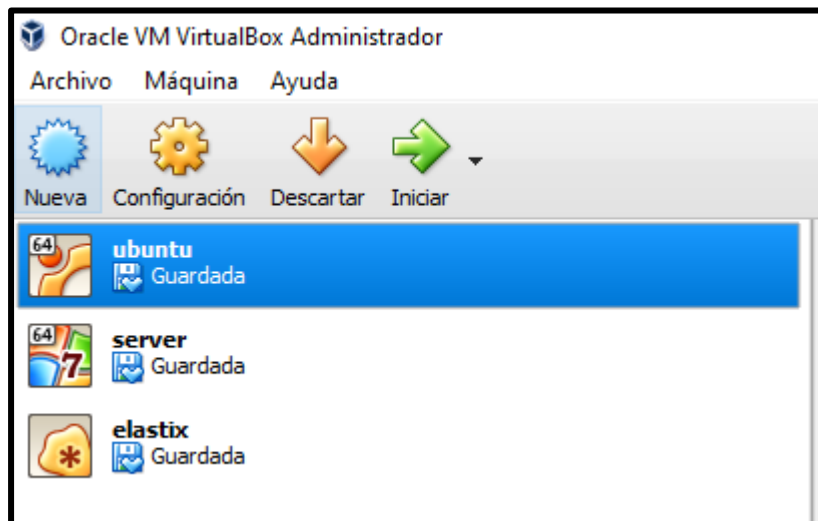


Figura 1. Creación de máquina virtual



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

**Paso 2:** A continuación, se asigna un nombre a la máquina virtual y se escoge la versión que en este caso es de **Ubuntu (64-bit)** como se muestra en la Figura 2.

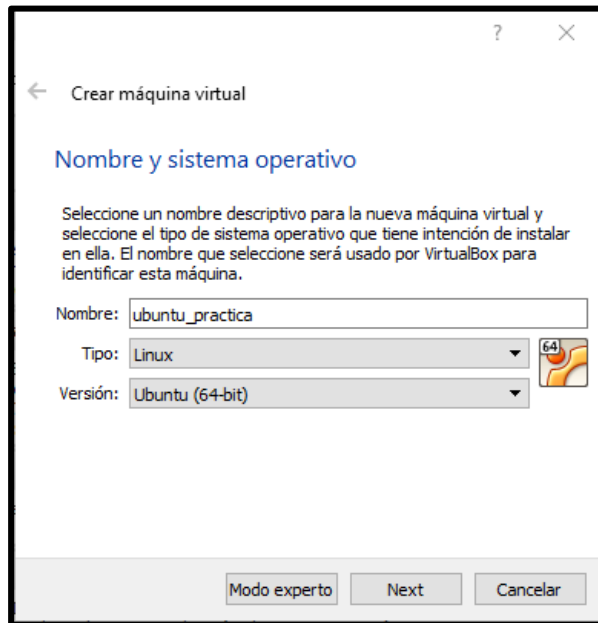
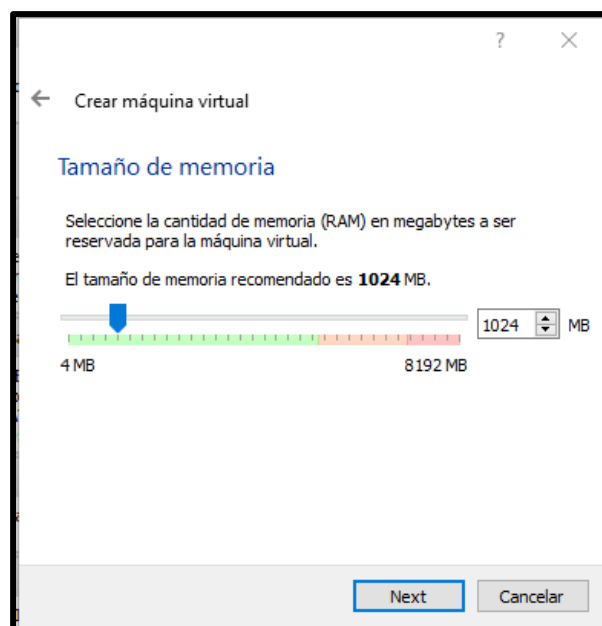


Figura 2. Asignación de nombre y sistema operativo

**Paso 3:** Luego se escoge el tamaño de la memoria RAM, se recomienda seleccionar un tamaño mínimo de 2GB para que el sistema opere con normalidad como se muestra en la Figura 3.





# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

Figura 3. Selección del tamaño de la memoria RAM

**Paso 4:** Después se elige la opción “**Crear un disco duro virtual ahora**” como se observa en la Figura 4.

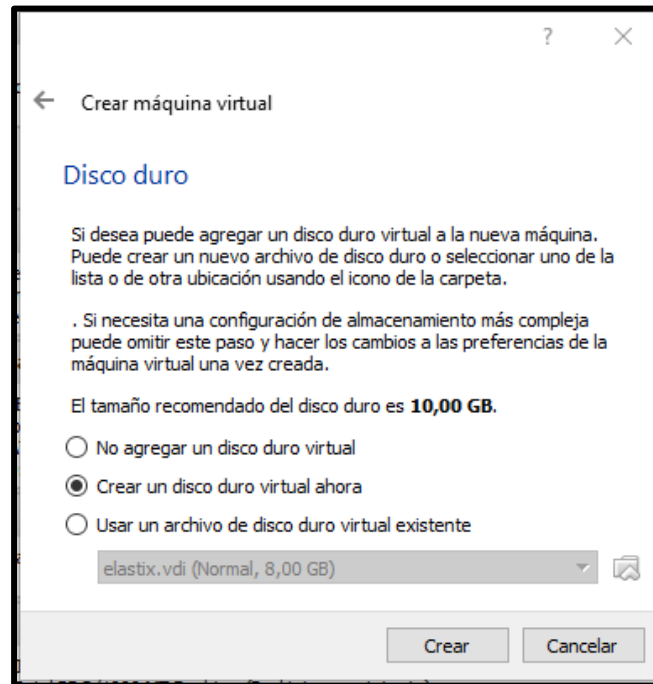
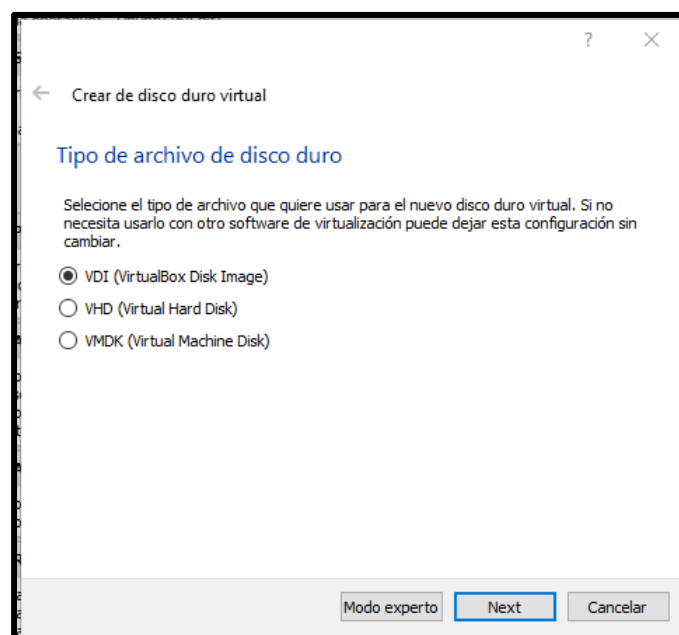


Figura 4. Creación del disco duro virtual

**Paso 5:** Luego se escoge el tipo de archivo VDI como se muestra en la Figura 5.





# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

Figura 5. Selección del tipo de archivo de disco duro

**Paso 6:** A continuación, se escoge el tipo de almacenamiento **Reservado dinámicamente** como se observa en la Figura 6.

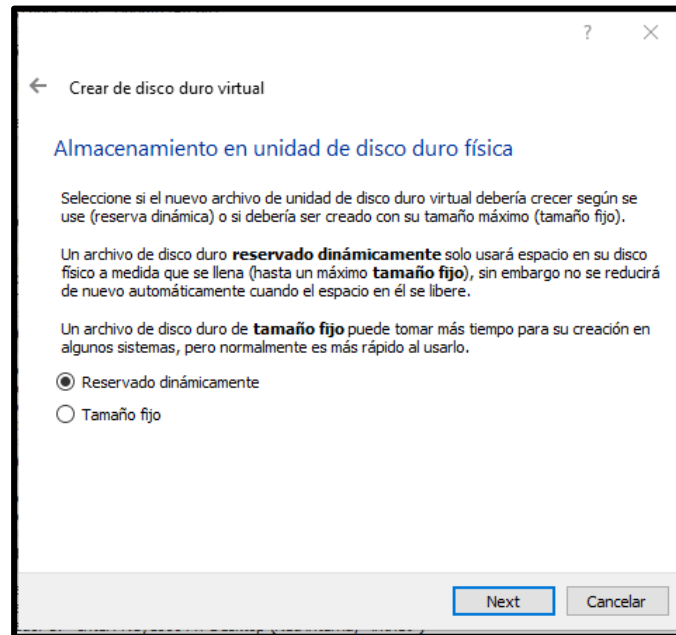
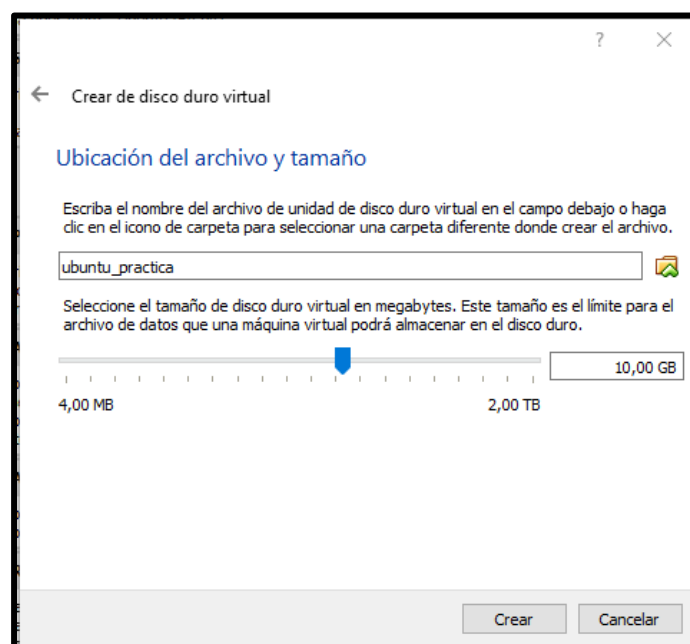


Figura 6. Selección de almacenamiento en unidad de disco duro física

**Paso 7:** Por último, escoger el tamaño del disco virtual que en este caso se recomienda duplicar el valor que da por defecto como se puede observar en la Figura 7.





# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

Figura 7. Asignación del tamaño del disco duro virtual

**Paso 8:** Una vez configurado las especificaciones de la máquina virtual, en la parte de configuración dirigirse a **Almacenamiento** y en la opción **Vacío** cargar la imagen de la distro *Ubuntu* como se muestra en la Figura 8.

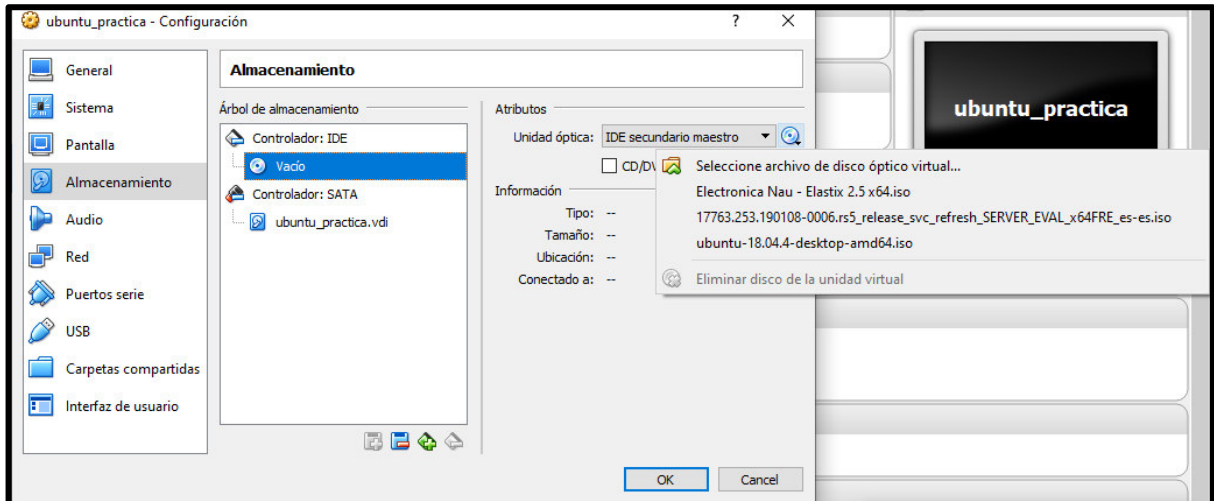
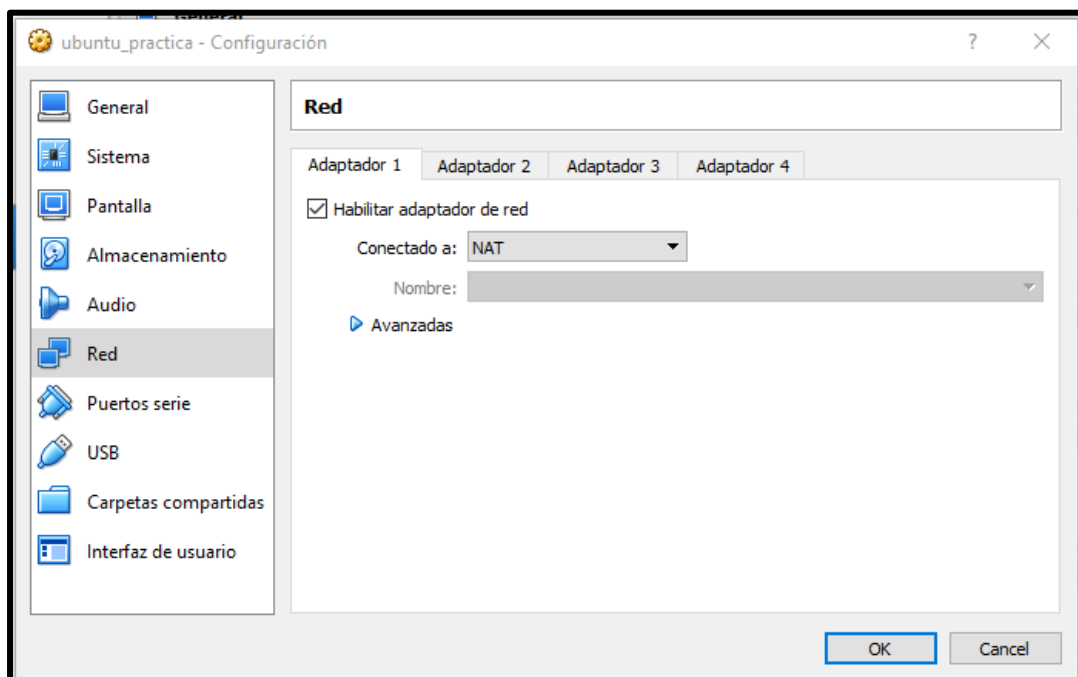


Figura 8. Selección de archivo de disco óptico virtual

**Paso 9:** Luego en la opción de Red habilitar todos los adaptadores y escoger la opción NAT en al menos uno para que pueda tener acceso a Internet como se muestra en la Figura 9.





# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

Figura 9. Habilitación de los adaptadores de red

**Paso 10:** Se ha creado una máquina virtual. Luego seleccionar la máquina virtual creada e iniciarla para comenzar su instalación como se muestra en la Figura 10.

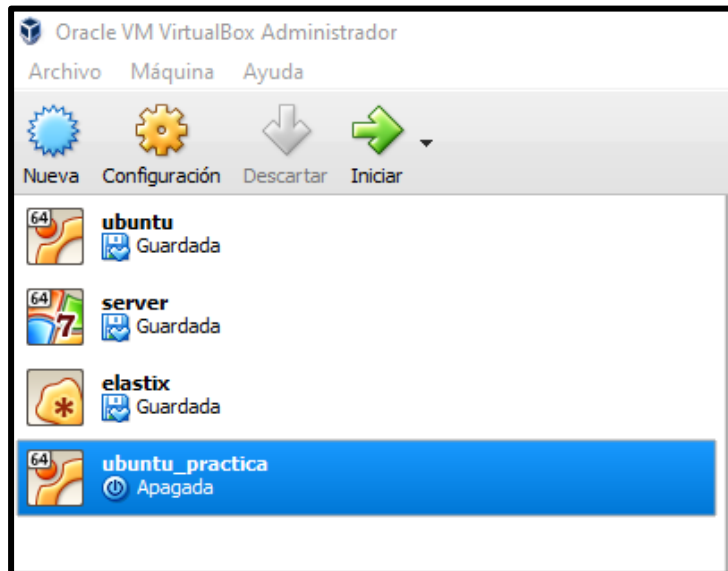
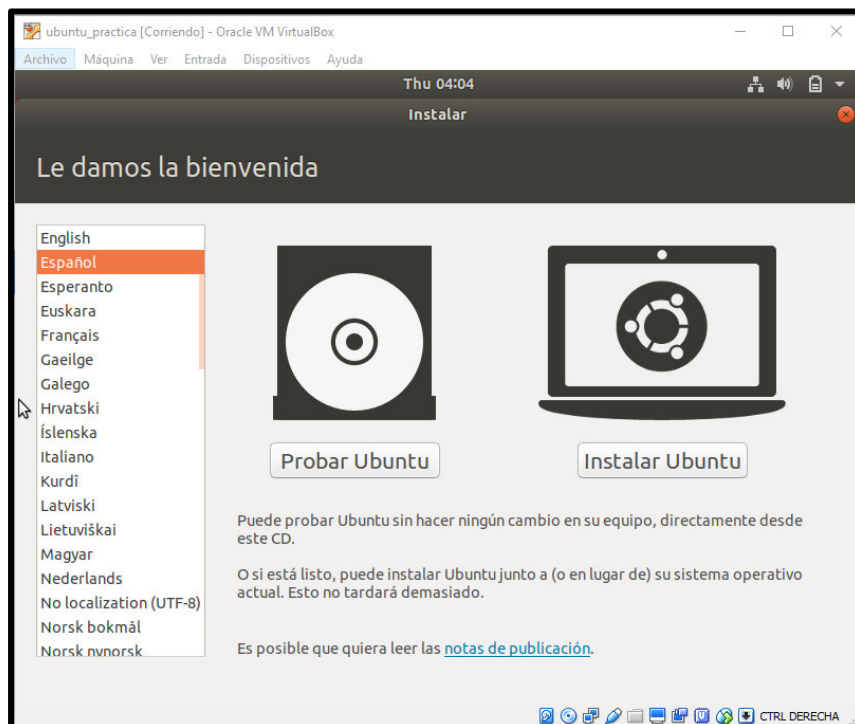


Figura 10. Iniciación de la máquina virtual creada

**Paso 11:** A continuación, se elige el idioma en el cual se va a configurar la máquina virtual, en este caso **Español** como se observa en la Figura 11.





# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

Figura 11. Selección del idioma

**Paso 12:** También se escoge el tipo del teclado con el cual se manejarán las diferentes funciones y comandos a ejecutarse como se muestra en la Figura 12.

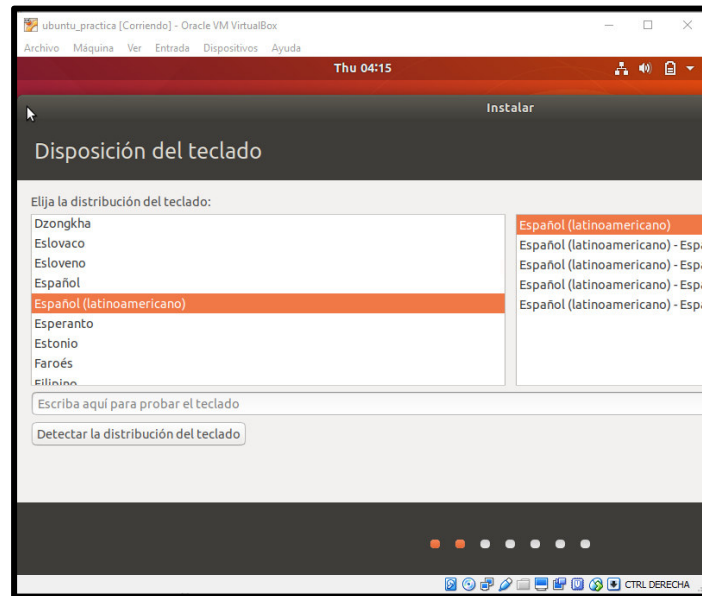
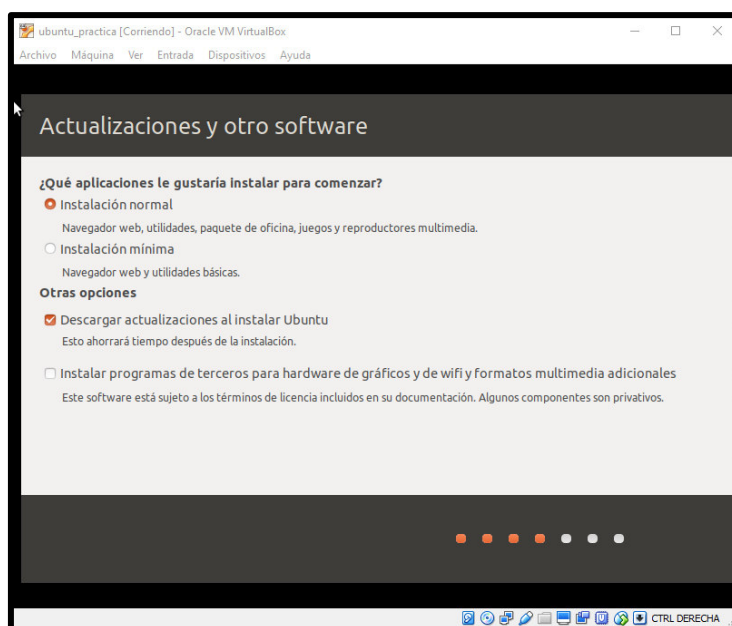


Figura 12. Selección del tipo de teclado

**Paso 13:** Luego seleccionar la opción **Instalación normal** y **Descargar actualizaciones al instalar Ubuntu** en caso de que existan nuevas actualizaciones como se muestra en la Figura 13.







# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

Figura 13. Selección del tipo de instalación y de las actualizaciones de la distribución

**Paso 14:** Después se escoge la opción **Borrar disco e instalar *Ubuntu*** como una manera de formatearlo como se muestra en la Figura 14.

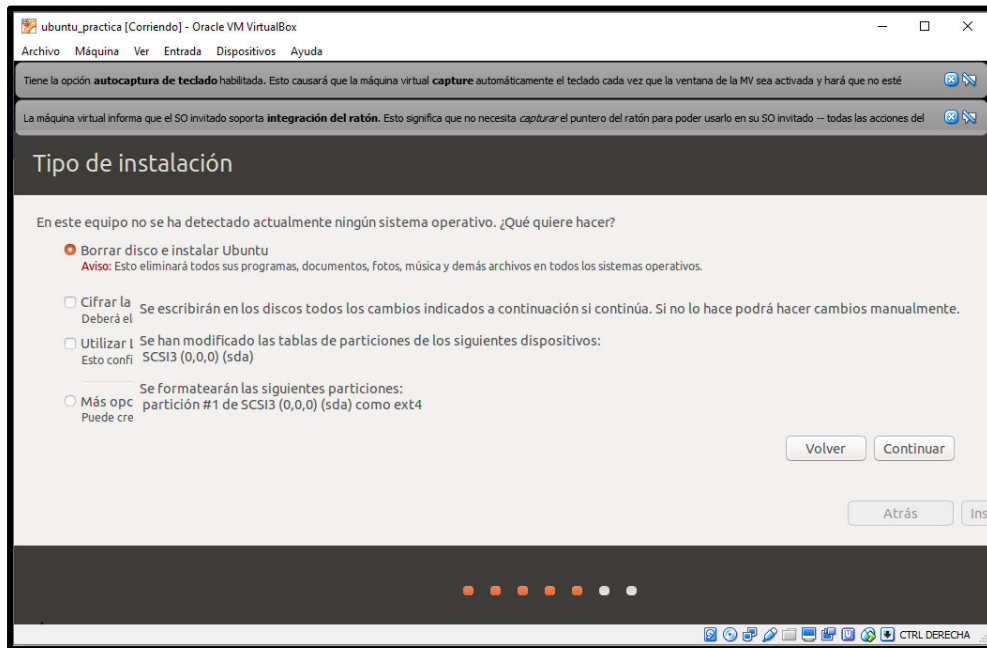
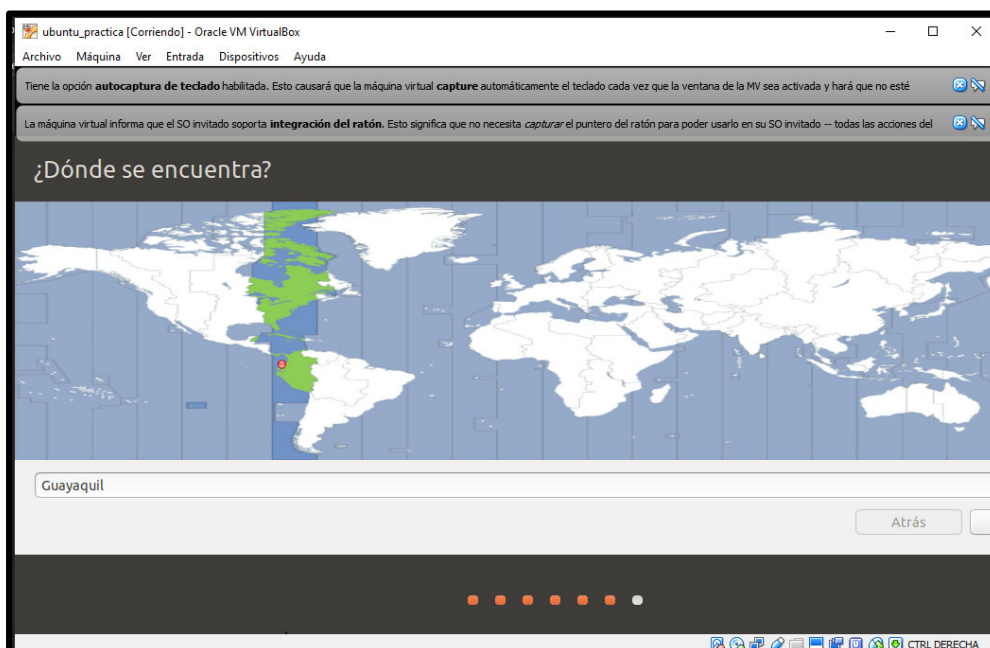


Figura 14. Selección del tipo de instalación en el disco virtual.

**Paso 15:** Luego se escoge la zona horaria, en este caso Guayaquil como se muestra en la Figura 15.





# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

Figura 15. Selección de la zona horaria

**Paso 16:** Por último, poner las credenciales con el nombre del equipo y una contraseña a conveniencia del usuario como se observa en la Figura 16.

Figura 16. Asignación de credenciales para la máquina virtual creada

### INSTALACIÓN DE QUAGGA

**Paso 1:** Descargar el paquete principal del *Software* el cual es *quagga* con el comando ***sudo apt-get install quagga\****

Se recomienda usar el parámetro (\*) para descargar todos los anexos referentes al paquete *Quagga*, tal como se aprecia en la Figura 1, se debe acceder como *root* para poder instalar el paquete.



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
tesis@tesis-VirtualBox:~$ sudo su
[sudo] contraseña para tesis:
root@tesis-VirtualBox:/home/tesis# sudo apt-get install quagga
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no
son necesarios.
  efibootmgr libegl1-mesa libfwup1 libllvm9 libwayland-egl1-mesa
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
  quagga-bgpd quagga-core quagga-isisd quagga-ospf6d quagga-ospfd quagga-pimd
  quagga-ripd quagga-ripngd
Paquetes sugeridos:
  snmpd
Se instalarán los siguientes paquetes NUEVOS:
  quagga quagga-bgpd quagga-core quagga-isisd quagga-ospf6d quagga-ospfd
  quagga-pimd quagga-ripd quagga-ripngd
0 actualizados, 9 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 1.256 kB de archivos.
Se utilizarán 5.504 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://ec.archive.ubuntu.com/ubuntu bionic/main amd64 quagga-core amd64 1
.2.4-1 [567 kB]
Des:2 http://ec.archive.ubuntu.com/ubuntu bionic/main amd64 quagga-bgpd amd64 1
.2.4-1 [258 kB]
```

```
Des:2 http://ec.archive.ubuntu.com/ubuntu bionic/main amd64 quagga-bgpd amd64 1
.2.4-1 [258 kB]
Des:3 http://ec.archive.ubuntu.com/ubuntu bionic/main amd64 quagga-isisd amd64
1.2.4-1 [111 kB]
Des:4 http://ec.archive.ubuntu.com/ubuntu bionic/main amd64 quagga-ospf6d amd64
1.2.4-1 [110 kB]
Des:5 http://ec.archive.ubuntu.com/ubuntu bionic/main amd64 quagga-ospfd amd64
1.2.4-1 [15,0 kB]
Des:6 http://ec.archive.ubuntu.com/ubuntu bionic/main amd64 quagga-pimd amd64 1
.2.4-1 [97,9 kB]
Des:7 http://ec.archive.ubuntu.com/ubuntu bionic/main amd64 quagga-ripd amd64 1
.2.4-1 [49,4 kB]
Des:8 http://ec.archive.ubuntu.com/ubuntu bionic/main amd64 quagga-ripngd amd64
1.2.4-1 [41,1 kB]
Des:9 http://ec.archive.ubuntu.com/ubuntu bionic/main amd64 quagga amd64 1.2.4-
1 [6.328 B]
Descargados 1.256 kB en 12s (108 kB/s)
Seleccionando el paquete quagga-core previamente no seleccionado.
(Leyendo la base de datos ... 165689 ficheros o directorios instalados actualme
nte.)
Preparando para desempaquetar .../0-quagga-core_1.2.4-1_amd64.deb ...
Añadiendo al usuario `quagga' al grupo `quaggavty' ...
Añadiendo al usuario quagga al grupo quaggavty
Hecho.
Desempaquetando quagga-core (1.2.4-1) ...
Seleccionando el paquete quagga-bgpd previamente no seleccionado.
Preparando para desempaquetar .../1-quagga-bgpd_1.2.4-1_amd64.deb ...
Desempaquetando quagga-bgpd (1.2.4-1) ...
Seleccionando el paquete quagga-isisd previamente no seleccionado.
```



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
Seleccionando el paquete quagga-isisd previamente no seleccionado.
Preparando para desempaquetar .../2-quagga-isisd_1.2.4-1_amd64.deb ...
Desempaquetando quagga-isisd (1.2.4-1) ...
Seleccionando el paquete quagga-ospf6d previamente no seleccionado.
Preparando para desempaquetar .../3-quagga-ospf6d_1.2.4-1_amd64.deb ...
Desempaquetando quagga-ospf6d (1.2.4-1) ...
Seleccionando el paquete quagga-ospfd previamente no seleccionado.
Preparando para desempaquetar .../4-quagga-ospfd_1.2.4-1_amd64.deb ...
Desempaquetando quagga-ospfd (1.2.4-1) ...
Seleccionando el paquete quagga-pimd previamente no seleccionado.
Preparando para desempaquetar .../5-quagga-pimd_1.2.4-1_amd64.deb ...
Desempaquetando quagga-pimd (1.2.4-1) ...
Seleccionando el paquete quagga-ripd previamente no seleccionado.
Preparando para desempaquetar .../6-quagga-ripd_1.2.4-1_amd64.deb ...
Desempaquetando quagga-ripd (1.2.4-1) ...
Seleccionando el paquete quagga-ripngd previamente no seleccionado.
Preparando para desempaquetar .../7-quagga-ripngd_1.2.4-1_amd64.deb ...
Desempaquetando quagga-ripngd (1.2.4-1) ...
Seleccionando el paquete quagga previamente no seleccionado.
Preparando para desempaquetar .../8-quagga_1.2.4-1_amd64.deb ...
Desempaquetando quagga (1.2.4-1) ...
Configurando quagga-core (1.2.4-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/zebra.service → /lib/systemd/system/zebra.service.
Configurando quagga-bgpd (1.2.4-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/bgpd.service → /lib/systemd/system/bgpd.service.
Configurando quagga-pimd (1.2.4-1) ...
```

```
Configurando quagga-pimd (1.2.4-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/pimd.service → /lib/systemd/system/pimd.service.
Configurando quagga-ripd (1.2.4-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ripd.service → /lib/systemd/system/ripd.service.
Configurando quagga-ripngd (1.2.4-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ripngd.service → /lib/systemd/system/ripngd.service.
Configurando quagga-ospf6d (1.2.4-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ospf6d.service → /lib/systemd/system/ospf6d.service.
Configurando quagga-isisd (1.2.4-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/isisd.service → /lib/systemd/system/isisd.service.
Configurando quagga-ospfd (1.2.4-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ospfd.service → /lib/systemd/system/ospfd.service.
Configurando quagga (1.2.4-1) ...
Procesando disparadores para man-db (2.8.3-2ubuntu0.1) ...
```



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
root@tesis-VirtualBox:/home/tesis# sudo apt-get install quagga*
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Nota, seleccionando «quagga-ripd» para el global «quagga*»
Nota, seleccionando «quagga-isisd» para el global «quagga*»
Nota, seleccionando «quagga-ospfd» para el global «quagga*»
Nota, seleccionando «quagga-doc» para el global «quagga*»
Nota, seleccionando «quagga-core» para el global «quagga*»
Nota, seleccionando «quagga-bgpd» para el global «quagga*»
Nota, seleccionando «quagga-pimd» para el global «quagga*»
Nota, seleccionando «quagga-ospf6d» para el global «quagga*»
Nota, seleccionando «quagga-ripngd» para el global «quagga*»
Nota, seleccionando «quagga» para el global «quagga*»
quagga ya está en su versión más reciente (1.2.4-1).
quagga-bgpd ya está en su versión más reciente (1.2.4-1).
fijado quagga-bgpd como instalado manualmente.
quagga-core ya está en su versión más reciente (1.2.4-1).
fijado quagga-core como instalado manualmente.
quagga-isisd ya está en su versión más reciente (1.2.4-1).
fijado quagga-isisd como instalado manualmente.
quagga-ospf6d ya está en su versión más reciente (1.2.4-1).
fijado quagga-ospf6d como instalado manualmente.
quagga-ospfd ya está en su versión más reciente (1.2.4-1).
fijado quagga-ospfd como instalado manualmente.
quagga-pimd ya está en su versión más reciente (1.2.4-1).
```

```
fijado quagga-pimd como instalado manualmente.
quagga-ripd ya está en su versión más reciente (1.2.4-1).
fijado quagga-ripd como instalado manualmente.
quagga-ripngd ya está en su versión más reciente (1.2.4-1).
fijado quagga-ripngd como instalado manualmente.
Los paquetes indicados a continuación se instalaron de forma automática y ya no
son necesarios.
efibootmgr libegl1-mesa libfwup1 libllvm9 libwayland-egl1-mesa
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes NUEVOS:
quagga-doc
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 6 no actualizados.
Se necesita descargar 868 kB de archivos.
Se utilizarán 928 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://ec.archive.ubuntu.com/ubuntu bionic/main amd64 quagga-doc all 1.2.
4-1 [868 kB]
Descargados 868 kB en 5s (181 kB/s)
Seleccionando el paquete quagga-doc previamente no seleccionado.
(Leyendo la base de datos ... 165773 ficheros o directorios instalados actualme
nte.)
Preparando para desempaquetar .../quagga-doc_1.2.4-1_all.deb ...
Desempaquetando quagga-doc (1.2.4-1) ...
Configurando quagga-doc (1.2.4-1) ...
Procesando disparadores para install-info (6.5.0.dfsg.1-2) ...
```

Figura 1. Instalación de paquete *Quagga* en *Ubuntu*

**Paso 2:** Una vez instalado todo el paquete de *Quagga*, se procede a instalar *telnet*, para poder crear las sesiones de configuración del *router*, con el comando `sudo apt-get install telnet`, tal como se aprecia en la Figura 2.



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
root@tesis-VirtualBox:/home/tesis# sudo apt-get install telnet
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
telnet ya está en su versión más reciente (0.17-41).
fijado telnet como instalado manualmente.
Los paquetes indicados a continuación se instalaron de forma automática y ya no
son necesarios.
  efibootmgr libegl1-mesa libfwup1 libllvm9 libwayland-egl1-mesa
Utilice «sudo apt autoremove» para eliminarlos.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 6 no actualizados.
```

Figura 2. Instalación de paquete *Telnet* en *Ubuntu*

**Paso 3:** Editar el archivo `/etc/sysctl.conf` y des comentar la línea `net.ipv4.ip_forward=1` y `net.ipv6.conf.all.forwarding=1`, para que el sistema, libere el tráfico por IPv4 e IPv6, como se visualiza en la Figura 3.

```
root@tesis-VirtualBox:/# nano /etc/sysctl.conf
```

```
# 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
```

Figura 3. Modificación de archivo para liberar tráfico IPv4 e IPv6

**Paso 4:** Copiar los archivos de cada demonio a un directorio individual, como en la Figura 4, con la siguiente estructura (`cp /usr/share/doc/quagga-core/examples/demonio.conf.sample /etc/quagga/demonio.conf`).

Tomar en cuenta que el término demonio, hace referencia a los demonios que ejecuta *quagga* en su sistema (*zebra.bgpd*, *ripd*, *ripngd*, *ospf6d*, *ospf6d*, *isisd*) y el *vttysh* es la interfaz virtual que corre el SO para configurar el *router*, sin necesidad de entrar a cada demonio en particular.

```
root@tesis-VirtualBox:/# cp /usr/share/doc/quagga-core/examples/vtysh.conf.sample /etc/quagga/vtysh.conf
root@tesis-VirtualBox:/# cp /usr/share/doc/quagga-core/examples/zebra.conf.sample /etc/quagga/zebra.conf
root@tesis-VirtualBox:/# cp /usr/share/doc/quagga-core/examples/bgpd.conf.sample /etc/quagga/bgpd.conf
```



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
root@tesis-VirtualBox:/# cp /usr/share/doc/quagga-core/examples/ospfd.conf.sample /etc/quagga/ospfd.conf
root@tesis-VirtualBox:/# cp /usr/share/doc/quagga-core/examples/ospf6d.conf.sample /etc/quagga/ospf6d.conf
root@tesis-VirtualBox:/# cp /usr/share/doc/quagga-core/examples/ripd.conf.sample /etc/quagga/ripd.conf
root@tesis-VirtualBox:/# cp /usr/share/doc/quagga-core/examples/ripngd.conf.sample /etc/quagga/ripngd.conf
root@tesis-VirtualBox:/# cp /usr/share/doc/quagga-core/examples/isisd.conf.sample /etc/quagga/isisd.conf
```

Figura 4. Copia de archivos ejemplo a archivos nuevos

**Paso 5:** Con la copia de los ficheros de todos los demonios ya efectuada, se asigna permisos y propiedad, de todos los archivos existentes en el paquete.

Se asigna todos los demonios al grupo *quagga* para la gestión en el sistema y *quaggavty* solo con *vttysh* con los comandos (***sudo chown quagga:quagga /etc/quagga/\*.conf***) y (***sudo chown quagga:quaggavty /etc/quagga/vtysh.conf***), respectivamente como en la Figura 5.

```
root@tesis-VirtualBox:/# sudo chown quagga:quagga /etc/quagga/*.conf
root@tesis-VirtualBox:/# sudo chown quagga:quaggavty /etc/quagga/vtysh.conf
```

Figura 5. Asignación de propiedad a los grupos *quagga* y *quaggavty*

**Paso 6:** La asignación del permiso **640** será de lectura y escritura a todos los archivos de configuración de todos los demonios del paquete, con el comando ***sudo chmod 640 /etc/quagga/\*.conf***

El parámetro (\*) se usa para indicar que el cambio se efectúa a todos los archivos que compongan ese directorio, tal como se aprecia en la Figura 6.

```
root@tesis-VirtualBox:/# sudo chmod 640 /etc/quagga/*.conf
```

Figura 6. Asignación de permiso de lectura y escritura a los archivos de configuración

**Paso 7:** Se procede a la escritura de registros, se crea un directorio para que ejecute los registros del sistema, con el comando ***mkdir /var/log/quagga/***, como en la Figura 7.

```
root@tesis-VirtualBox:/# mkdir /var/log/quagga/
```

Figura 7. Creación de fichero para ejecución de registros

**Paso 8:** Asignación al grupo *quagga* como propietario de la carpeta de los registros recientemente creados, dicha asignación se realizará con el comando ***chown quagga:quagga /var/log/quagga/***, como en la Figura 8.



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
root@tesis-VirtualBox:/# chown quagga:quagga /var/log/quagga/
```

Figura 8. Asignación de propiedad al grupo *Quagga* del fichero de registros

**Paso 9:** Crear un archivo que contenga los registros del demonio principal del *software* (*zebra*), con el comando ***touch /var/log/zebra.conf***, como en la Figura 9.

```
root@tesis-VirtualBox:/# touch /var/log/zebra.log
```

Figura 9. Creación de archivo de registro de *zebra*

**Paso 10:** Asignación a *quagga* como propietario del archivo recientemente agregado, con el comando ***chown quagga:quagga /var/log/zebra.log***, como en la Figura 10.

```
root@tesis-VirtualBox:/# chown quagga:quagga /var/log/zebra.log
```

Figura 10. Asignación de propiedad al grupo *Quagga* del archivo de registros de *zebra*

Para inicializar los demonios se debe arrancarlos y consultar su estado, con la siguiente estructura del comando:

***“sudo service demonio start”***

***“sudo service demonio status”***

Tomar en cuenta que el término demonio hace referencia a los protocolos de enrutamiento que ejecuta *Quagga* en su sistema (***zebra, bgpd, ripd, ripngd, ospf6d, ospf6d, isisd***)

La activación de los demonios se debe ejecutar tal como se aprecia en las Figuras de la 11 a la 17.

```
root@tesis-VirtualBox:/# sudo service zebra start
root@tesis-VirtualBox:/# sudo service zebra status
● zebra.service - GNU Zebra routing manager
   Loaded: loaded (/lib/systemd/system/zebra.service; enabled; vendor preset: e
   Active: active (running) since Wed 2020-11-11 00:43:29 -05; 6s ago
     Docs: man:zebra
   Process: 6853 ExecStart=/usr/sbin/zebra -d -A 127.0.0.1 -f /etc/quagga/zebra.
   Process: 6852 ExecStartPre=/bin/chown -f quagga:quaggavty /etc/quagga/vtysh.c
   Process: 6851 ExecStartPre=/bin/chown -f quagga:quagga /etc/quagga/zebra.conf
   Process: 6850 ExecStartPre=/bin/chmod -f 640 /etc/quagga/vtysh.conf /etc/quag
   Process: 6849 ExecStartPre=/sbin/ip route flush proto zebra (code=exited, sta
 Main PID: 6855 (zebra)
   Tasks: 1 (limit: 4667)
   CGroup: /system.slice/zebra.service
           └─6855 /usr/sbin/zebra -d -A 127.0.0.1 -f /etc/quagga/zebra.conf
```

Figura 11. Arranque y consulta de estado de demonio “*zebra*”





# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
root@tesis-VirtualBox:/# sudo service bgpd start
root@tesis-VirtualBox:/# sudo service bgpd status
● bgpd.service - BGP routing daemon
   Loaded: loaded (/lib/systemd/system/bgpd.service; enabled; vendor preset: en
   Active: active (running) since Wed 2020-11-11 00:44:11 -05; 15s ago
     Docs: man:bgpd
   Process: 6891 ExecStart=/usr/sbin/bgpd -d -A 127.0.0.1 -f /etc/quagga/bgpd.co
   Process: 6890 ExecStartPre=/bin/chown -f quagga:quagga /etc/quagga/bgpd.conf
   Process: 6889 ExecStartPre=/bin/chmod -f 640 /etc/quagga/bgpd.conf (code=exit
   Main PID: 6893 (bgpd)
     Tasks: 1 (limit: 4667)
    CGroup: /system.slice/bgpd.service
           └─6893 /usr/sbin/bgpd -d -A 127.0.0.1 -f /etc/quagga/bgpd.conf

nov 11 00:44:11 tesis-VirtualBox systemd[1]: Starting BGP routing daemon...
nov 11 00:44:11 tesis-VirtualBox systemd[1]: Started BGP routing daemon.
lines 1-14/14 (END)
```

Figura 12. Arranque y consulta de estado de demonio “bgpd”

```
root@tesis-VirtualBox:/# sudo service ospfd start
root@tesis-VirtualBox:/# sudo service ospfd status
● ospfd.service - OSPF routing daemon
   Loaded: loaded (/lib/systemd/system/ospfd.service; enabled; vendor preset: e
   Active: active (running) since Wed 2020-11-11 01:06:18 -05; 2s ago
     Docs: man:ospfd
   Process: 7388 ExecStart=/usr/sbin/ospfd -d -A 127.0.0.1 -f /etc/quagga/ospfd.
   Process: 7387 ExecStartPre=/bin/chown -f quagga:quagga /etc/quagga/ospfd.conf
   Process: 7386 ExecStartPre=/bin/chmod -f 640 /etc/quagga/ospfd.conf (code=exi
   Main PID: 7389 (ospfd)
     Tasks: 1 (limit: 4667)
    CGroup: /system.slice/ospfd.service
           └─7389 /usr/sbin/ospfd -d -A 127.0.0.1 -f /etc/quagga/ospfd.conf

nov 11 01:06:18 tesis-VirtualBox systemd[1]: Starting OSPF routing daemon...
nov 11 01:06:18 tesis-VirtualBox systemd[1]: Started OSPF routing daemon.
```

Figura 13. Arranque y consulta de estado de demonio “ospfd”

```
root@tesis-VirtualBox:/# sudo service ospf6d start
root@tesis-VirtualBox:/# sudo service ospf6d status
● ospf6d.service - OSPF routing daemon for IPv6
   Loaded: loaded (/lib/systemd/system/ospf6d.service; enabled; vendor preset:
   Active: active (running) since Wed 2020-11-11 01:06:39 -05; 3s ago
     Docs: man:ospf6d
   Process: 7426 ExecStart=/usr/sbin/ospf6d -d -A ::1 -f /etc/quagga/ospf6d.conf
   Process: 7425 ExecStartPre=/bin/chown -f quagga:quagga /etc/quagga/ospf6d.con
   Process: 7424 ExecStartPre=/bin/chmod -f 640 /etc/quagga/ospf6d.conf (code=ex
   Main PID: 7427 (ospf6d)
     Tasks: 1 (limit: 4667)
    CGroup: /system.slice/ospf6d.service
           └─7427 /usr/sbin/ospf6d -d -A ::1 -f /etc/quagga/ospf6d.conf

nov 11 01:06:39 tesis-VirtualBox systemd[1]: Starting OSPF routing daemon for I
nov 11 01:06:39 tesis-VirtualBox systemd[1]: Started OSPF routing daemon for IP
```

Figura 14. Arranque y consulta de estado de demonio “ospf6d”



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
root@tesis-VirtualBox:/# sudo service ripd start
root@tesis-VirtualBox:/# sudo service ripd status
● ripd.service - RIP routing daemon
   Loaded: loaded (/lib/systemd/system/ripd.service; enabled; vendor preset: en
   Active: active (running) since Wed 2020-11-11 01:07:02 -05; 4s ago
     Docs: man:ripd
   Process: 7464 ExecStart=/usr/sbin/ripd -d -A 127.0.0.1 -f /etc/quagga/ripd.co
   Process: 7463 ExecStartPre=/bin/chown -f quagga:quagga /etc/quagga/ripd.conf
   Process: 7462 ExecStartPre=/bin/chmod -f 640 /etc/quagga/ripd.conf (code=exit
   Main PID: 7465 (ripd)
     Tasks: 1 (limit: 4667)
    CGroup: /system.slice/ripd.service
           └─7465 /usr/sbin/ripd -d -A 127.0.0.1 -f /etc/quagga/ripd.conf

nov 11 01:07:02 tesis-VirtualBox systemd[1]: Starting RIP routing daemon...
nov 11 01:07:02 tesis-VirtualBox systemd[1]: Started RIP routing daemon.
```

Figura 15. Arranque y consulta de estado de demonio “ripd”

```
root@tesis-VirtualBox:/# sudo service ripngd start
root@tesis-VirtualBox:/# sudo service ripngd status
● ripngd.service - RIP routing daemon for IPv6
   Loaded: loaded (/lib/systemd/system/ripngd.service; enabled; vendor preset:
   Active: active (running) since Wed 2020-11-11 01:07:26 -05; 2s ago
     Docs: man:ripngd
   Process: 7508 ExecStart=/usr/sbin/ripngd -d -A ::1 -f /etc/quagga/ripngd.conf
   Process: 7507 ExecStartPre=/bin/chown -f quagga:quagga /etc/quagga/ripngd.con
   Process: 7506 ExecStartPre=/bin/chmod -f 640 /etc/quagga/ripngd.conf (code=ex
   Main PID: 7509 (ripngd)
     Tasks: 1 (limit: 4667)
    CGroup: /system.slice/ripngd.service
           └─7509 /usr/sbin/ripngd -d -A ::1 -f /etc/quagga/ripngd.conf

nov 11 01:07:26 tesis-VirtualBox systemd[1]: Starting RIP routing daemon for IP
nov 11 01:07:26 tesis-VirtualBox systemd[1]: Started RIP routing daemon for IPv
```

Figura 16. Arranque y consulta de estado de demonio “ripngd”

```
root@tesis-VirtualBox:/# sudo service isisd start
root@tesis-VirtualBox:/# sudo service isisd status
● isisd.service - IS-IS routing daemon
   Loaded: loaded (/lib/systemd/system/isisd.service; enabled; vendor preset: e
   Active: active (running) since Wed 2020-11-11 01:07:52 -05; 3s ago
     Docs: man:isisd
   Process: 7545 ExecStart=/usr/sbin/isisd -d -A 127.0.0.1 -f /etc/quagga/isisd.
   Process: 7544 ExecStartPre=/bin/chown -f quagga:quagga /etc/quagga/isisd.conf
   Process: 7543 ExecStartPre=/bin/chmod -f 640 /etc/quagga/isisd.conf (code=exi
   Main PID: 7546 (isisd)
     Tasks: 1 (limit: 4667)
    CGroup: /system.slice/isisd.service
           └─7546 /usr/sbin/isisd -d -A 127.0.0.1 -f /etc/quagga/isisd.conf
```

Figura 17. Arranque y consulta de estado de demonio “isisd”

**Paso 11:** Comprobar si la activación de los demonios fue la adecuada con el comando:

**“sudo systemctl is-enabled demonio.service”**



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

**“sudo systemctl enable demonio.service”**

Tomar en cuenta que el término demonio hace referencia a los protocolos de enrutamiento que ejecuta *Quagga* en su sistema (*zebra*, *bgpd*, *ripd*, *ripngd*, *ospfd*, *ospf6d*, *isisd*)

La verificación de la activación de los demonios se debe ejecutar tal como se aprecia en la Figura 18.

```
root@tesis-VirtualBox:/# sudo systemctl is-enabled zebra.service
enabled
root@tesis-VirtualBox:/# sudo systemctl is-enabled bgpd.service
enabled
root@tesis-VirtualBox:/# sudo systemctl is-enabled zebra.service
enabled
root@tesis-VirtualBox:/# sudo systemctl enable zebra.service
root@tesis-VirtualBox:/# sudo systemctl enable bgpd.service
```

```
root@tesis-VirtualBox:/# systemctl is-enabled ospfd
enabled
root@tesis-VirtualBox:/# systemctl is-enabled ospf6d
enabled
root@tesis-VirtualBox:/# systemctl is-enabled ripd
enabled
root@tesis-VirtualBox:/# systemctl is-enabled ripngd
enabled
root@tesis-VirtualBox:/# systemctl is-enabled isisd
enabled
root@tesis-VirtualBox:/# systemctl enable isisd
root@tesis-VirtualBox:/# systemctl enable ospfd
root@tesis-VirtualBox:/# systemctl enable ospf6d
root@tesis-VirtualBox:/# systemctl enable ripd
root@tesis-VirtualBox:/# systemctl enable ripngd
```

Figura 18. Verificación de activación de cada demonio del sistema

**Paso 12:** Para finalizar se realizará la apertura de sesiones de todos los demonios del sistema, con sus respectivas terminales y en la terminal virtual general **vttysh**, con los siguientes comandos:

**“sudo telnet localhost puerto TCP de cada demonio”**

**2601 interfaz principal del router (zebra)**

**2602 ripd**

**2603 ripngd**

**2604 isisd**

**2605 bgpd**

**2606 ospf6d**



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

2607 ospf6d

**vttysh terminal general del sistema**

Iniciar sesión con la contraseña de cada demonio que es la misma, en este caso **password: zebra**, tal como se aprecia en las Figura 19 a la 21.

```
root@tesis-VirtualBox:/# sudo vtysh
Hello, this is Quagga (version 1.2.4).
Copyright 1996-2005 Kunihiro Ishiguro, et al.
tesis-VirtualBox#
```

Figura 19. Inicio de sesión en la terminal virtual

```
root@tesis-VirtualBox:/# sudo telnet localhost 2605
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Hello, this is Quagga (version 1.2.4).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

User Access Verification

Password:
bgpd>
```

Figura 20. Inicio de sesión en el demonio *bgpd*

```
root@tesis-VirtualBox:/# sudo telnet localhost 2601
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Hello, this is Quagga (version 1.2.4).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

User Access Verification

Password:
Router>
```

Figura 21. Inicio de sesión en el demonio principal *zebra*



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

#### CONCLUSIONES

- 1) La administración de un sistema virtual deriva en consecuencias positivas respecto a la ocupación de espacios físicos y mantenimiento de este ya que en la mayoría de los casos las correcciones de errores se enfocan solo a nivel *software* y no hardware que generalmente resulta más caro y complejo.
- 2) Usar un sistema de licencia libre para la ejecución de un *software* de *routing*, en un entorno didáctico (generalmente en universidades), es beneficioso para el estudiante ya que tiene más retos que un emulador de redes, donde ciertos parámetros del sistema ya están por defecto.
- 3) La implementación física en el sistema operativo *RASPBIAN OS*, no tiene diferencia alguna con la instalación virtual, el único aspecto diferente es el nombre de las interfaces de comunicaciones, en *Ubuntu* se denominan *enp0s* y en *RASPBIAN* se denominan *eth*.

#### RECOMENDACIONES

- 1) Para mejorar un entorno de aprendizaje más profesional, se recomienda utilizar la ISO de la consola de cualquier distribución de *Linux*, en este caso se podrá realizar todo el trabajo en un entorno minimalista, libre de distracciones.



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

#### PRÁCTICA 2

1. **TEMA:** Práctica de Routing con rutas estáticas en *Quagga*, simulando una RED WAN.

#### 2. DESARROLLO DE LA PRÁCTICA

**NOTA:** El tiempo para realizar la práctica es de 60 minutos. Es necesario que el alumno tome capturas de todo el proceso realizado en la práctica para la posterior evidencia en el informe.

#### CONEXIÓN DE LOS 2 ROUTERS QUAGGA CON SUS ADAPTADORES DE RED

Nota: Encender los 2 *routers Quagga* (las 2 máquinas virtuales de *Ubuntu*) para realizar la configuración simultáneamente.

**Paso 1:** Previo a encender el *router Quagga* (máquina virtual), abrir el apartado de configuraciones de las 2 máquinas virtuales para configurar los adaptadores en red interna con el nombre RED1 o el que dé por defecto el programa tal como se aprecia en las Figuras 1 y 2, para simular una conexión virtual entre los 2 sistemas.

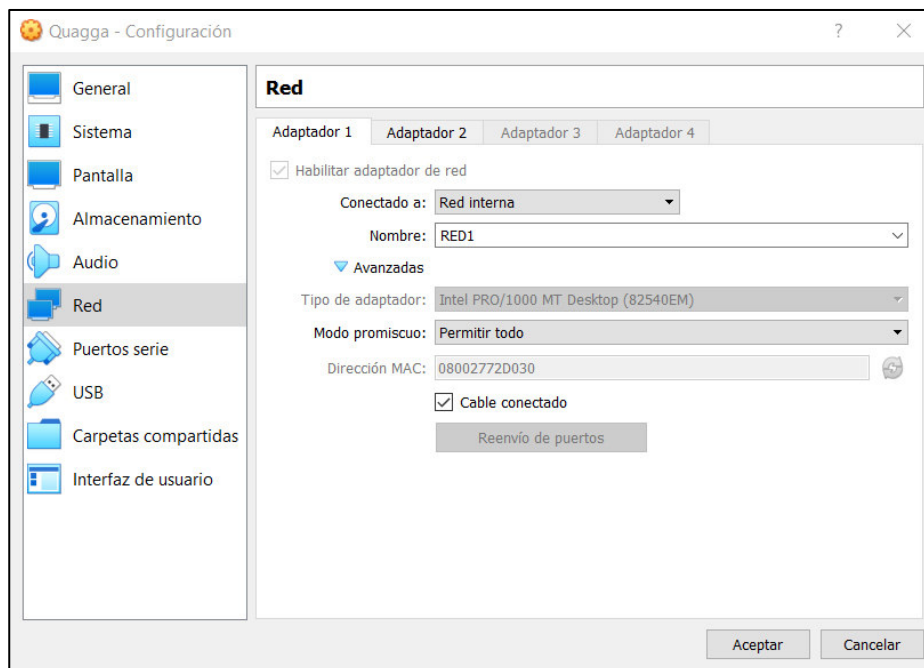


Figura 1. Configuración de adaptador 1 de red en Red interna



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

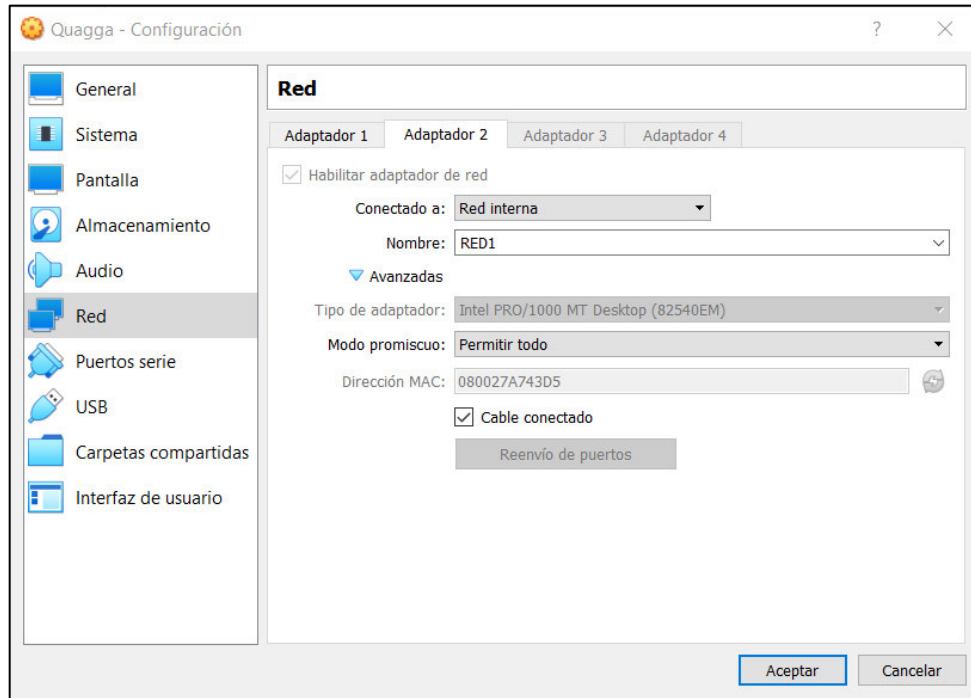


Figura 2. Configuración de adaptador 1 de red en Red interna

## RUTAS ESTÁTICAS CON IPv4

**Paso 1:** Conectar computadoras mediante una red virtual

Iniciar las máquinas virtuales que pertenecerán a nuestra de red con la distribución *Ubuntu*.

Con las maquinas ya encendidas procederemos a visualizar los controladores de red de esta, si aparece un error similar a la Figura 1, referente a que no está instalado el paquete disponible para mostrar las interfaces, descargar el paquete **net-tools**, con el comando **“sudo apt install net-tools”**, tal como se aprecia en la Figura 2.

```
router1@router1-VirtualBox:~$ ifconfig
Command 'ifconfig' not found, but can be installed with:
sudo apt install net-tools
```

Figura 1. Máquina sin paquete net-tools



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
router1@router1-VirtualBox:~$ sudo apt install net-tools
[sudo] password for router1:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  efibootmgr libegl1-mesa libfwup1 libllvm9 libwayland-egl1-mesa
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  net-tools
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 194 kB of archives.
After this operation, 803 kB of additional disk space will be used.
Get:1 http://ec.archive.ubuntu.com/ubuntu bionic/main amd64 net-tools amd64 1.6
0+git20161116.90da8a0-1ubuntu1 [194 kB]
Fetched 194 kB in 4s (54,6 kB/s)
Selecting previously unselected package net-tools.
(Reading database ... 163746 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20161116.90da8a0-1ubuntu1_amd64.deb .
..
Unpacking net-tools (1.60+git20161116.90da8a0-1ubuntu1) ...
Setting up net-tools (1.60+git20161116.90da8a0-1ubuntu1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
```

Figura 2. Comando de instalación de paquete “net-tools”

**Paso 2:** Una vez instalado el paquete **net-tools**, se procederá a ejecutarlo para verificar cuantas interfaces de red están disponibles en el sistema, con el comando **ifconfig**, tal como se aprecia en la Figura 3. Las interfaces de red activas tendrán que ser las necesarias de acuerdo con el esquema de red propuesto.

```
root@router1-VirtualBox:/home/router1# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.1.1.1 netmask 255.255.255.0 broadcast 10.1.1.255
    inet6 2001:db8:4::1 prefixlen 64 scopeid 0x0<global>
    inet6 fe80::a00:27ff:fe72:d030 prefixlen 64 scopeid 0x20<link>
    inet6 2001:db8:3::1 prefixlen 64 scopeid 0x0<global>
    ether 08:00:27:72:d0:30 txqueuelen 1000 (Ethernet)
    RX packets 634 bytes 61722 (61.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 484 bytes 44571 (44.5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.101.1 netmask 255.255.255.0 broadcast 192.168.101.255
    inet6 2001:db8:4::1 prefixlen 64 scopeid 0x0<global>
    inet6 2001:db8:1::1 prefixlen 64 scopeid 0x0<global>
    inet6 fe80::a00:27ff:fea7:43d5 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:a7:43:d5 txqueuelen 1000 (Ethernet)
    RX packets 411 bytes 42436 (42.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 147 bytes 16472 (16.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Figura 3. Interfaces disponibles en el sistema





# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

**Paso 3:** Una vez activadas las interfaces de red necesarias, se ingresará a cada *router* para su respectiva configuración, con el comando ***telnet localhost 2601***, como en la Figura 4, para acceder al sistema, la contraseña será “zebra”, la contraseña dependerá de cada “demonio” a utilizar, tomando en cuenta que cada uno de ellos representa un protocolo de enrutamiento diferente.

```
root@router1-VirtualBox:/home/router1# telnet localhost 2601
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Hello, this is Quagga (version 1.2.4).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

User Access Verification

Password:
Router> en
```

Figura 4. Acceso al *router*

**Paso 4:** Una vez ingresado al *router*, se procederá a correr el comando “***show running-config***”, para visualizar las configuraciones previas que ha tenido el *router*, en este caso al ser un sistema nuevo, las únicas acciones que mostrará el *router* serán las de acceso y las interfaces de red disponibles, como se aprecia en la Figura 5.

```
Router# show running-config

Current configuration:
!
hostname Router
password zebra
enable password zebra
!
interface enp0s3
!
interface enp0s8
!
interface enp0s9
!
interface lo
!
ip forwarding
ipv6 forwarding
!
!
line vty
!
end
```

Figura 5. Comando ***show running-config***

**Paso 5:** Para proceder en la configuración de las direcciones IP, se ingresará a la consola del *router* con el comando “***configure terminal***”, y se asignará una dirección IP



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

en las interfaces que estarán operativas, en este caso serán las interfaces **enp0s3** y **enp0s8**, a las cuales se les asignará las direcciones 10.0.0.1/24 (mascara 24) y 192.168.2.1/24, respectivamente con el comando **ip address** y guardando los cambios con el comando **do wr**, tal como se aprecia en la Figura 6 y la respectiva verificación en la Figura 7.

```
Router# configure terminal
Router(config)# inter
Router(config)# interface enp0s3
Router(config-if)# ip address 10.0.0.1/24
Router(config-if)# do wr
Configuration saved to /etc/quagga/zebra.conf
Router(config-if)# exit
Router(config)# interface enp0s8
Router(config-if)# ip address 192.168.2.1/24
Router(config-if)# do wr
Configuration saved to /etc/quagga/zebra.conf
Router(config-if)# exit
Router(config)# do wr
Configuration saved to /etc/quagga/zebra.conf
Router(config)# exit
```

Figura 6. Asignación de direcciones IPv4

```
Router# show run
Current configuration:
!
hostname Router
password zebra
enable password zebra
!
interface enp0s3
 ip address 10.0.0.1/24
!
interface enp0s8
 ip address 192.168.2.1/24
!
interface enp0s9
!
interface lo
!
ip forwarding
ipv6 forwarding
!
!
line vty
!
end
```

Figura 7. Verificación de asignación de direcciones en *Router 1*

**Paso 6:** Se repite el proceso para el segundo *router*, tomando en cuenta el “salto” en la asignación de las direcciones IP, como se puede visualizar en las Figuras 8 y 9.



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
Router> en
Password:
Router# configur
Router# configure ter
Router# configure terminal
Router(config)# inter
Router(config)# interface enp0s3
Router(config-if)# ip address 10.0.0.2/24
Router(config-if)# do wr
Configuration saved to /etc/quagga/zebra.conf
Router(config-if)# exit
Router(config)# interface enp0s8
Router(config-if)# ip address 192.168.1.1/24
Router(config-if)# do wr
Configuration saved to /etc/quagga/zebra.conf
Router(config-if)# exit
Router(config)# do wr
Configuration saved to /etc/quagga/zebra.conf
Router(config)# exit
```

Figura 8. Asignación de direcciones IPv4 en el *Router 2*

```
Router(config)# show run

Current configuration:
!
hostname Router
password zebra
enable password zebra
!
interface enp0s3
 ip address 10.0.0.2/24
!
interface enp0s8
 ip address 192.168.1.1/24
!
interface enp0s9
!
interface lo
!
ip forwarding
ipv6 forwarding
!
!
line vty
!
end
```

Figura 9. Verificación de asignación de direcciones en *Router 2*

**Paso 7:** Para la aplicación de rutas estáticas, se usa la misma sintaxis que en cualquier otro *router* de marca comercial, en este caso será con el comando “**ip route**” con la dirección de RED y su respectiva máscara de red, además con la dirección IP por la que tendrá que “aprender” el direccionamiento, la interfaz de salida, siguiendo el esquema de red, planteado en el inicio de la práctica, tal como se aprecia en la Figura 10 y verificación en la Figura 11.



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
Router(config)# conf t
% [ZEBRA] Unknown command: conf t
Router(config)# ip route 192.168.2.0/24 10.0.0.1
Router(config)# do wr
Configuration saved to /etc/quagga/zebra.conf
Router(config)# exit
Router# exit
```

Figura 10. Aplicación de rutas estáticas en *Router 1*

```
Router# sh run

Current configuration:
!
hostname Router
password zebra
enable password zebra
!
interface enp0s3
 ip address 10.0.0.2/24
!
interface enp0s8
 ip address 192.168.1.1/24
!
interface enp0s9
!
interface lo
!
ip route 192.168.1.0/24 10.0.0.1
!
ip forwarding
ipv6 forwarding
!
!
line vty
!
end
```

Figura 11. Verificación de aplicación de rutas estáticas en *Router 1*

**Paso 8:** Se repite el mismo proceso para el segundo *router*, tomando en consideración la dirección IP por donde saldrá el tráfico de ese extremo de la red, tal como se aprecia en la Figura 12 y verificación en la Figura 13.

```
Router> en
Password:
Router# conf t
Router(config)# ip route 192.168.1.0/24 10.0.0.1
Router(config)# do wr
Configuration saved to /etc/quagga/zebra.conf
Router(config)# exit
```

Figura 11. Aplicación de rutas estáticas en *Router 1*



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
Router(config)# sh run
Current configuration:
!
hostname Router
password zebra
enable password zebra
!
interface enp0s3
 ip address 10.0.0.2/24
!
interface enp0s8
 ip address 192.168.1.1/24
!
interface enp0s9
!
interface lo
!
ip route 192.168.2.0/24 10.0.0.1
!
ip forwarding
ipv6 forwarding
!
!
line vty
!
end
```

Figura 13. Verificación de aplicación de rutas estáticas en *Router 2*

**Paso 9:** Para verificar si la aplicación de rutas estáticas se realizó con éxito, se tendrá que mandar un mensaje de comprobación con el comando “*ping*” a la dirección IP de salida de tráfico, correspondiente, como en las Figuras 14 y 15 y la comunicación entre *routers* como en las Figuras 16 y 17.

```
root@router2-VirtualBox:/home/router2# ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
64 bytes from 192.168.2.1: icmp_seq=1 ttl=64 time=0.520 ms
64 bytes from 192.168.2.1: icmp_seq=2 ttl=64 time=1.07 ms
64 bytes from 192.168.2.1: icmp_seq=3 ttl=64 time=1.09 ms
64 bytes from 192.168.2.1: icmp_seq=4 ttl=64 time=1.02 ms
^C
--- 192.168.2.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3022ms
rtt min/avg/max/mdev = 0.520/0.930/1.098/0.240 ms
```

Figura 14. Comprobación de salida de tráfico de *router 2* al *router 1*



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
root@router1-VirtualBox:/home/router1# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0.623 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.294 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=1.12 ms
^C
--- 192.168.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2055ms
rtt min/avg/max/mdev = 0.294/0.680/1.124/0.341 ms
```

Figura 15. Comprobación de salida de tráfico de *router 1* al *router 2*

```
root@router1-VirtualBox:/home/router1# ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.717 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.606 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.923 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.527 ms
^C
--- 10.0.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3038ms
rtt min/avg/max/mdev = 0.527/0.693/0.923/0.149 ms
```

Figura 16. Comprobación de comunicación entre *routers*

```
root@router2-VirtualBox:/home/router2# ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=0.248 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.457 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.461 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.630 ms
^C
--- 10.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3075ms
rtt min/avg/max/mdev = 0.248/0.449/0.630/0.135 ms
```

Figura 17. Comprobación de comunicación entre *routers*

## CONCLUSIONES

- 1) El enrutamiento estático realizado en el *router Quagga* es muy similar al de un *router* de marca comercial, la sintaxis, manejo de interfaces de comunicación, configuración y consulta de la tabla de enrutamiento.
- 2) El enrutamiento estático reduce el tiempo de procesamiento del *router*, al reducir el tamaño de la tabla de enrutamiento y usar siempre la misma ruta para el envío de paquetes.



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

#### RECOMENDACIONES

- 1) Utilizar direcciones IPv4 en el enlace entre *routers* para aprovechar todos los recursos disponibles, en este caso sería con una máscara 30 (x.x.x.x/30).
- 2) Segmentar adecuadamente la red para evitar conflictos de direccionamiento y enrutamiento.



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

#### PRÁCTICA 3

**1. TEMA:** Práctica de Routing con el protocolo de enrutamiento dinámico RIP en *Quagga*, simulando una RED WAN.

#### 2. DESARROLLO DE LA PRÁCTICA

**NOTA:** El tiempo para realizar la práctica es de 60 minutos. Es necesario que el alumno tome capturas de todo el proceso realizado en la práctica para la posterior evidencia en el informe.

#### CONEXIÓN DE LOS 3 *ROUTERS QUAGGA* CON SUS ADAPTADORES DE RED

**NOTA:** Encender los 3 *routers Quagga* (las 3 máquinas virtuales de *Ubuntu*) para realizar la configuración simultáneamente.

#### Protocolo de enrutamiento RIP con direcciones IPv4

Las prácticas se engloban en un entorno virtual sin embargo se pueden ejemplificar con los módulos *Raspberry Pi* disponibles en el laboratorio.

**Paso 1:** Conectar computadoras mediante una red virtual. Iniciar las máquinas virtuales que pertenecerán a nuestra de red con la distribución Ubuntu.

Con las maquinas ya encendidas procederemos a visualizar los controladores de red de esta, si aparece un error similar a la Figura 1, referente a que no está instalado el paquete disponible para mostrar las interfaces, descargar el paquete “**net-tools**”, tal como se aprecia en la Figura 2.

```
router1@router1-VirtualBox:~$ ifconfig
Command 'ifconfig' not found, but can be installed with:
sudo apt install net-tools
```

Figura 1. Maquina sin paquete net-tools





# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
root@andres:/home/andres# sudo apt install net-tools
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
 net-tools
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 108 no actualizados.
Se necesita descargar 194 kB de archivos.
Se utilizarán 803 kB de espacio de disco adicional después de esta operación.
Des:1 http://ec.archive.ubuntu.com/ubuntu bionic/main amd64 net-tools amd64 1.60+git20161116.90da8a0-1ubuntu1 [194
 kB]
Descargados 194 kB en 1s (139 kB/s)
Seleccionando el paquete net-tools previamente no seleccionado.
(Leyendo la base de datos ... 129264 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar ../net-tools_1.60+git20161116.90da8a0-1ubuntu1_amd64.deb ...
Desempaquetando net-tools (1.60+git20161116.90da8a0-1ubuntu1) ...
Configurando net-tools (1.60+git20161116.90da8a0-1ubuntu1) ...
Procesando disparadores para man-db (2.8.3-2ubuntu0.1) ...
```

Figura 2. Comando de instalación de paquete “net-tools”

**Paso 2:** Una vez instalado el paquete net-tools, se procederá a ejecutarlo para verificar cuantas interfaces de red están disponibles en el sistema, con el comando “*ifconfig*”, tal como se aprecia en la Figura 3. Las interfaces de red activas tendrán que ser las necesarias de acuerdo con el esquema de red propuesto.

```
root@router1-VirtualBox:/home/router1# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.1.1.1 netmask 255.255.255.0 broadcast 10.1.1.255
    inet6 2001:db8:4::1 prefixlen 64 scopeid 0x0<global>
    inet6 fe80::a00:27ff:fe72:d030 prefixlen 64 scopeid 0x20<link>
    inet6 2001:db8:3::1 prefixlen 64 scopeid 0x0<global>
    ether 08:00:27:72:d0:30 txqueuelen 1000 (Ethernet)
    RX packets 634 bytes 61722 (61.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 484 bytes 44571 (44.5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.101.1 netmask 255.255.255.0 broadcast 192.168.101.255
    inet6 2001:db8:4::1 prefixlen 64 scopeid 0x0<global>
    inet6 2001:db8:1::1 prefixlen 64 scopeid 0x0<global>
    inet6 fe80::a00:27ff:fea7:43d5 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:a7:43:d5 txqueuelen 1000 (Ethernet)
    RX packets 411 bytes 42436 (42.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 147 bytes 16472 (16.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Figura 3. Interfaces disponibles en el sistema

**Paso 3:** Con privilegios de administrador (*root*), se entra al demonio principal del sistema con el comando “*telnet localhost 2601*”, este demonio representará la gestión central de *router*, lo iniciaremos con *enable*, como se observa en las Figuras 4 y 5, respectivamente.



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
root@angelo-VirtualBox:/home/angelo# telnet localhost 2601
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Hello, this is Quagga (version 1.2.4).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

User Access Verification

Password:
Router> zebra
```

Figura 4. Acceso al demonio central “zebra”

```
Router> en
Password:
Router#
```

Figura 5. Activación del *router*

**Paso 4:** Se configura de acuerdo con el diagrama especificado la primera dirección IP y la respectiva máscara de red, en este caso será máscara 24, que se asignará a la interfaz enp0s3 y la segunda en la interfaz enp0s8, que será la que dirija todo el tráfico que llegue de todo el extremo de la topología, tal como se aprecia en la Figura 6.

```
Router(config)# interface enp0s3
Router(config-if)# ip address 10.0.0.1/30
Router(config-if)# no sh
Router(config-if)# no shutdown
Router(config-if)# do wr
Configuration saved to /etc/quagga/zebra.conf
Router(config-if)#
```

```
Router(config)# interface enp0s8
Router(config-if)# ip address 192.168.1.1/24
Router(config-if)# no sh
Router(config-if)# no shutdown
Router(config-if)# do wr
Configuration saved to /etc/quagga/zebra.conf
```

```
Router(config)# int enp0s9
Router(config-if)# ip add
Router(config-if)# ip address 20.0.0.2/30
Router(config-if)# no sh
Router(config-if)# no shutdown
Router(config-if)# do wr
Configuration saved to /etc/quagga/zebra.conf
Router(config-if)# exit
Router(config)#
```

Figura 6. Asignación de las direcciones IP en las interfaces de comunicación del *Router 1*



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

**Paso 5:** Se realizará el mismo proceso para el segundo *router*, tomando en cuenta que esta dirección IP será el siguiente salto a la asignada en esa interfaz y la otra dirección ip perteneciente a otra red, para evitar conflictos de direccionamiento, como se aprecia en la Figura 7.

```
Router2# conf t
Router2(config)# inter
Router2(config)# interface enp0s3
Router2(config-if)# ip add
Router2(config-if)# ip address 10.0.0.2/24
Router2(config-if)# no sh
Router2(config-if)# exit
Router2(config)# interface enp0s8
Router2(config-if)# ip address 192.168.2.1/24
Router2(config-if)# no sh
Router2(config-if)# no shutdown
Router2(config-if)# do wr
Configuration saved to /etc/quagga/zebra.conf

Router2(config)# interface enp0s9
Router2(config-if)# ip add
Router2(config-if)# ip address 30.0.0.2/30
Router2(config-if)# no sh
Router2(config-if)# no shutdown
Router2(config-if)# do wr
Configuration saved to /etc/quagga/zebra.conf
Router2(config-if)#
```

Figura 7. Asignación de las direcciones IP en las interfaces de comunicación del *router*

2

```
Router3(config)# inter
Router3(config)# interface enp0s3
Router3(config-if)# ip add
Router3(config-if)# ip address 20.0.0.1/30
Router3(config-if)# no sh
Router3(config-if)# no shutdown
Router3(config-if)# do wr
Configuration saved to /etc/quagga/zebra.conf
Router3(config-if)# exit
Router3(config)# interface enp0s8
Router3(config-if)# ip address 30.0.0.1/30
Router3(config-if)# no sh
Router3(config-if)# no shutdown
Router3(config-if)# do wr
Configuration saved to /etc/quagga/zebra.conf
Router3(config-if)# exit
Router3(config)# interface enp0s9
Router3(config-if)# ip address 192.168.3.1/24
Router3(config-if)# no sh
Router3(config-if)# no shutdown
Router3(config-if)# do wr
Configuration saved to /etc/quagga/zebra.conf
Router3(config-if)#
```

Figura 8. Asignación de las direcciones IP en las interfaces de comunicación del *router*

3



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

**Paso 6:** Ahora se aplicará el protocolo de direccionamiento dinámico en este caso será *ripd*, logueándose en el demonio de RIP con el comando “**telnet localhost 2602**”, el número 2602 es el puerto TCP al que está asignado el demonio, como se visualiza en la Figura 9.

```
root@angelo-VirtualBox:/home/angelo# telnet localhost 2602
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Hello, this is Quagga (version 1.2.4).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

User Access Verification

Password:
ripd> zebra
% [RIP] Unknown command: zebra
ripd> en
ripd#
```

Figura 9. Acceso al demonio *ripd*

**Paso 7:** Se aplica el protocolo de enrutamiento dinámico rip, en ambas interfaces del *router 1*, en este caso, al usar este protocolo de enrutamiento dinámico, se debe usar **la dirección de red**, en las redes directamente conectadas al *router*, en este en las 2 interfaces de comunicación disponibles, sin olvidar de guardar las configuraciones realizadas con el comando “**do wr**”, como se aprecia en la Figura 10.

```
ripd(config)# router rip
ripd(config-router)# version 2
ripd(config-router)# network 10.0.0.0/30
ripd(config-router)# network 20.0.0.0/30
ripd(config-router)# do wr
Configuration saved to /etc/quagga/ripd.conf
ripd(config-router)#

ripd(config-if)# router rip
ripd(config-router)# version 2
ripd(config-router)# network 192.168.1.0/24
ripd(config-router)# do wr
Configuration saved to /etc/quagga/ripd.conf
ripd(config-router)#
```

Figura 10. Aplicación de protocolo de enrutamiento dinámico RIP en *Router 1*

**Paso 8:** Se realizará exactamente el mismo proceso con el segundo y tercer *router*, de igual manera tomando en cuenta todas las redes directamente conectadas, tal como se



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

ilustró en el diagrama propuesto al inicio de la práctica, como se observa en las Figuras 11 y 12, respectivamente.

```
ripd(config)# router rip
ripd(config-router)# version 2
ripd(config-router)# network 10.0.0.0/30
ripd(config-router)# network 30.0.0.0/30
ripd(config-router)# do wr
Configuration saved to /etc/quagga/ripd.conf
ripd(config-router)#
```

```
ripd(config-if)# router rip
ripd(config-router)# version 2
ripd(config-router)# network 192.168.2.0/24
ripd(config-router)# do wr
Configuration saved to /etc/quagga/ripd.conf
ripd(config-router)# exit
```

Figura 11. Aplicación de protocolo de enrutamiento dinámico RIP en *Router 2*

```
ripd(config)# router rip
ripd(config-router)# version 2
ripd(config-router)# network
ripd(config-router)# network 20.0.0.0/30
ripd(config-router)# network 30.0.0.0/30
ripd(config-router)# network 192.168.3.0/24
ripd(config-router)# do wr
Configuration saved to /etc/quagga/ripd.conf
ripd(config-router)# exit
ripd(config)#
```

Figura 12. Aplicación de protocolo de enrutamiento dinámico RIP en *Router 3*

**Paso 9:** Ahora comprobamos si da conectividad de extremo a extremo, para verificar la operatividad del protocolo.

Primero se prueba la comunicación del *router 1* con el *router 2*, con el comando “**ping**”, de igual manera que la Figura 13.

```
root@router1-VirtualBox:/home/router1# ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data:
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.406 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.458 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.493 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.518 ms
^C
--- 10.0.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3049ms
rtt min/avg/max/mdev = 0.406/0.468/0.518/0.049 ms
```



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
root@router2-VirtualBox:~# ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=0.366 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.501 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.310 ms
^C
--- 10.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2025ms
rtt min/avg/max/mdev = 0.310/0.392/0.501/0.081 ms
```

Figura 13. Prueba de conectividad entre *routers*

**Paso 10:** Ahora para comprobar la conectividad de extremo a extremo, se tendrá que realizar la misma prueba hacia las direcciones IP de salida de los *routers* para comprobar si el tráfico está llegando a cada una de las interfaces, como se visualiza en las Figuras 14, 15 y 16.

```
root@router1-VirtualBox:/home/router1# ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
64 bytes from 192.168.2.1: icmp_seq=1 ttl=64 time=0.367 ms
64 bytes from 192.168.2.1: icmp_seq=2 ttl=64 time=0.475 ms
64 bytes from 192.168.2.1: icmp_seq=3 ttl=64 time=0.475 ms
64 bytes from 192.168.2.1: icmp_seq=4 ttl=64 time=0.434 ms
^C
--- 192.168.2.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3060ms
rtt min/avg/max/mdev = 0.367/0.437/0.475/0.051 ms
```

Figura 14. Prueba de conectividad de extremo a extremo

```
root@router2-VirtualBox:~# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0.024 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.040 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=0.095 ms
^C
--- 192.168.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2025ms
rtt min/avg/max/mdev = 0.024/0.053/0.095/0.030 ms
```

Figura 15. Prueba de conectividad de extremo a extremo

```
root@router1-VirtualBox:~# ping 192.168.3.1
PING 192.168.3.1 (192.168.3.1) 56(84) bytes of data.
64 bytes from 192.168.3.1: icmp_seq=1 ttl=64 time=0.670 ms
64 bytes from 192.168.3.1: icmp_seq=2 ttl=64 time=0.982 ms
64 bytes from 192.168.3.1: icmp_seq=3 ttl=64 time=1.24 ms
64 bytes from 192.168.3.1: icmp_seq=4 ttl=64 time=0.939 ms
^C
--- 192.168.3.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 0.670/0.959/1.247/0.206 ms
```

Figura 16. Prueba de conectividad de extremo a extremo



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

**Paso 11:** Como se comprobó, el protocolo de enrutamiento se realizó correctamente y no hay pérdidas de datos en la red y lo podemos consultar en la tabla de enrutamiento de ambos *routers* con el comando “*show ip rip*”, como se aprecia en las Figuras 17,18 y 19.

```
ripd# show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
  (n) - normal, (s) - static, (d) - default, (r) - redistribute,
  (i) - interface

Network          Next Hop          Metric From      Tag Time
C(i) 10.0.0.0/30  0.0.0.0           1 self           0
C(i) 20.0.0.0/30  0.0.0.0           1 self           0
R(n) 30.0.0.0/30  10.1.1.1          2 10.1.1.1       0 03:00
C(i) 192.168.1.0/24 0.0.0.0           1 self           0
R(n) 192.168.2.0/24 10.1.1.1          2 10.1.1.1       0 03:00
R(n) 192.168.3.0/24 20.0.0.1          2 20.0.0.1       0 02:33
ripd#
```

Figura 17. Consulta de tabla de enrutamiento RIP en *Router 1*

```
ripd# show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
  (n) - normal, (s) - static, (d) - default, (r) - redistribute,
  (i) - interface

Network          Next Hop          Metric From      Tag Time
C(i) 10.0.0.0/30  0.0.0.0           1 self           0
R(n) 20.0.0.0/30  30.0.0.1          2 30.0.0.1       0 02:33
C(i) 30.0.0.0/30  0.0.0.0           1 self           0
R(n) 192.168.1.0/24 10.1.1.2          2 10.1.1.2       0 03:00
C(i) 192.168.2.0/24 0.0.0.0           1 self           0
R(n) 192.168.3.0/24 30.0.0.1          2 30.0.0.1       0 02:33
ripd#
```

Figura 18. Consulta de tabla de enrutamiento RIP en *Router 2*

```
ripd# show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
  (n) - normal, (s) - static, (d) - default, (r) - redistribute,
  (i) - interface

Network          Next Hop          Metric From      Tag Time
R(n) 10.0.0.0/30  30.0.0.2          2 30.0.0.2       0 02:54
C(i) 20.0.0.0/30  0.0.0.0           1 self           0
C(i) 30.0.0.0/30  0.0.0.0           1 self           0
R(n) 192.168.1.0/24 20.0.0.2          2 20.0.0.2       0 02:53
R(n) 192.168.2.0/24 30.0.0.2          2 30.0.0.2       0 02:54
C(i) 192.168.3.0/24 0.0.0.0           1 self           0
ripd#
```

Figura 19. Consulta de tabla de enrutamiento RIP en *Router 3*



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

#### CONCLUSIONES

- 1) El protocolo de enrutamiento vector distancia RIP, es ideal para este diseño de topología ya que el tiempo de respuesta de la tabla de enrutamiento de este es de aproximadamente 30 segundos, lo que deriva en un rendimiento excepcional en este tipo de redes pequeñas que puedes homologarse a una red LAN empresarial o una red WAN privatizada.

#### RECOMENDACIONES

- 1) Utilizar un diseño de red para referenciar el diseño planteado, ya que la plataforma de *routing* no tiene una interfaz gráfica intuitiva clara, como si lo tienen otros tales como *Cisco Packet Tracer* o *GNS3*, esto para diferenciar claramente las interfaces de comunicación a configurar.
- 2) Segmentar adecuadamente la red para evitar conflictos de direccionamiento y enrutamiento.





# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

#### PRÁCTICA 4

**1. TEMA:** Práctica de Routing con el protocolo de enrutamiento dinámico OSPF en *Quagga*, simulando una RED WAN.

#### 2. DESARROLLO DE LA PRÁCTICA

**NOTA:** El tiempo para realizar la práctica es de 60 minutos. Es necesario que el alumno tome capturas de todo el proceso realizado en la práctica para la posterior evidencia en el informe.

#### CONEXIÓN DE LOS 2 *ROUTERS QUAGGA* CON SUS ADAPTADORES DE RED

Nota: Encender los 2 *routers Quagga* (las 2 máquinas virtuales de *Ubuntu*) para realizar la configuración simultáneamente.

#### PROTOCOLO OSPF CON IPv4

**Paso 1:** Iniciar las máquinas virtuales que pertenecerán a nuestra de red con la distribución *Ubuntu*.

Con las maquinas ya encendidas procederemos a visualizar los controladores de red de esta, si aparece un error similar a la Figura 1, referente a que no está instalado el paquete disponible para mostrar las interfaces, descargar el paquete “**net-tools**”, tal como se aprecia en la Figura 2.

```
router1@router1-VirtualBox:~$ ifconfig
Command 'ifconfig' not found, but can be installed with:
sudo apt install net-tools
```

Figura 1. Maquina sin paquete net-tools



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
root@andres:/home/andres# sudo apt install net-tools
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
 net-tools
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 108 no actualizados.
Se necesita descargar 194 kB de archivos.
Se utilizarán 803 kB de espacio de disco adicional después de esta operación.
Des:1 http://ec.archive.ubuntu.com/ubuntu bionic/main amd64 net-tools amd64 1.60+git20161116.90da8a0-1ubuntu1 [194
kB]
Descargados 194 kB en 1s (139 kB/s)
Seleccionando el paquete net-tools previamente no seleccionado.
(Leyendo la base de datos ... 129264 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar ../net-tools_1.60+git20161116.90da8a0-1ubuntu1_amd64.deb ...
Desempaquetando net-tools (1.60+git20161116.90da8a0-1ubuntu1) ...
Configurando net-tools (1.60+git20161116.90da8a0-1ubuntu1) ...
Procesando disparadores para man-db (2.8.3-2ubuntu0.1) ...
```

Figura 2. Comando de instalación de paquete “net-tools”

Una vez instalado el paquete net-tools, se procederá a ejecutarlo para verificar cuantas interfaces de red están disponibles en el sistema, con el comando “*ifconfig*”, tal como se aprecia en la Figura 3. Las interfaces de red activas tendrán que ser las necesarias de acuerdo con el esquema de red propuesto.

```
root@router1-VirtualBox:/home/router1# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.1.1.1 netmask 255.255.255.0 broadcast 10.1.1.255
    inet6 2001:db8:4::1 prefixlen 64 scopeid 0x0<global>
    inet6 fe80::a00:27ff:fe72:d030 prefixlen 64 scopeid 0x20<link>
    inet6 2001:db8:3::1 prefixlen 64 scopeid 0x0<global>
    ether 08:00:27:72:d0:30 txqueuelen 1000 (Ethernet)
    RX packets 634 bytes 61722 (61.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 484 bytes 44571 (44.5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.101.1 netmask 255.255.255.0 broadcast 192.168.101.255
    inet6 2001:db8:4::1 prefixlen 64 scopeid 0x0<global>
    inet6 2001:db8:1::1 prefixlen 64 scopeid 0x0<global>
    inet6 fe80::a00:27ff:fea7:43d5 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:a7:43:d5 txqueuelen 1000 (Ethernet)
    RX packets 411 bytes 42436 (42.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 147 bytes 16472 (16.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Figura 3. Interfaces disponibles en el sistema

**Paso 2:** Con privilegios de administrador (*root*), se entra al demonio principal del sistema con el comando “*telnet localhost 2601*”, este demonio representará la gestión central de *router*, lo iniciaremos con *enable*, como se observa en las Figuras 4 y 5, respectivamente.



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
root@angelo-VirtualBox:/home/angelo# telnet localhost 2601
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Hello, this is Quagga (version 1.2.4).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

User Access Verification

Password:
Router> zebra
```

Figura 4. Acceso al demonio central “zebra”

```
Router> en
Password:
Router#
```

Figura 5. Activación del *router*

**Paso 3:** Se configura de acuerdo con el diagrama especificado la primera dirección ip y la respectiva mascara de red, en este caso será mascara 24, que se asignara a la interfaz enp0s3 y la segunda en la interfaz enp0s8, que será la que direcciona todo el tráfico que llegue de todo el extremo de la topología, tal como se aprecia en la Figura 6.

```
Router# conf t
Router(config)# inter
Router(config)# interface enp0s3
Router(config-if)# ip add
Router(config-if)# ip address 33.1.1.1/24
Router(config-if)# no sh
Router(config-if)# sh
show      shutdown
Router(config-if)# no sh
Router(config-if)# no shutdown
Router(config-if)# do we
% [ZEBRA] Unknown command: do we
Router(config-if)# do wr
Configuration saved to /etc/quagga/zebra.conf
Router(config-if)# exit
Router(config)# interface enp0s8
Router(config-if)# ip address 122.1.1.1/24
Router(config-if)# no sh
Router(config-if)# no shutdown
Router(config-if)# do wr
Configuration saved to /etc/quagga/zebra.conf
Router(config-if)#
```

Figura 6. Asignación de las direcciones IP en las interfaces de comunicación del *router*



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

**Paso 4:** Se repite el proceso, tal como se observa en la Figura 7, tomando en cuenta que esta dirección IP será el siguiente salto a la asignada en esa interfaz y otra dirección IP perteneciente a otra red, referenciando que es el *Router 2*.

```
Router2> en
Password:
Router2# configure terminal
Router2(config)# interface enp0s3
Router2(config-if)# ip address 33.1.1.2/24
Router2(config-if)# no shutdown
Router2(config-if)# do wr
Configuration saved to /etc/quagga/zebra.conf
Router2(config-if)# exit
Router2(config)# interface enp0s8
Router2(config-if)# ip address 133.1.1.1/24
Router2(config-if)# no shutdown
Router2(config-if)# do wr
Configuration saved to /etc/quagga/zebra.conf
```

Figura 7. Asignación de las direcciones IP en las interfaces de comunicación del *Router 2*

**Paso 5:** Ahora se aplicará el protocolo de direccionamiento dinámico en este caso será *OSPFd*, logueándose en el demonio de *ospf6d* con el comando “*telnet localhost 2604*”, el número 2604 es el puerto TCP al que está asignado el demonio.

Para la aplicación del protocolo de enrutamiento dinámico OSPF, hay que tomar en cuenta las siguientes consideraciones:

- Usar las direcciones de red
- Utilizar las redes directamente conectadas al router, basándose en el esquema de red propuesto.
- Especificar un área con un número determinado, área a la que pertenecerá dicho router.

Consideraciones que son aplicadas, tal como se observa en la Figura 8.



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
root@router1-VirtualBox:/home/router1# telnet localhost 2604
Trying 127.0.0.1...
Connected to localhost.
E:Rhythmbox character is '^]'.

Hello, this is Quagga (version 1.2.4).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

User Access Verification

Password:
ospfd> enable
ospfd# configure terminal
ospfd(config)# interface enp0s3
ospfd(config-if)# router ospf
ospfd(config-router)# ospf router-id 33.1.1.1
ospfd(config-router)# network 33.1.1.0/24 area 0
ospfd(config-router)# network 122.1.1.0/24 area 0
ospfd(config-router)# do wr
Configuration saved to /etc/quagga/ospfd.conf
ospfd(config-router)#
```

Figura 8. Aplicación de protocolo de enrutamiento dinámico OSPF en *Router 1*

**Paso 6:** Para el segundo *router* es la misma dinámica, tomando en cuenta de igual manera los mismos factores que para el *router 1*, el “identificador ospf” del *router* cambiará, referenciando que no es un campo obligatorio, en el primer *router* se utilizó la dirección IP que se le asignó a la interfaz, en este caso será de igual manera, la dirección **33.1.1.2** para el segundo *router* y la dirección **33.1.1.1** para el primer *router*, tal como está en la Figura 9.



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
root@router2-VirtualBox:~# telnet localhost 2604
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Hello, this is Quagga (version 1.2.4).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

User Access Verification

Password:
ospfd> enable
ospfd# configure terminal
ospfd(config)# router ospf
ospfd(config-router)# ospf router-id 33.1.1.2
ospfd(config-router)# network 33.1.1.0/24 area 0
ospfd(config-router)# network 133.1.1.0/24 area 0
ospfd(config-router)# do wr
Configuration saved to /etc/quagga/ospfd.conf
ospfd(config-router)#
```

Figura 9. Aplicación de protocolo de enrutamiento dinámico OSPF en *Router 2*

**Paso 7:** Ahora se comprueba si da conectividad de extremo a extremo, para verificar la operatividad del protocolo.

Primero se prueba la comunicación del *router 1* con el *router 2*, con el comando “**ping**”, de igual manera que la Figura 10.

```
root@router1-VirtualBox:~# ping 33.1.1.2
PING 33.1.1.2 (33.1.1.2) 56(84) bytes of data.
64 bytes from 33.1.1.2: icmp_seq=1 ttl=64 time=0.463 ms
64 bytes from 33.1.1.2: icmp_seq=2 ttl=64 time=0.688 ms
64 bytes from 33.1.1.2: icmp_seq=3 ttl=64 time=0.508 ms
64 bytes from 33.1.1.2: icmp_seq=4 ttl=64 time=0.664 ms
^C
--- 33.1.1.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3062ms
rtt min/avg/max/mdev = 0.463/0.580/0.688/0.101 ms
root@router1-VirtualBox:~#

root@router2-VirtualBox:~# ping 33.1.1.1
PING 33.1.1.1 (33.1.1.1) 56(84) bytes of data.
64 bytes from 33.1.1.1: icmp_seq=1 ttl=64 time=0.325 ms
64 bytes from 33.1.1.1: icmp_seq=2 ttl=64 time=1.04 ms
64 bytes from 33.1.1.1: icmp_seq=3 ttl=64 time=0.570 ms
64 bytes from 33.1.1.1: icmp_seq=4 ttl=64 time=0.345 ms
^C
--- 33.1.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3018ms
rtt min/avg/max/mdev = 0.325/0.570/1.041/0.288 ms
root@router2-VirtualBox:~#
```

Figura 10. Prueba de conectividad entre *routers*



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

**Paso 8:** Ahora para comprobar la conectividad de extremo a extremo, se tendrá que realizar la misma prueba hacia las direcciones IP de salida de los *routers* para comprobar si el tráfico está llegando a cada una de las interfaces, como se visualiza en la Figura 11.

```
root@router1-VirtualBox:~# ping 133.1.1.1
PING 133.1.1.1 (133.1.1.1) 56(84) bytes of data.
64 bytes from 133.1.1.1: icmp_seq=1 ttl=64 time=0.514 ms
64 bytes from 133.1.1.1: icmp_seq=2 ttl=64 time=0.519 ms
64 bytes from 133.1.1.1: icmp_seq=3 ttl=64 time=0.518 ms
64 bytes from 133.1.1.1: icmp_seq=4 ttl=64 time=0.451 ms
^C
--- 133.1.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3070ms
rtt min/avg/max/mdev = 0.451/0.500/0.519/0.036 ms
root@router1-VirtualBox:~#
```

```
root@router2-VirtualBox:~# ping 122.1.1.1
PING 122.1.1.1 (122.1.1.1) 56(84) bytes of data.
64 bytes from 122.1.1.1: icmp_seq=1 ttl=64 time=0.392 ms
64 bytes from 122.1.1.1: icmp_seq=2 ttl=64 time=0.420 ms
64 bytes from 122.1.1.1: icmp_seq=3 ttl=64 time=0.667 ms
64 bytes from 122.1.1.1: icmp_seq=4 ttl=64 time=0.341 ms
^C
--- 122.1.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3059ms
rtt min/avg/max/mdev = 0.341/0.455/0.667/0.125 ms
root@router2-VirtualBox:~#
```

Figura 11. Prueba de conectividad de extremo a extremo

**Paso 9:** Como se comprobó, el protocolo de enrutamiento se realizó correctamente y no hay pérdidas de datos en la red y lo podemos consultar en la tabla de enrutamiento del demonio con el comando **“show ip ospf neighbor”**, tal como se visualiza en las Figuras 12 y 13 y sobre todo la tabla de enrutamiento de ambos *routers*, con el comando **“show ip route”** como se observa en las Figuras 14 y 15, la cual indica todos los parámetros de la aplicación de un protocolo en el sistema.



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
ospfd> en
ospfd# show ip ospf neighbor

Neighbor ID      Pri State           Dead Time Address      Interface
  RXmtL RqstL DBsmL
33.1.1.2         1 Full/Backup      33.749s 33.1.1.2      enp0s3:33.1.1.1
  0         0         0
ospfd#
```

Figura 12. Consulta de tabla de enrutamiento ospf en *Router 1*

```
ospfd> enable
ospfd# show ip ospf neighbor

Neighbor ID      Pri State           Dead Time Address      Interface
  RXmtL RqstL DBsmL
33.1.1.1         1 Full/DR          35.526s 33.1.1.1      enp0s3:33.1.1.2
  0         0         0
ospfd#
```

Figura 13. Consulta de tabla de enrutamiento ospf en *Router 2*

```
Router# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, P - PIM, A - Babel, N - NHRP,
       > - selected route, * - FIB route

C>* 1.1.1.1/32 is directly connected, lo
B>* 2.2.2.2/32 [20/0] via 10.1.1.2, enp0s3, 06:45:12
C>* 10.0.0.0/24 is directly connected, enp0s3
C>* 10.1.1.0/24 is directly connected, enp0s3
O 33.1.1.0/24 [110/10] is directly connected, enp0s3, 00:49:13
C>* 33.1.1.0/24 is directly connected, enp0s3
O 122.1.1.0/24 [110/10] is directly connected, enp0s8, 00:48:55
C>* 122.1.1.0/24 is directly connected, enp0s8
C>* 127.0.0.0/8 is directly connected, lo
O 133.1.1.0/24 [110/20] via 33.1.1.2, enp0s3, 00:40:00
B>* 133.1.1.0/24 [20/0] via 10.1.1.2, enp0s3, 00:58:12
B>* 169.254.0.0/16 [20/0] via 10.1.1.2, enp0s3, 06:45:12
C>* 192.168.1.0/24 is directly connected, enp0s8
S 192.168.1.0/24 [1/0] via 10.0.0.2, enp0s3
R 192.168.2.0/24 [120/2] via 10.1.1.2, enp0s3, 05:30:31
B>* 192.168.2.0/24 [20/0] via 10.1.1.2, enp0s3, 06:29:02
C>* 192.168.101.0/24 is directly connected, enp0s8
```

Figura 14. Consulta de tabla de enrutamiento de *Router 1*





# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
Router2# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, P - PIM, A - Babel, N - NHRP,
       > - selected route, * - FIB route

B>* 1.1.1.1/32 [20/0] via 10.1.1.1, enp0s3, 06:51:24
C>* 2.2.2.2/32 is directly connected, lo
C>* 10.0.0.0/24 is directly connected, enp0s3
C>* 10.1.1.0/24 is directly connected, enp0s3
O 33.1.1.0/24 [110/10] is directly connected, enp0s3, 00:46:25
C>* 33.1.1.0/24 is directly connected, enp0s3
O 122.1.1.0/24 [110/20] via 33.1.1.1, enp0s3, 00:46:15
B>* 122.1.1.0/24 [20/0] via 10.1.1.1, enp0s3, 01:08:18
C>* 127.0.0.0/8 is directly connected, lo
O 133.1.1.0/24 [110/10] is directly connected, enp0s8, 00:46:13
C>* 133.1.1.0/24 is directly connected, enp0s8
K>* 169.254.0.0/16 is directly connected, enp0s3
R 192.168.1.0/24 [120/2] via 10.1.1.1, enp0s3, 05:40:38
C>* 192.168.1.0/24 is directly connected, enp0s8
C>* 192.168.2.0/24 is directly connected, enp0s8
B>* 192.168.101.0/24 [20/0] via 10.1.1.1, enp0s3, 06:51:24
Router2#
```

Figura 15. Consulta de tabla de enrutamiento de *Router 2*

## CONCLUSIONES

- 1) El protocolo de enrutamiento de estado de enlace OSPF, no es adecuado para diseños de red tan pequeños ya que esta patentado para redes WAN más robustas, el principal elemento diferenciador respecto a RIP es que este no maneja intervalos de tiempo para intercambiar las tablas de enrutamiento del *router*.
- 2) El concepto de área que entra en la configuración OSPF se basa en un grupo de *routers* que intercambian tablas de enrutamiento.

## RECOMENDACIONES

- 1) Utilizar un diseño de red para referenciar el diseño planteado, ya que la plataforma de *routing* no tiene una interfaz gráfica intuitiva clara, como si lo tienen otros tales como *Cisco Packet Tracer* o *GNS3*, esto para diferenciar claramente las interfaces de comunicación a configurar.
- 2) Segmentar adecuadamente la red para evitar conflictos de direccionamiento y enrutamiento.



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

#### PRÁCTICA 5

1. **TEMA:** Práctica de Routing con el protocolo de enrutamiento dinámico BGP en *Quagga*, simulando una RED WAN.

#### 2. DESARROLLO DE LA PRÁCTICA

**NOTA:** El tiempo para realizar la práctica es de 60 minutos. Es necesario que el alumno tome capturas de todo el proceso realizado en la práctica para la posterior evidencia en el informe.

#### CONEXIÓN DE LOS 2 *ROUTERS QUAGGA* CON SUS ADAPTADORES DE RED

**NOTA:** Encender los 2 *routers Quagga* (las 2 máquinas virtuales de *Ubuntu*) para realizar la configuración simultáneamente.

#### PROTOCOLO BGP CON IPv4

**Paso 1:** Con privilegios de administrador (*root*), se entra al demonio principal del sistema con el comando “*telnet localhost 2601*”, este demonio representará la gestión central de *router*, lo iniciaremos con **enable**, como se observa en las Figuras 1 y 2, respectivamente y se realizará en ambos *routers*.

```
root@angelo-VirtualBox:/home/angelo# telnet localhost 2601
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Hello, this is Quagga (version 1.2.4).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

User Access Verification

Password:
Router> zebra
```

Figura 1. Acceso al demonio central “zebra”

```
Router> en
Password:
Router#
```

Figura 2. Activación del *router*



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

**Paso 2:** Se configura de acuerdo con el diagrama especificado la primera dirección ip y la respectiva mascara de red, en este caso será mascara 24, que se asignara a la interfaz enp0s3 y la segunda en la interfaz enp0s8, que será la que direcciona todo el tráfico que llegue de todo el extremo de la topología, tal como se aprecia en la Figura 3.

```
Router# configure terminal
Router(config)# interface enp0s3
Router(config-if)# description torouter2
Router(config-if)# link-detect
Router(config-if)# ip address 10.1.1.2/24
Router(config-if)# ipv6 nd suppress-ra
Router(config-if)# exit
Router(config)# interface enp0s8
Router(config-if)# description clientesrouter1
Router(config-if)# link-detect
Router(config-if)# ip address 192.168.1.1/24
Router(config-if)# ipv6 nd suppress-ra
Router(config-if)# exit
Router(config)# interface lo
Router(config-if)# description loopback
Router(config-if)# ip address 2.2.2.2/32
Router(config-if)# exit
Router(config)# router-id 2.2.2.2
Router(config)# ip forwarding
Router(config)# line vty
Router(config-line)# exec-timeout 22222 0
Router(config-line)# exit
Router(config)# do wr
Configuration saved to /etc/quagga/zebra.conf
Router(config)#
```

Figura 3. Asignación de las direcciones IP en las interfaces de comunicación del *Router 1*

**Paso 3:** Se realizará el mismo proceso para el segundo *router*, tomando en cuenta que esta dirección ip será el siguiente salto a la asignada en esa interfaz y la otra dirección IP perteneciente a otra red, para evitar conflictos de direccionamiento, como en la Figura 4.



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
Router2# configure terminal
Router2(config)# interface enp0s3
Router2(config-if)# description torouter1
Router2(config-if)# link-detect
Router2(config-if)# ip address 10.1.1.1/24
Router2(config-if)# ipv6 nd suppress-ra
Router2(config-if)# exit
Router2(config)# interface enp0s8
Router2(config-if)# description clientesrouter2
Router2(config-if)# link-detect
Router2(config-if)# ip address 192.168.101.1/24
Router2(config-if)# ipv6 nd suppress-ra
Router2(config-if)# exit
Router2(config)# interface lo
Router2(config-if)# description loopback
Router2(config-if)# ip address 1.1.1.1/32
Router2(config-if)# exit
Router2(config)# router-id 1.1.1.1
Router2(config)# ip forwarding
Router2(config)# exit
Router2# do wr
% [ZEBRA] Unknown command: do wr
Router2# configure terminal
Router2(config)# do wr
Configuration saved to /etc/quagga/zebra.conf
Router2(config)# line vty
Router2(config-line)# exec-timeout 2222 0
Router2(config-line)# exit
Router2(config)# do wr
```

Figura 4. Asignación de las direcciones IP en las interfaces de comunicación del *Router 2*

**Paso 4:** Con la asignación de las direcciones IP en cada interfaz de comunicación se procede a configurar el protocolo de enrutamiento dinámico BGP, conectándose con la consola del respectivo demonio, en este caso será con el comando **“telnet localhost 2605 ó telnet localhost bgpd”** tal como se aprecia en la Figura 5, proceso que se realiza en ambos *routers*.



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
root@router1-VirtualBox:~# telnet localhost 2605
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Hello, this is Quagga (version 1.2.4).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

User Access Verification

Password:
bgpd> enable
bgpd# configure terminal
bgpd(config)#
```

Figura 5. Conexión con el demonio de enrutamiento *bgpd*

**Paso 5:** Ahora se aplica el protocolo de enrutamiento dinámico en ambos *routers*, tomando en cuenta varias consideraciones, como estas:

1. Borrar el AS por defecto que tiene el *router*, el cual es AS: 7675
2. Asignar un área de servicio a cada *router* que será nombrada con cualquier número, en el caso del primer *router* con un AS: 200 y para el segundo *router* con un AS: 100.
3. Utilizar el identificador del *router* que se asignó previamente.
4. Utilizar la dirección de red directamente conectada al *router*.

Todas estas consideraciones se pueden observar tanto para el *router* 1 y 2 en las Figuras 6 y 7, respectivamente.

```
bgpd# conf t
bgpd(config)# no router bgp 7675
```

```
bgpd# configure terminal
bgpd(config)# router bgp 200
bgpd(config-router)# bgp router-id 2.2.2.2
bgpd(config-router)# network 192.168.1.0/24
bgpd(config-router)# redistribute kernel
bgpd(config-router)# redistribute connected
bgpd(config-router)# neighbor 10.1.1.1 remote-as 100
bgpd(config-router)# exit
bgpd(config)# line vty
bgpd(config-line)# exec-timeout 32768 0
bgpd(config-line)# exit
bgpd(config)# do wr
Configuration saved to /etc/quagga/bgpd.conf
bgpd(config)#
```

Figura 6. Aplicación de enrutamiento BGP en *router* 1



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
bgpd# conf t
bgpd(config)# no router bgp 7675

bgpd# configure terminal
bgpd(config)# router bgp 100
bgpd(config-router)# bgp router-id 1.1.1.1
bgpd(config-router)# network 192.168.101.0/24
bgpd(config-router)# redistribute kernel
bgpd(config-router)# redistribute connected
bgpd(config-router)# neighbor 10.1.1.2 remote-as 200
bgpd(config-router)# exit
bgpd(config)# line vty
bgpd(config-line)# exec-timeout 32768 0
bgpd(config-line)# exit
bgpd(config)# do wr
Configuration saved to /etc/quagga/bgpd.conf
bgpd(config)#
```

Figura 7. Aplicación de enrutamiento BGP en *router 2*

**Paso 6:** Ahora se comprueba si los paquetes llegan de extremo a extremo con una prueba de *ping*, entre *routers* como en las Figuras 8 y 9 y de extremo a extremo como en las Figuras 10 y 11.

```
root@router1-VirtualBox:~# ping 10.1.1.1
PING 10.1.1.1 (10.1.1.1) 56(84) bytes of data.
64 bytes from 10.1.1.1: icmp_seq=1 ttl=64 time=0.580 ms
64 bytes from 10.1.1.1: icmp_seq=2 ttl=64 time=1.11 ms
64 bytes from 10.1.1.1: icmp_seq=3 ttl=64 time=0.966 ms
64 bytes from 10.1.1.1: icmp_seq=4 ttl=64 time=1.19 ms
^C
--- 10.1.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3032ms
rtt min/avg/max/mdev = 0.580/0.961/1.190/0.237 ms
```

Figura 8. Prueba de *ping* entre *routers*

```
root@router2-VirtualBox:~# ping 10.1.1.2
PING 10.1.1.2 (10.1.1.2) 56(84) bytes of data.
64 bytes from 10.1.1.2: icmp_seq=1 ttl=64 time=0.515 ms
64 bytes from 10.1.1.2: icmp_seq=2 ttl=64 time=1.15 ms
64 bytes from 10.1.1.2: icmp_seq=3 ttl=64 time=1.03 ms
64 bytes from 10.1.1.2: icmp_seq=4 ttl=64 time=1.02 ms
^C
--- 10.1.1.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3016ms
rtt min/avg/max/mdev = 0.515/0.931/1.159/0.248 ms
```

Figura 9. Prueba de *ping* entre *routers*



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
root@router1-VirtualBox:~# ping 192.168.101.1
PING 192.168.101.1 (192.168.101.1) 56(84) bytes of data.
64 bytes from 192.168.101.1: icmp_seq=1 ttl=64 time=0.620 ms
64 bytes from 192.168.101.1: icmp_seq=2 ttl=64 time=1.04 ms
64 bytes from 192.168.101.1: icmp_seq=3 ttl=64 time=1.03 ms
64 bytes from 192.168.101.1: icmp_seq=4 ttl=64 time=0.885 ms
^C
--- 192.168.101.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3024ms
rtt min/avg/max/mdev = 0.620/0.897/1.048/0.172 ms
```

Figura 10. Prueba de extremo-extremo de *router 1* a 2

```
root@router2-VirtualBox:~# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0.568 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=1.24 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=0.859 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=1.25 ms
^C
--- 192.168.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3033ms
rtt min/avg/max/mdev = 0.568/0.982/1.254/0.288 ms
```

Figura 11. Prueba de extremo-extremo de *router 2* a 1

**Paso 7:** Al final se comprueba si la tabla de *routing* del *router* se actualizó en ambos sistemas, tal como se aprecia en las Figuras 12 y 13, y la tabla BGP de igual manera como en las Figuras 14 y 15.

```
root@router1-VirtualBox:~# route -n
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
1.1.1.1          10.1.1.1       255.255.255.255 UGH    20    0      0 enp0s3
10.1.1.0         0.0.0.0        255.255.255.0  U     0     0      0 enp0s3
10.1.1.0         33.1.1.2       255.255.255.0  UG    20    0      0 enp0s3
20.0.0.0         0.0.0.0        255.255.255.252 U     0     0      0 enp0s3
33.1.1.0         0.0.0.0        255.255.255.0  U     0     0      0 enp0s3
122.1.1.0        0.0.0.0        255.255.255.0  U     0     0      0 enp0s8
133.1.1.0        10.1.1.1       255.255.255.0  UG    20    0      0 enp0s3
192.168.1.0      0.0.0.0        255.255.255.0  U     0     0      0 enp0s8
192.168.2.0      10.1.1.1       255.255.255.0  UG    20    0      0 enp0s3
192.168.101.0   10.1.1.1       255.255.255.0  UG    20    0      0 enp0s3
root@router1-VirtualBox:~#
```

Figura 12. Tabla de *routing* del *Router 1*



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```

root@router2-VirtualBox:~# route -n
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
2.2.2.2          10.1.1.2       255.255.255.255 UGH    20    0      0 enp0s3
10.1.1.0         0.0.0.0        255.255.255.0  U     0     0      0 enp0s3
20.0.0.0         0.0.0.0        255.255.255.252 U     0     0      0 enp0s3
33.1.1.0         0.0.0.0        255.255.255.0  U     0     0      0 enp0s3
122.1.1.0        10.1.1.2       255.255.255.0  UG    20    0      0 enp0s3
133.1.1.0        0.0.0.0        255.255.255.0  U     0     0      0 enp0s8
169.254.0.0     0.0.0.0        255.255.0.0    U     1000  0      0 enp0s3
192.168.1.0     10.1.1.2       255.255.255.0  UG    20    0      0 enp0s3
192.168.2.0     0.0.0.0        255.255.255.0  U     0     0      0 enp0s8
192.168.101.0   0.0.0.0        255.255.255.0  U     0     0      0 enp0s8
root@router2-VirtualBox:~#

```

Figura 13. Tabla de *routing* del *Router 2*

```

bgpd# show ip bgp
BGP table version is 0, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*> 1.1.1.1/32       10.1.1.1          0             0 100 ?
*> 2.2.2.2/32       0.0.0.0           0             32768 ?
* 10.1.1.0/24      10.1.1.1          0             0 100 ?
*>                  0.0.0.0           0             32768 ?
* 20.0.0.0/30      10.1.1.1          0             0 100 ?
*>                  0.0.0.0           0             32768 ?
* 33.1.1.0/24      10.1.1.1          0             0 100 ?
*>                  0.0.0.0           0             32768 ?
*> 122.1.1.0/24     0.0.0.0           0             32768 ?
*> 133.1.1.0/24     10.1.1.1          0             0 100 ?
* 192.168.1.0      0.0.0.0           0             32768 ?
*>                  0.0.0.0           0             32768 i
*> 192.168.2.0      10.1.1.1          0             0 100 ?
*> 192.168.101.0   10.1.1.1          0             0 100 i

Displayed 10 out of 14 total prefixes

```

Figura 14. Tabla BGP del *Router 1*





# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
bgpd> enable
bgpd# show ip bgp
BGP table version is 0, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*> 1.1.1.1/32       0.0.0.0           0             32768 ?
*> 2.2.2.2/32       10.1.1.2          0             0 200 ?
* 10.1.1.0/24      10.1.1.2          0             0 200 ?
*>                  0.0.0.0           0             32768 ?
* 20.0.0.0/30      10.1.1.2          0             0 200 ?
*>                  0.0.0.0           0             32768 ?
* 33.1.1.0/24      10.1.1.2          0             0 200 ?
*>                  0.0.0.0           0             32768 ?
*> 122.1.1.0/24     10.1.1.2          0             0 200 ?
*> 133.1.1.0/24     0.0.0.0           0             32768 ?
*> 192.168.1.0      10.1.1.2          0             0 200 i
*> 192.168.2.0      0.0.0.0           0             32768 ?
* 192.168.101.0    0.0.0.0           0             32768 ?
*>                  0.0.0.0           0             32768 i
```

Figura 15. Tabla BGP del *Router 2*

## CONCLUSIONES

- 1) El protocolo de enrutamiento vector distancia BGP, por la flexibilidad de funcionamiento es la opción por excelencia para comunicar todo el tráfico hacia Internet, así que es usado por la mayoría de ISP en el mundo.
- 2) El concepto de sistema autónomo (AS) que entra en la configuración BGP, se refiere a un conjunto de enrutadores que son comandados bajo la misma administración.
- 3) El direccionamiento de tráfico que se ejecuta dentro de un único AS se denomina como BGP interno o IBGP, cuando este direccionamiento se utiliza para conectar un AS a otros sistemas autónomos, se denomina BGP externo o EBGP.

## RECOMENDACIONES

- 1) Utilizar un diseño de red para referenciar el diseño planteado, ya que la plataforma de *routing* no tiene una interfaz gráfica intuitiva clara, como si lo tienen otros tales como *Cisco Packet Tracer* o *GNS3*, esto para diferenciar claramente las interfaces de comunicación a configurar.
- 2) Segmentar adecuadamente la red para evitar conflictos de direccionamiento y enrutamiento.



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

#### PRÁCTICA 6

1. **TEMA:** Práctica de Routing con el protocolo de enrutamiento dinámico RIPng en *Quagga*, simulando una RED WAN.

#### 2. DESARROLLO DE LA PRÁCTICA

**NOTA:** El tiempo para realizar la práctica es de 60 minutos. Es necesario que el alumno tome capturas de todo el proceso realizado en la práctica para la posterior evidencia en el informe.

#### CONEXIÓN DE LOS 2 *ROUTERS QUAGGA* CON SUS ADAPTADORES DE RED

Nota: Encender los 2 *routers Quagga* (las 2 máquinas virtuales de *Ubuntu*) para realizar la configuración simultáneamente.

#### PROTOCOLO DE ENRUTAMIENTO RIP CON DIRECCIONES IPV6

**Paso 1:** Con privilegios de administrador (*root*), se entra al demonio principal del sistema con el comando “*telnet localhost 2601*” o en este caso particular en la interfaz general (*vtys*), con el comando “*vtys*” este demonio representa la gestión central de *router* sin necesidad de iniciar sesión en *zebra*, como se observa en la Figura 1, para realizar toda la práctica o simplemente para asignar las direcciones IPv6.

```
root@router1-VirtualBox:~# vtysh
Hello, this is Quagga (version 1.2.4).
Copyright 1996-2005 Kunihiro Ishiguro, et al.
router1-VirtualBox# █
```

Figura 1. Acceso a la interfaz principal del sistema vtysh

**Paso 2:** Se configura de acuerdo con el diagrama especificado la primera dirección IP y la respectiva máscara de red, en este caso será máscara 64, que se asignará a la interfaz *enp0s3* y la segunda en la interfaz *enp0s8*, que será la que dirija todo el tráfico que llegue de todo el extremo de la topología, tal como se aprecia en la Figura 2.



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
router1-VirtualBox# configure terminal
router1-VirtualBox(config)# interface enp0s3
router1-VirtualBox(config-if)# ipv6 address 2001:db8:3::1/64
router1-VirtualBox(config-if)# ipv6 address 2001:db8:4::1/64
router1-VirtualBox(config-if)# do wr
Building Configuration...
Configuration saved to /etc/quagga/zebra.conf
Configuration saved to /etc/quagga/ripd.conf
Configuration saved to /etc/quagga/ripngd.conf
Configuration saved to /etc/quagga/ospfd.conf
Configuration saved to /etc/quagga/ospf6d.conf
Configuration saved to /etc/quagga/bgpd.conf
Configuration saved to /etc/quagga/isisd.conf
[OK]
router1-VirtualBox(config-if)# exit
router1-VirtualBox(config)# do wr
Building Configuration...
Configuration saved to /etc/quagga/zebra.conf
Configuration saved to /etc/quagga/ripd.conf
Configuration saved to /etc/quagga/ripngd.conf
Configuration saved to /etc/quagga/ospfd.conf
Configuration saved to /etc/quagga/ospf6d.conf
Configuration saved to /etc/quagga/bgpd.conf
Configuration saved to /etc/quagga/isisd.conf
[OK]
```

Figura 2. Asignación de las direcciones IP en las interfaces de comunicación del *Router 1*

**Paso 3:** Se verifica que estén asignadas las direcciones en las respectivas interfaces con el comando ***“show interface”*** tal como se aprecia en la Figura 3.



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
router1-VirtualBox# show interface
Interface enp0s3 is up, line protocol is up
Link ups:      1 last: Sun, 09 May 2021 18:51:13 -0500
Link downs:    0 last: (never)
vrf: 0
index 2 metric 0 mtu 1500
flags: <UP,BROADCAST,RUNNING,MULTICAST>
Type: Ethernet
HWaddr: 08:00:27:72:d0:30
inet 10.1.1.1/24 broadcast 10.1.1.255
inet6 2001:db8:3::1/64
inet6 fe80::a00:27ff:fe72:d030/64
inet6 2001:db8:4::1/64
Interface enp0s8 is up, line protocol is up
Link ups:      1 last: Sun, 09 May 2021 18:51:13 -0500
Link downs:    0 last: (never)
vrf: 0
index 3 metric 0 mtu 1500
flags: <UP,BROADCAST,RUNNING,MULTICAST>
Type: Ethernet
HWaddr: 08:00:27:a7:43:d5
inet 192.168.101.1/24 broadcast 192.168.101.255
inet6 2001:db8:1::1/64
inet6 fe80::a00:27ff:fea7:43d5/64
inet6 2001:db8:4::1/64
```

Figura 3. Interfaces de comunicación del *Router 1*

**Paso 4:** Se realizará el mismo proceso para el segundo *router*, tomando en cuenta que esta dirección IP será el siguiente salto a la asignada en esa interfaz y la otra dirección IP perteneciente a otra red, para evitar conflictos de direccionamiento, tal como se observa en la Figura 4.

```
router2-VirtualBox# configure terminal
router2-VirtualBox(config)# hostname Router2
Router2(config)# interface enp0s3
Router2(config-if)# ipv6 address 2001:db8:3::2/64
```

```
router2-VirtualBox(config)# interface enp0s8
router2-VirtualBox(config-if)# ipv6 address 2005:db8:5::1/64
router2-VirtualBox(config-if)# do wr
Building Configuration...
Configuration saved to /etc/quagga/zebra.conf
Configuration saved to /etc/quagga/ripd.conf
Configuration saved to /etc/quagga/ripngd.conf
Configuration saved to /etc/quagga/ospfd.conf
Configuration saved to /etc/quagga/ospf6d.conf
Configuration saved to /etc/quagga/bgpd.conf
Configuration saved to /etc/quagga/isisd.conf
[OK]
```

Figura 4. Asignación de las direcciones IP en las interfaces de comunicación del *Router 2*



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

**Paso 5:** Se verifica que estén asignadas las direcciones en las respectivas interfaces con el comando “*show interface*” tal como se aprecia en la Figura 5.

```
router2-VirtualBox# show interface
Interface enp0s3 is up, line protocol is up
  Link ups:      1 last: Mon, 10 May 2021 00:18:15 -0500
  Link downs:   0 last: (never)
  vrf: 0
  index 2 metric 0 mtu 1500
  flags: <UP,BROADCAST,RUNNING,MULTICAST>
  Type: Ethernet
  HWaddr: 08:00:27:82:8d:7d
  inet 10.1.1.2/24 broadcast 10.1.1.255
  inet6 2001:db8:3::2/64
  inet6 2001:db8:5::1/64
  inet6 fe80::a00:27ff:fe82:8d7d/64
Interface enp0s8 is up, line protocol is up
  Link ups:      1 last: Mon, 10 May 2021 00:18:15 -0500
  Link downs:   0 last: (never)
  vrf: 0
  index 3 metric 0 mtu 1500
  flags: <UP,BROADCAST,RUNNING,MULTICAST>
  Type: Ethernet
  HWaddr: 08:00:27:70:56:6d
  inet 192.168.1.1/24 broadcast 192.168.1.255
  inet6 2001:db8:2::1/64
  inet6 2001:db8:5::1/64
  inet6 fe80::a00:27ff:fe70:566d/64
```

Figura 5. Interfaces de comunicación del *Router 2*

**Paso 6:** Ahora se aplica el protocolo de direccionamiento dinámico en este caso será *ripngd*, logueándose con su respectivo demonio *ripngd*, con el comando “*telnet ::1 2603*”, el número 2603 es el puerto TCP al que está asignado el demonio como se visualiza en la Figura 6.

```
root@router1-VirtualBox:~# telnet ::1 2603
Trying ::1...
Connected to ::1.
Escape character is '^]'.

Hello, this is Quagga (version 1.2.4).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

User Access Verification

Password:
ripngd> enable
ripngd#
```

Figura 6. Acceso al demonio *ripngd*



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

**Paso 7:** Se aplica el protocolo de enrutamiento dinámico RIP, en ambas interfaces del *router* 1, en este caso, al usar este protocolo de enrutamiento dinámico, se debe usar **la dirección de red**, las redes directamente conectadas al *router*, en las 2 interfaces de comunicación disponibles, sin olvidar de guardar las configuraciones realizadas con el comando “**do wr**”, como se aprecia en la Figura 7.

```
router1-VirtualBox(config)# router ripng
router1-VirtualBox(config-router)# version 2
% Unknown command.
router1-VirtualBox(config-router)# network 2001:db8::3::/64
router1-VirtualBox(config-router)# network 2001:db8::4::/64
router1-VirtualBox(config-router)# do wr
Building Configuration...
Configuration saved to /etc/quagga/zebra.conf
Configuration saved to /etc/quagga/ripd.conf
Configuration saved to /etc/quagga/ripngd.conf
Configuration saved to /etc/quagga/ospfd.conf
Configuration saved to /etc/quagga/ospf6d.conf
Configuration saved to /etc/quagga/bgpd.conf
Configuration saved to /etc/quagga/isisd.conf
[OK]
```

```
router1-VirtualBox(config-router)# log file /var/log/quagga/ripd.log
router1-VirtualBox(config)# do wr
Building Configuration...
Configuration saved to /etc/quagga/zebra.conf
Configuration saved to /etc/quagga/ripd.conf
Configuration saved to /etc/quagga/ripngd.conf
Configuration saved to /etc/quagga/ospfd.conf
Configuration saved to /etc/quagga/ospf6d.conf
Configuration saved to /etc/quagga/bgpd.conf
Configuration saved to /etc/quagga/isisd.conf
[OK]
```

Figura 7. Aplicación de protocolo de enrutamiento dinámico RIPng en *Router* 1

**Paso 8:** Se realizará exactamente el mismo proceso con el segundo *router*, de igual manera tomando en cuenta todas las redes directamente conectadas, tal como se ilustró en el diagrama propuesto al inicio de la práctica y como se ilustra en la Figura 8.



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
router2-VirtualBox(config)# router ripng
router2-VirtualBox(config-router)# network 2001:db8:3::/64
There is same network configuration 2001:db8:3::/64
router2-VirtualBox(config-router)# network 2001:db8:5::/64
There is same network configuration 2001:db8:5::/64
router2-VirtualBox(config-router)# do wr
Building Configuration...
Configuration saved to /etc/quagga/zebra.conf
Configuration saved to /etc/quagga/ripd.conf
Configuration saved to /etc/quagga/ripngd.conf
Configuration saved to /etc/quagga/ospfd.conf
Configuration saved to /etc/quagga/ospf6d.conf
Configuration saved to /etc/quagga/bgpd.conf
Configuration saved to /etc/quagga/isisd.conf
[OK]
```

Figura 8. Aplicación de protocolo de enrutamiento dinámico RIPng en *Router 2*

**Paso 9:** Ahora se comprueba si existe conectividad de extremo a extremo, para verificar la operatividad del protocolo.

Primero se prueba la comunicación del *router 1* con el *router 2*, con el comando “*ping*” hacia las direcciones IP de salida de los *routers* para comprobar si el tráfico está llegando a cada una de las interfaces, como se visualiza en las Figuras 9 y 10.

```
root@router1-VirtualBox:~# ping 2001:db8:5::1
PING 2001:db8:5::1(2001:db8:5::1) 56 data bytes
64 bytes from 2001:db8:5::1: icmp_seq=1 ttl=64 time=0.408 ms
64 bytes from 2001:db8:5::1: icmp_seq=2 ttl=64 time=0.484 ms
64 bytes from 2001:db8:5::1: icmp_seq=3 ttl=64 time=0.483 ms
^C
--- 2001:db8:5::1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2031ms
rtt min/avg/max/mdev = 0.408/0.458/0.484/0.039 ms
```

Figura 9. Prueba de extremo-extremo *Router 1* a *Router 2*

```
root@router2-VirtualBox:/home/router2# ping 2001:db8:4::1
PING 2001:db8:4::1(2001:db8:4::1) 56 data bytes
64 bytes from 2001:db8:4::1: icmp_seq=1 ttl=64 time=0.565 ms
64 bytes from 2001:db8:4::1: icmp_seq=2 ttl=64 time=0.968 ms
64 bytes from 2001:db8:4::1: icmp_seq=3 ttl=64 time=0.602 ms
64 bytes from 2001:db8:4::1: icmp_seq=4 ttl=64 time=0.642 ms
^C
--- 2001:db8:4::1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3024ms
rtt min/avg/max/mdev = 0.565/0.694/0.968/0.161 ms
```

Figura 10. Prueba de extremo-extremo *Router 2* a *Router 1*



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

Como se comprobó, el protocolo de enrutamiento se realizó correctamente y no hay pérdidas de datos en la red y lo podemos consultar en la tabla de enrutamiento de ambos *routers* con el comando “*show ipv6 rip*”, como se aprecia en las Figuras 11 y 12.

```
Router# show ipv6 route
Codes: K - kernel route, C - connected, S - static, R - RIPng,
       O - OSPFv6, I - IS-IS, B - BGP, A - Babel, N - NHRP,
       > - selected route, * - FIB route

C>* ::1/128 is directly connected, lo
C>* 2001:db8:1::/64 is directly connected, enp0s8
B>* 2001:db8:2::/48 [20/0] via fe80::a00:27ff:fe82:8d7d, enp0s3, 00:57:20
C>* 2001:db8:3::/64 is directly connected, enp0s3
C>* 2001:db8:4::/64 is directly connected, enp0s8
R>* 2001:db8:5::/64 [120/2] via fe80::a00:27ff:fe82:8d7d, enp0s3, 00:03:48
C * fe80::/64 is directly connected, enp0s3
C>* fe80::/64 is directly connected, enp0s8
```

Figura 11. Consulta de tabla de enrutamiento IPv6 en *Router 1*

```
Router# sh ipv6 route
Codes: K - kernel route, C - connected, S - static, R - RIPng,
       O - OSPFv6, I - IS-IS, B - BGP, A - Babel, N - NHRP,
       > - selected route, * - FIB route

C>* ::1/128 is directly connected, lo
B>* 2001:db8:1::/48 [20/0] via fe80::a00:27ff:fe72:d030, enp0s3, 01:12:52
C>* 2001:db8:2::/64 is directly connected, enp0s8
C>* 2001:db8:3::/64 is directly connected, enp0s3
R>* 2001:db8:4::/64 [120/2] via fe80::a00:27ff:fe72:d030, enp0s8, 00:19:14
C>* 2001:db8:5::/64 is directly connected, enp0s8
C * fe80::/64 is directly connected, enp0s3
C>* fe80::/64 is directly connected, enp0s8
```

Figura 12. Consulta de tabla de enrutamiento IPv6 en *Router 2*

## CONCLUSIONES

- 1) El protocolo de enrutamiento vector distancia RIPng, es ideal para este diseño de topología ya que el tiempo de respuesta de la tabla de enrutamiento de este es de aproximadamente 30 segundos, lo que deriva en un rendimiento excepcional en este tipo de redes pequeñas que puedes homologarse a una red LAN empresarial o una red WAN privatizada.
- 2) La actualización a ng (nueva generación), implica que el protocolo soporta direcciones IPv6, sin ningún diferenciador en cuanto se refiere a configuración del *router* ya que se usa la misma sintaxis.





# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

#### RECOMENDACIONES

- 1) Utilizar un diseño de red para referenciar el diseño planteado, ya que la plataforma de *routing* no tiene una interfaz gráfica intuitiva clara, como si lo tienen otros tales como *Cisco Packet Tracer* o *GNS3*, esto para diferenciar claramente las interfaces de comunicación a configurar.
- 2) Segmentar adecuadamente la red para evitar conflictos de direccionamiento y enrutamiento.



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

#### PRÁCTICA 7

1. **TEMA:** Práctica de Routing con el protocolo de enrutamiento dinámico OSPF6d en *Quagga*, simulando una RED WAN.

#### 2. DESARROLLO DE LA PRÁCTICA

**NOTA:** El tiempo para realizar la práctica es de 60 minutos. Es necesario que el alumno tome capturas de todo el proceso realizado en la práctica para la posterior evidencia en el informe.

#### CONEXIÓN DE LOS 2 *ROUTERS QUAGGA* CON SUS ADAPTADORES DE RED

**NOTA:** Encender los 2 *routers Quagga* (las 2 máquinas virtuales de *Ubuntu*) para realizar la configuración simultáneamente.

#### PROTOCOLO OSPF CON DIRECCIONES IPv6

**Paso 1:** Con privilegios de administrador (*root*), se entra a la consola principal del sistema, la cual ejecutará todos los demonios y asignación de direcciones IP de forma simultánea sin necesidad de entrar a cada demonio de manera individual, se ejecuta con el comando “*vttysh*”, como se observa en las Figuras 1 y 2, y se realizará en ambos *routers*.

```
root@router1-VirtualBox:~# vtysh
Hello, this is Quagga (version 1.2.4).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

router1-VirtualBox# configure terminal
router1-VirtualBox(config)#
```

Figura 1. Acceso a la consola principal del *Router 1*

```
root@router2-VirtualBox:~# vtysh
Hello, this is Quagga (version 1.2.4).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

router2-VirtualBox# configure terminal
router2-VirtualBox(config)#
```

Figura 2. Acceso a la consola principal del *Router 2*

**Paso 2:** Se configura de acuerdo con el diagrama especificado la primera dirección ip y la respectiva mascara de red, en este caso será mascara 64, que se asignara a la



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

interfaz enp0s3 y la segunda en la interfaz enp0s8, que será la que direcciona todo el tráfico que llegue de todo el extremo de la topología, tal como se aprecia en la Figura 3.

```
router1-VirtualBox(config)# interface enp0s3
router1-VirtualBox(config-if)# ipv6 address 2001:10:10:1::1/64
router1-VirtualBox(config-if)# ipv6 ospf6 network broadcast
router1-VirtualBox(config-if)# ipv6 ospf6 instance-id 1
router1-VirtualBox(config-if)# no shutdown
router1-VirtualBox(config-if)# do wr
Building Configuration...
Configuration saved to /etc/quagga/zebra.conf
Configuration saved to /etc/quagga/ripd.conf
Configuration saved to /etc/quagga/ripngd.conf
Configuration saved to /etc/quagga/ospfd.conf
Configuration saved to /etc/quagga/ospf6d.conf
Configuration saved to /etc/quagga/bgpd.conf
Configuration saved to /etc/quagga/isisd.conf
[OK]
router1-VirtualBox(config-if)# exit
router1-VirtualBox(config)# interface enp0s8
router1-VirtualBox(config-if)# ipv6 address 2001:210:13:1::1/64
router1-VirtualBox(config-if)# ipv6 ospf6 network broadcast
router1-VirtualBox(config-if)# ipv6 ospf6 instance-id 3
router1-VirtualBox(config-if)# no shutdown
router1-VirtualBox(config-if)# do wr
Building Configuration...
Configuration saved to /etc/quagga/zebra.conf
Configuration saved to /etc/quagga/ripd.conf
Configuration saved to /etc/quagga/ripngd.conf
Configuration saved to /etc/quagga/ospfd.conf
Configuration saved to /etc/quagga/ospf6d.conf
```

```
Router(config)# ipv6 forwarding
Router(config)# ipv6 route 2001:210:13:1::/64 enp0s3
Router(config)# ipv6 route 2001:192:168:1::/64 enp0s3
Router(config)# do wr
Configuration saved to /etc/quagga/zebra.conf
```

Figura 3. Asignación de las direcciones IP en las interfaces de comunicación del *router*

1

**Paso 3:** Se realizará el mismo proceso para el segundo *router*, tomando en cuenta que esta dirección ip será el siguiente salto a la asignada en esa interfaz y la otra dirección IP perteneciente a otra red, para evitar conflictos de direccionamiento, como en la Figura 4.



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
router2-VirtualBox(config)# interface enp0s3
router2-VirtualBox(config-if)# ipv6 address 2001:10:10:1::2/64
router2-VirtualBox(config-if)# ipv6 ospf6 network broadcast
router2-VirtualBox(config-if)# ipv6 ospf6 instance-id 1
router2-VirtualBox(config-if)# no shutdown
router2-VirtualBox(config-if)# do wr
Building Configuration...
Configuration saved to /etc/quagga/zebra.conf
Configuration saved to /etc/quagga/ripd.conf
Configuration saved to /etc/quagga/ripngd.conf
Configuration saved to /etc/quagga/ospfd.conf
Configuration saved to /etc/quagga/ospf6d.conf
Configuration saved to /etc/quagga/bgpd.conf
Configuration saved to /etc/quagga/isisd.conf
[OK]
router2-VirtualBox(config-if)# exit
router2-VirtualBox(config)# interface enp0s8
router2-VirtualBox(config-if)# ip address 2001:192:168:1::1/64
% Unknown command.
router2-VirtualBox(config-if)# ipv6 address 2001:192:168:1::1/64
router2-VirtualBox(config-if)# ipv6 ospf6 network broadcast
router2-VirtualBox(config-if)# ipv6 ospf6 instance-id 3
router2-VirtualBox(config-if)# no shutdown
router2-VirtualBox(config-if)# do wr
Building Configuration...
Configuration saved to /etc/quagga/zebra.conf

Router2(config)# ipv6 forwarding
Router2(config)# ipv6 route 2001:210:13:1::/64
% Command incomplete.
Router2(config)# ipv6 route 2001:210:13:1::/64 enp0s3
Router2(config)# ipv6 route 2001:192:168:1::/64 enp0s3
Router2(config)# do wr
Configuration saved to /etc/quagga/zebra.conf
```

Figura 4. Asignación de las direcciones IP en las interfaces de comunicación del *Router 2*

**Paso 4:** Ahora se aplicará el protocolo de enrutamiento dinámico en este caso será OSPF6, *logueándose* en el demonio de **ospf6d** en la misma consola principal (**vtysh**) o conectándose al demonio principal, con el comando “**telnet ::1 2607**”, el número 2607 es el puerto TCP al que está asignado el demonio.

Para la aplicación del protocolo de enrutamiento dinámico OSPF, hay que tomar en cuenta las siguientes consideraciones:

- Usar las direcciones de red
- Utilizar las redes directamente conectadas al *router* basándose en el esquema de red
- Especificar un área con un número determinado
- Establecer un rango con las direcciones utilizadas previamente y dependiendo del caso pertenezcan a la misma área en este caso será 0.0.0.0



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

Consideraciones que se pueden observar en la Figura 5 para el *Router 1* y la Figura 6 para el *Router 2*.

```
router1-VirtualBox# configure terminal
router1-VirtualBox(config)# router ospf6
router1-VirtualBox(config-ospf6)# log-adjacency-changes
router1-VirtualBox(config-ospf6)# router-id 10
malformed OSPF Router-ID: 10
router1-VirtualBox(config-ospf6)# router-id 10.10.1.1
router1-VirtualBox(config-ospf6)# interface enp0s3 area 0.0.0.0
router1-VirtualBox(config-ospf6)# interface enp0s8 area 0.0.0.0
router1-VirtualBox(config-ospf6)# area 0.0.0.0 range 2001:10:10:1::/64
router1-VirtualBox(config-ospf6)# area 0.0.0.0 range 2001:210:13:1::/64
router1-VirtualBox(config-ospf6)# redistribute connected
router1-VirtualBox(config-ospf6)# redistribute kernel
router1-VirtualBox(config-ospf6)# redistribute static
router1-VirtualBox(config-ospf6)# do wr
Building Configuration...
Configuration saved to /etc/quagga/zebra.conf
Configuration saved to /etc/quagga/ripd.conf
Configuration saved to /etc/quagga/ripngd.conf
Configuration saved to /etc/quagga/ospfd.conf
Configuration saved to /etc/quagga/ospf6d.conf
Configuration saved to /etc/quagga/bgpd.conf
Configuration saved to /etc/quagga/isisd.conf
[OK]
```

Figura 5. Aplicación de protocolo de enrutamiento dinámico OSPF en versión 6

```
router2-VirtualBox(config)# router ospf6
router2-VirtualBox(config-ospf6)# log-adjacency-changes
router2-VirtualBox(config-ospf6)# router-id 10.10.1.2
router2-VirtualBox(config-ospf6)# interface enp0s3 area 0.0.0.0
router2-VirtualBox(config-ospf6)# interface enp0s8 area 0.0.0.0
router2-VirtualBox(config-ospf6)# area 0.0.0.0 range 2001:10:10:1::/64
router2-VirtualBox(config-ospf6)# area 0.0.0.0 range 2001:192.168.1::/64
Malformed argument: 2001:192.168.1::/64
router2-VirtualBox(config-ospf6)# area 0.0.0.0 range 2001:192.168:1::/64
Malformed argument: 2001:192.168:1::/64
router2-VirtualBox(config-ospf6)# area 0.0.0.0 range 2001:192:168:1::/64
router2-VirtualBox(config-ospf6)# redistribute connected
router2-VirtualBox(config-ospf6)# redistribute kernel
router2-VirtualBox(config-ospf6)# redistribute static
router2-VirtualBox(config-ospf6)# do wr
Building Configuration...
Configuration saved to /etc/quagga/zebra.conf
Configuration saved to /etc/quagga/ripd.conf
Configuration saved to /etc/quagga/ripngd.conf
Configuration saved to /etc/quagga/ospfd.conf
Configuration saved to /etc/quagga/ospf6d.conf
Configuration saved to /etc/quagga/bgpd.conf
Configuration saved to /etc/quagga/isisd.conf
[OK]
```

Figura 6. Aplicación de protocolo de enrutamiento dinámico OSPF en versión 6



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

**Paso 5:** Para comprobar la comunicación de ambos *routers* con el protocolo de enrutamiento, se efectúa una prueba simple de *ping* entre *routers* como en las Figuras 7 y 8 y de extremo a extremo como en las Figuras 9 y 10.

```
root@router1-VirtualBox:~# ping 2001:10:10:1::2
PING 2001:10:10:1::2(2001:10:10:1::2) 56 data bytes
64 bytes from 2001:10:10:1::2: icmp_seq=1 ttl=64 time=1.15 ms
64 bytes from 2001:10:10:1::2: icmp_seq=2 ttl=64 time=1.09 ms
64 bytes from 2001:10:10:1::2: icmp_seq=3 ttl=64 time=0.997 ms
64 bytes from 2001:10:10:1::2: icmp_seq=4 ttl=64 time=0.649 ms
^C
--- 2001:10:10:1::2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 0.649/0.973/1.155/0.195 ms
```

Figura 7. *Ping* entre *routers*

```
root@router2-VirtualBox:~# ping 2001:10:10:1::1
PING 2001:10:10:1::1(2001:10:10:1::1) 56 data bytes
64 bytes from 2001:10:10:1::1: icmp_seq=1 ttl=64 time=0.469 ms
64 bytes from 2001:10:10:1::1: icmp_seq=2 ttl=64 time=0.627 ms
64 bytes from 2001:10:10:1::1: icmp_seq=3 ttl=64 time=0.604 ms
64 bytes from 2001:10:10:1::1: icmp_seq=4 ttl=64 time=0.476 ms
^C
--- 2001:10:10:1::1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3070ms
rtt min/avg/max/mdev = 0.469/0.544/0.627/0.072 ms
```

Figura 8. *Ping* entre *routers*

```
root@router1-VirtualBox:~# ping 2001:192:168:1::1
PING 2001:192:168:1::1(2001:192:168:1::1) 56 data bytes
64 bytes from 2001:192:168:1::1: icmp_seq=1 ttl=64 time=1.03 ms
64 bytes from 2001:192:168:1::1: icmp_seq=2 ttl=64 time=0.624 ms
64 bytes from 2001:192:168:1::1: icmp_seq=3 ttl=64 time=1.03 ms
64 bytes from 2001:192:168:1::1: icmp_seq=4 ttl=64 time=1.14 ms
^C
--- 2001:192:168:1::1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3031ms
rtt min/avg/max/mdev = 0.624/0.960/1.149/0.200 ms
```

Figura 9. *Ping* de extremo a extremo de *Router 1* a *Router 2*



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
root@router2-VirtualBox:~# ping 2001:210:13:1::1
PING 2001:210:13:1::1(2001:210:13:1::1) 56 data bytes
64 bytes from 2001:210:13:1::1: icmp_seq=1 ttl=64 time=1.17 ms
64 bytes from 2001:210:13:1::1: icmp_seq=2 ttl=64 time=0.952 ms
64 bytes from 2001:210:13:1::1: icmp_seq=3 ttl=64 time=1.09 ms
64 bytes from 2001:210:13:1::1: icmp_seq=4 ttl=64 time=1.02 ms
64 bytes from 2001:210:13:1::1: icmp_seq=5 ttl=64 time=1.03 ms
^C
--- 2001:210:13:1::1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 0.952/1.056/1.176/0.080 ms
```

Figura 10. Ping de extremo a extremo de Router 2 a Router 1

**Paso 6:** Y para verificar que la tabla de enrutamiento se actualizó con las direcciones IP asignadas lo ejecutamos con el comando “*show ip route*” y “*show ip route ospf6*” tal como se ve en las Figuras 11 y 12, respectivamente.

```
Router# show ipv6 route
Codes: K - kernel route, C - connected, S - static, R - RIPng,
       O - OSPFv6, I - IS-IS, B - BGP, A - Babel, N - NHRP,
       > - selected route, * - FIB route

C>* ::1/128 is directly connected, lo
O  2001:10:10:1::/64 [110/10] is directly connected, enp0s3, 02:04:23
C>* 2001:10:10:1::/64 is directly connected, enp0s3
O  2001:210:13:1::/64 [110/10] is directly connected, enp0s8, 02:04:23
C>* 2001:210:13:1::/64 is directly connected, enp0s8
O  2001:db8:1::/64 [110/10] is directly connected, enp0s8, 02:04:23
C>* 2001:db8:1::/64 is directly connected, enp0s8
B>* 2001:db8:2::/48 [20/0] via fe80::a00:27ff:fe82:8d7d, enp0s3, 03:33:47
O  2001:db8:3::/64 [110/10] is directly connected, enp0s3, 02:04:23
C>* 2001:db8:3::/64 is directly connected, enp0s3
O  2001:db8:4::/64 [110/10] is directly connected, enp0s3, 02:04:23
C * 2001:db8:4::/64 is directly connected, enp0s3
C>* 2001:db8:4::/64 is directly connected, enp0s8
R>* 2001:db8:5::/64 [120/2] via fe80::a00:27ff:fe82:8d7d, enp0s8, 03:33:46
C * fe80::/64 is directly connected, enp0s3
C>* fe80::/64 is directly connected, enp0s8
```

Figura 11. Tabla de enrutamiento de direcciones IPv6

```
Router# show ipv6 route ospf6
Codes: K - kernel route, C - connected, S - static, R - RIPng,
       O - OSPFv6, I - IS-IS, B - BGP, A - Babel, N - NHRP,
       > - selected route, * - FIB route

O  2001:10:10:1::/64 [110/10] is directly connected, enp0s3, 02:08:08
O  2001:210:13:1::/64 [110/10] is directly connected, enp0s8, 02:08:08
O  2001:db8:1::/64 [110/10] is directly connected, enp0s8, 02:08:08
O  2001:db8:3::/64 [110/10] is directly connected, enp0s3, 02:08:08
O  2001:db8:4::/64 [110/10] is directly connected, enp0s3, 02:08:08
```

Figura 12. Tabla de enrutamiento del protocolo ospf6



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

#### CONCLUSIONES

- 1) El protocolo de enrutamiento de estado de enlace OSPF6d, compatible con direcciones IPv6, no es adecuado para diseños de red tan pequeños ya que está patentado para redes WAN más robustas, el principal elemento diferenciador respecto a RIP es que este no maneja intervalos de tiempo para intercambiar las tablas de enrutamiento del *router*.
- 2) En este caso el demonio de *Quagga* es "OSPF6d" pero el protocolo en general que soporta direcciones IPv6 es OSPFv3.
- 3) El concepto de área que entra en la configuración OSPF se basa en un grupo de *routers* que intercambian tablas de enrutamiento.

#### RECOMENDACIONES

- 1) Utilizar un diseño de red para referenciar el diseño planteado, ya que la plataforma de *routing* no tiene una interfaz gráfica intuitiva clara, como si lo tienen otros tales como *Cisco Packet Tracer* o *GNS3*, esto para diferenciar claramente las interfaces de comunicación a configurar.
- 2) Segmentar adecuadamente la red para evitar conflictos de direccionamiento y enrutamiento.





# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

#### PRÁCTICA 8

**1. TEMA:** Práctica de Routing con el protocolo de enrutamiento dinámico BGP IPv6 en *Quagga*, simulando una RED WAN.

#### 2. DESARROLLO DE LA PRÁCTICA

**NOTA:** El tiempo para realizar la práctica es de 60 minutos. Es necesario que el alumno tome capturas de todo el proceso realizado en la práctica para la posterior evidencia en el informe.

#### CONEXIÓN DE LOS 2 *ROUTERS QUAGGA* CON SUS ADAPTADORES DE RED

Nota: Encender los 2 *routers Quagga* (las 2 máquinas virtuales de *Ubuntu*) para realizar la configuración simultáneamente.

#### PROTOCOLO BGP CON IPv6

**Paso 1:** Con privilegios de administrador (*root*), se entra al demonio principal del sistema con el comando “*telnet localhost 2601*”, este demonio representará la gestión central de *router*, lo iniciaremos con **enable**, como se observa en las Figuras 1 y 2, respectivamente.

```
root@angelo-VirtualBox:/home/angelo# telnet localhost 2601
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Hello, this is Quagga (version 1.2.4).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

User Access Verification

Password:
Router> zebra
```

Figura 1. Acceso al demonio central “zebra”

```
Router> en
Password:
Router#
```

Figura 2. Activación del *router*

**Paso 2:** Se configura de acuerdo con el diagrama especificado la primera dirección ip en versión 6 y la respectiva mascara de red, en este caso será mascara 64, que se asignara a la interfaz *enp0s3* y la segunda en la interfaz *enp0s8*, que será la que



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

direccione todo el tráfico que llegue de todo el extremo de la topología, tal como se aprecia en la Figura 3 y la respectiva comprobación en la Figura 4.

Hay que tomar en consideración que para asignar direcciones IP en versión 6 el comando a ejecutar es “**ipv6 address AB00:C00D:E0F0::/64**”, tomando en cuenta que este es un ejemplo genérico, para asignar la dirección IPv6 correcta, se debe referenciar la topología de red planteada.

```
Router# configure ter
Router(config)# inter
Router(config)# interface enp0s3
Router(config-if)# ipv6 address 2001:db8:3::1/120
Router(config-if)# do wr
Configuration saved to /etc/quagga/zebra.conf
Router(config-if)# exit
Router(config)# interface enp0s3
Router(config-if)# exit
Router(config)# interface enp0s8
Router(config-if)# ipv6 address 4000:ac:4::1/64
Router(config-if)# do wr
Configuration saved to /etc/quagga/zebra.conf
```

Figura 3. Asignación de las direcciones IP en las interfaces de comunicación del *Router1*

```
Router# show interface
Interface enp0s3 is up, line protocol is up
  Link ups:      6 last: Wed, 29 Sep 2021 22:20:18 -0500
  Link downs:   5 last: Wed, 29 Sep 2021 22:20:16 -0500
  vrf: 0
  Description: torouter2
  index 2 metric 0 mtu 1500
  flags: <UP,BROADCAST,RUNNING,MULTICAST>
  Type: Ethernet
  HWaddr: 08:00:27:72:d0:30
  inet 10.0.0.1/30 broadcast 10.0.0.3
  inet 10.1.1.2/30 broadcast 10.1.1.3
  inet 33.1.1.1/24 broadcast 33.1.1.255
  inet6 fe80::a00:27ff:fe72:d030/64
  inet6 2001:ab::1/10
  inet6 2001:db8:3::1/120
```



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
Interface enp0s8 is up, line protocol is up
Link ups:      5 last: Wed, 29 Sep 2021 22:20:23 -0500
Link downs:    4 last: Wed, 29 Sep 2021 22:20:17 -0500
vrf: 0
Description: clientesrouter1
index 3 metric 0 mtu 1500
flags: <UP,BROADCAST,RUNNING,MULTICAST>
Type: Ethernet
HWaddr: 08:00:27:a7:43:d5
inet 122.1.1.1/24 broadcast 122.1.1.255
inet 192.168.1.1/24 broadcast 192.168.1.255
inet6 fe80::a00:27ff:fea7:43d5/64
inet6 2005:cd:3::1/64
inet6 4000:ac:4::1/64
```

Figura 4. Verificación de las direcciones IP asignadas en las interfaces de comunicación del *Router1*

**Paso 3:** Se realizará el mismo proceso para el segundo *router*, tomando en cuenta que esta dirección IPv6 será el siguiente salto a la asignada en esa interfaz y la otra dirección IPv6 perteneciente a la otra red propuesta en la topología inicial, consideraciones que se toman para evitar conflictos de direccionamiento tal como se aprecia en la Figura 5 y la respectiva comprobación en la Figura 6.

```
Router2# configure terminal
Router2(config)# interfac
Router2(config)# interface enp0s3
Router2(config-if)# ipv6 ad
Router2(config-if)# ipv6 address 2001:db8:3::2/120
Router2(config-if)# do wr
Configuration saved to /etc/quagga/zebra.conf
Router2(config-if)# exit
Router2(config)# interface enp0s8
Router2(config-if)# ipv6 address 4000:ac:5::1/64
Router2(config-if)# do wr
Configuration saved to /etc/quagga/zebra.conf
```

Figura 5. Asignación de las direcciones IP en las interfaces de comunicación del *Router 2*



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
Router2# show interface
Interface enp0s3 is up, line protocol is up
  Link ups:      5 last: Wed, 29 Sep 2021 22:14:36 -0500
  Link downs:    4 last: Wed, 29 Sep 2021 22:14:30 -0500
  vrf: 0
  Description: torouter1
  index 2 metric 0 mtu 1500
  flags: <UP,BROADCAST,RUNNING,MULTICAST>
  Type: Ethernet
  HWaddr: 08:00:27:82:8d:7d
  inet 10.0.0.2/30 broadcast 10.0.0.3
  inet 10.1.1.1/30 broadcast 10.1.1.3
  inet 33.1.1.2/24 broadcast 33.1.1.255
  inet6 fe80::a00:27ff:fe82:8d7d/64
  inet6 2001:ab::2/10
  inet6 2001:db8:3::2/120

Interface enp0s8 is up, line protocol is up
  Link ups:      5 last: Wed, 29 Sep 2021 22:14:36 -0500
  Link downs:    4 last: Wed, 29 Sep 2021 22:14:30 -0500
  vrf: 0
  Description: clientesrouter2
  index 3 metric 0 mtu 1500
  flags: <UP,BROADCAST,RUNNING,MULTICAST>
  Type: Ethernet
  HWaddr: 08:00:27:70:56:6d
  inet 133.1.1.1/24 broadcast 133.1.1.255
  inet 192.168.2.1/24 broadcast 192.168.2.255
  inet 192.168.101.1/24 broadcast 192.168.101.255
  inet6 fe80::a00:27ff:fe70:566d/64
  inet6 2005:cd:4::1/64
  inet6 4000:ac:5::1/64
```

Figura 6. Verificación de las direcciones IP asignadas en las interfaces de comunicación del *Router 2*

**Paso 4:** A continuación, se hace *ping* entre los dos *routers* únicamente para comprobar que haya conexión entre los mismos, como se observa en las Figuras 7 y 8.

```
root@router1-VirtualBox:~# ping 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=64 time=0.394 ms
64 bytes from 2001:db8:3::2: icmp_seq=2 ttl=64 time=0.555 ms
64 bytes from 2001:db8:3::2: icmp_seq=3 ttl=64 time=0.460 ms
64 bytes from 2001:db8:3::2: icmp_seq=4 ttl=64 time=0.504 ms
^C
--- 2001:db8:3::2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3067ms
rtt min/avg/max/mdev = 0.394/0.478/0.555/0.061 ms
```

Figura 7. Prueba de conexión del *Router1* hacia el *Router2*



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
root@router2-VirtualBox:/home/router2# ping 2001:db8:3::1
PING 2001:db8:3::1(2001:db8:3::1) 56 data bytes
64 bytes from 2001:db8:3::1: icmp_seq=1 ttl=64 time=0.622 ms
64 bytes from 2001:db8:3::1: icmp_seq=2 ttl=64 time=0.970 ms
64 bytes from 2001:db8:3::1: icmp_seq=3 ttl=64 time=1.00 ms
64 bytes from 2001:db8:3::1: icmp_seq=4 ttl=64 time=1.23 ms
^C
--- 2001:db8:3::1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3036ms
rtt min/avg/max/mdev = 0.622/0.956/1.231/0.218 ms
```

Figura 8. Prueba de conexión del *Router2* hacia el *Router1*

**Paso 5:** Para aplicar el protocolo de enrutamiento BGP con direcciones IPv6, en primero lugar se debe eliminar el área de servicio que tiene por defecto el *router* que en este caso es el 7675 con el comando “**no router bgp 7675**”, para luego poner el área de servicio a seleccionar para el nuevo ruteo tal como se observa en la Figura 9.

Tomar en cuenta que este proceso se lo realiza para **todos los routers** a configurar.

```
bgpd> en
bgpd# conf t
bgpd(config)# router bgp 200
BGP is already running; AS is 7675
bgpd(config)# no router bgp 7675
bgpd(config)# router bgp 200
bgpd(config-router)# exit
bgpd(config)# exit
bgpd# exit
```

Figura 9. Eliminación de área de servicio por defecto.

**Paso 6:** Ahora se aplicará el protocolo de enrutamiento dinámico en este caso será BGPd, *logueándose* en el demonio de **bgpd** con el comando “**telnet localhost 2605**”, para la aplicación del protocolo de enrutamiento dinámico BGP, hay que tomar en cuenta algunas consideraciones, entre las cuales, usar las direcciones de red, utilizar las redes directamente conectadas al *router*, basándose en el esquema de red y especificar un área de servicio (AS) con un número determinado tal como se observa en la Figura 10.

Otros comandos específicos para aplicación del protocolo son:

- **no auto-summary** que sirve para evitar el proceso de sumarización de redes semejantes.
- **no synchronization** que sirve para evitar que todas las redes se anuncien sino solo las que se deseen, proceso que ya se lo realiza con el comando “network”.



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

- Address family ipv6 que sirve para que las direcciones pertenezcan a un solo conjunto.

Consideraciones para enrutar, que se aprecian tanto para el *Router 1* y *2*, en las Figuras 10 y 11.

```
root@router1-VirtualBox:/home/router1# telnet localhost 2605
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Hello, this is Quagga (version 1.2.4).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

User Access Verification

Password:
bgpd> en
```

```
bgpd(config)# router bgp 100
bgpd(config-router)# no aut
bgpd(config-router)# no auto-summary
bgpd(config-router)# no sy
bgpd(config-router)# no synchronization
bgpd(config-router)# neighbor 2001:db8:3::2 remote-as 200
bgpd(config-router)# do wr
Configuration saved to /etc/quagga/bgpd.conf
bgpd(config-router)# exit
bgpd(config)# router bgp 100
bgpd(config-router)# add
bgpd(config-router)# address-family ipv
ipv4      ipv6
bgpd(config-router)# address-family ipv6
bgpd(config-router-af)# net
bgpd(config-router-af)# network 4000:ac:4::/48
bgpd(config-router-af)# neig
bgpd(config-router-af)# neighbor 2001:db8:3::2 ac
bgpd(config-router-af)# neighbor 2001:db8:3::2 activate
bgpd(config-router-af)# do wr
Configuration saved to /etc/quagga/bgpd.conf
```

Figura 10. Aplicación de protocolo de enrutamiento dinámico BGP en *Router 1*



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

```
bgpd(config)# router bgp 200
bgpd(config-router)# no au
bgpd(config-router)# no auto-summary
bgpd(config-router)# no syn
bgpd(config-router)# no synchronization
bgpd(config-router)# neig
bgpd(config-router)# neighbor 2001:db8:3::1 remote-as 100
bgpd(config-router)# add
bgpd(config-router)# address-family ipv6
bgpd(config-router-af)# network 2001:db8:3::1 ac
bgpd(config-router-af)# network 2001:db8:3::1 acti
bgpd(config-router-af)# network 4000:ac:5::/48
bgpd(config-router-af)# neig
bgpd(config-router-af)# neighbor 2001:db8:3::1 ac
bgpd(config-router-af)# neighbor 2001:db8:3::1 activate
bgpd(config-router-af)# do wr
Configuration saved to /etc/quagga/bgpd.conf
bgpd(config-router-af)# exit
bgpd(config-router)# do wr
Configuration saved to /etc/quagga/bgpd.conf
bgpd(config-router)# exi
```

Figura 11. Aplicación de protocolo de enrutamiento dinámico BGP en *Router 2*

**Paso 7:** Una vez realizado el enrutamiento BGP IPv6, se procede a realizar las pruebas de conexión respectivas entre las salidas de ambos *routers* como se visualiza en las Figuras 12 y 13.

```
root@router1-VirtualBox:/home/router1# ping 2001:db8:2::1
PING 2001:db8:2::1(2001:db8:2::1) 56 data bytes
64 bytes from 2001:db8:2::1: icmp_seq=1 ttl=64 time=0.556 ms
64 bytes from 2001:db8:2::1: icmp_seq=2 ttl=64 time=1.20 ms
64 bytes from 2001:db8:2::1: icmp_seq=3 ttl=64 time=1.07 ms
^C
--- 2001:db8:2::1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2010ms
rtt min/avg/max/mdev = 0.556/0.947/1.209/0.281 ms
```

Figura 12. Prueba de conexión de interfaz de salida de *Router1* a interfaz de salida de *Router2*

```
root@router2-VirtualBox:/home/router2# ping 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=0.518 ms
64 bytes from 2001:db8:1::1: icmp_seq=2 ttl=64 time=0.966 ms
64 bytes from 2001:db8:1::1: icmp_seq=3 ttl=64 time=0.942 ms
64 bytes from 2001:db8:1::1: icmp_seq=4 ttl=64 time=0.514 ms
^C
--- 2001:db8:1::1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3009ms
rtt min/avg/max/mdev = 0.514/0.735/0.966/0.219 ms
```

Figura 13. Prueba de conexión de interfaz de salida de *Router2* a interfaz de salida de *Router1*



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

**Paso 8:** Para finalizar se consulta la actualización en la tabla de enrutamiento IPv6 para verificar su actualización, verificación efectuada como se aprecia en las Figuras 14 y 15, para el *router 1* y *2* respectivamente.

```
bgpd# show ipv6 bgp
BGP table version is 0, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*> 4000:ac:4::/48    ::                0      32768 i
*> 4000:ac:5::/48    2001:db8:3::2    0      0 200 i

Displayed 2 out of 2 total prefixes
```

Figura 14. Tabla de enrutamiento BGP con direcciones IPv6 en *router 1*

```
bgpd> enable
bgpd# show ipv6 bgp
BGP table version is 0, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*> 4000:ac:4::/48    2001:db8:3::1    0      0 100 i
*> 4000:ac:5::/48    ::                0      32768 i

Displayed 2 out of 2 total prefixes
```

Figura 15. Tabla de enrutamiento BGP con direcciones IPv6 en *router 2*

## CONCLUSIONES

- 1) El protocolo de enrutamiento vector distancia BGP, por la flexibilidad de funcionamiento es la opción por excelencia para comunicar todo el tráfico hacia Internet, así que es usado por la mayoría de ISP en el mundo.
- 2) El concepto de sistema autónomo (AS) que entra en la configuración BGP, se refiere a un conjunto de enrutadores que son comandados bajo la misma administración.
- 3) El direccionamiento de tráfico que se ejecuta dentro de un único AS se denomina como BGP interno o IBGP, cuando este direccionamiento se utiliza para conectar un AS a otros sistemas autónomos, se denomina BPG externo o EBGP.





# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS

### HOJA GUÍA PARA EL INSTRUCTOR

#### RECOMENDACIONES

- 1) Utilizar un diseño de red para referenciar el diseño planteado, ya que la plataforma de *routing* no tiene una interfaz gráfica intuitiva clara, como si lo tienen otros tales como *Cisco Packet Tracer* o *GNS3*, esto para diferenciar claramente las interfaces de comunicación a configurar.
- 2) Segmentar adecuadamente la red para evitar conflictos de direccionamiento y enrutamiento.