

# **ESCUELA POLITÉCNICA NACIONAL**

## **ESCUELA DE FORMACIÓN DE TECNÓLOGOS**

### **IMPLEMENTACIÓN DE UN BALANCEADOR DE CARGA DE SALIDA A INTERNET BASADO EN UN SERVIDOR LINUX**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO  
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR  
EN REDES Y TELECOMUNICACIONES**

**CARLOS ANDRÉS FLORES ORTIZ**

**DIRECTOR: LEANDRO ANTONIO PAZMIÑO ORTIZ**

**DMQ, FEBRERO 2024**

## CERTIFICACIONES

Yo, Carlos Andrés Flores Ortiz declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



---

**Carlos Andrés Flores Ortiz**

**carlos.flores05@epn.edu.ec**

**carlos\_fo1089@hotmail.com**

Certifico que el presente trabajo de integración curricular fue desarrollado por Carlos Andrés Flores Ortiz, bajo mi supervisión.



---

**LEANDRO ANTONIO PAZMIÑO ORTIZ**

**DIRECTOR**

**leandro.pazmino@epn.edu.ec**

## DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.



---

**CARLOS ANDRÉS FLORES ORTIZ**

**carlos.flores05@epn.edu.ec**

**carlos\_fo1089@hotmail.com**

## **DEDICATORIA**

Primero a Dios por su beneplácito, a mis seres queridos, por su apoyo, tiempo y cuidados; a mi novia, por su paciencia, comprensión y ternura; a los docentes, cuyo trabajo arduo modela de manera constante el mañana y cuyas enseñanzas valen mucho más que el mero conocimiento; a la vida, por aquellos caminos, enseñanzas y sueños.

## **AGRADECIMIENTO**

A todos aquellos que contribuyen a diario para el desarrollo tecnológico, desde el obrero y la herramienta más simple, hasta el científico más docto y estructura útil más compleja; porque los sueños son universales y las creaciones más geniales pueden venir de las mentes más improbables, ergo la perseverancia, pasión y dedicación marcan huella en la eternidad, más allá de toda cuantificación. Gracias a todos los grandes antes de mí, mi granito de conocimiento.

# ÍNDICE DE CONTENIDOS

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA .....	II
DEDICATORIA .....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDOS .....	V
RESUMEN.....	VIII
ABSTRACT.....	IX
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1 Objetivo general.....	1
1.2 Objetivos específicos.....	1
1.3 Alcance .....	1
1.4 Marco Teórico .....	2
Balanceador de carga.....	2
Sistema operativo Linux.....	3
Lenguaje YAML .....	4
Netplan.....	4
Módulo iproute2 .....	5
Lenguaje Pearl.....	5
Hardening.....	5
Firewall .....	5
Sistema de detección de intrusos .....	6
SSH .....	6
2 METODOLOGÍA.....	7
3 RESULTADOS.....	8
3.1 Identificación de los requerimientos para la implementación del prototipo .....	8
Implementación de la plataforma.....	8
Sistema operativo Linux.....	9

Módulo de enrutamiento .....	9
Módulo de balanceo .....	9
Sistema de monitoreo de tráfico de red .....	9
Seguridad .....	10
3.2 Determinación del hardware y software requeridos .....	10
Selección del software .....	10
Selección de la plataforma de emulación del prototipo .....	10
Estado de actividad .....	10
Selección del sistema operativo .....	12
Selección de módulos de balanceo de carga .....	13
Selección de módulos de seguridad .....	13
Selección del hardware .....	14
3.3 Diseño del prototipo de balanceador de carga de salida a Internet .....	14
Preparativos del entorno de simulación .....	14
Diagrama de red .....	15
Diseño del modelo de balanceo de carga .....	16
Configuración de puertos de red .....	17
Acceso dividido .....	20
Balanceador de carga .....	23
App Armor .....	27
Remover software inútil y servicios .....	28
Protocolos y Servicios sin encriptación .....	29
IDS Suricata .....	30
Tiempo NTP .....	31
Firewall: .....	32
Servidor SSH .....	33
Manejo de Cuentas de usuario: .....	34
Desactivar acceso de root por SSH .....	35
Limitar IP's para SSH .....	36

Desactivar fingerprints de antiguos accesos SSH .....	36
Monitoreo de eventos local o mail: .....	37
3.4 Implementación del prototipo de balanceador de carga de salida a Internet. 38	
Instalación del servidor balanceador de carga .....	38
3.5 Realización exhaustiva de pruebas de funcionamiento.....	41
Pruebas de trabajo del balanceador de carga .....	41
Mediciones de efectividad del balanceador de carga .....	49
Proceso de las pruebas estadísticas del balanceador .....	51
Pruebas estadísticas del balanceador de carga .....	58
4 CONCLUSIONES .....	61
5 RECOMENDACIONES .....	63
6 REFERENCIAS BIBLIOGRÁFICAS .....	64
7 ANEXOS.....	71
ANEXO I: Certificado de Originalidad .....	i
ANEXO II: Enlaces .....	ii
ANEXO III: Códigos Fuente .....	iv
Configuración .....	iv
Build.PL .....	xvi



## RESUMEN

El propósito de este trabajo es realizar una implementación de un balanceador de carga de salida a Internet basado en un servidor Linux, el cual permite equilibrar el tráfico de una red de área local (LAN), cuando se presentan dos conexiones de Internet. Esto se realiza mediante la implementación de un servidor, simulado en el *software Emulated Virtual Environment Next Generation* (EVE-NG), el cual lleva Ubuntu Server 22.04 como sistema operativo para, sobre este, colocar los módulos necesarios para realizar el balanceo de carga del tráfico entrante y saliente de las dos conexiones a Internet [1]. El balanceador, cuenta también con una funcionalidad de “peso”, mediante la cual se asignan relaciones o porcentajes de tráfico deseado en cada línea y sistema de *failover*, para la detección y corrección de funcionamiento ante fallos de conectividad.

Se presenta en la primera sección, de manera lineal, el plan de objetivos a cumplir y el alcance del proyecto, también se presentan los conceptos básicos requeridos para la comprensión del tema, junto con las temáticas a conocer que se mencionan en el resto del documento.

Se plantea en la sección dos, la metodología o los pasos que se siguieron para la conformación del proyecto, se describen de manera lineal, clara y concisa los pasos seguidos para su consecución.

Se describe en la sección tres, como se plantearon los requerimientos técnicos y mediante estos se identificaron los componentes necesarios para el proyecto, a continuación, se describe como se determinó el software y hardware a ser utilizado, para luego, pasar a su diseño y posterior implementación, terminado esto se describen las medidas de seguridad propuestas para asegurar el servidor.

En la sección final, constan las conclusiones y recomendaciones obtenidas de la consecución de objetivos, implementación y pruebas del proyecto, así como también las fuentes bibliográficas referenciales del conocimiento aplicado y anexos relacionados.

**PALABRAS CLAVE:** balanceo, carga, Internet, EVE-NG, Linux.

## ABSTRACT

*The purpose of this work is to implement an outbound load balancer to the Internet based on a Linux server, which allows balancing the traffic of a local area network (LAN), when there are two Internet connections. This is done by implementing a server, simulated in the Emulated Virtual Environment Next Generation (EVE-NG) software, which has Ubuntu Server 22.04 as operating system to place on it the necessary modules to perform the load balancing of the incoming and outgoing traffic of the two Internet connections [1]. The load balancer also has a "weight" functionality, through which ratios or percentages of desired traffic are assigned to each line and a failover system, for the detection and correction of operation in case of connectivity failures.*

*The first section presents, in a linear fashion, the plan of objectives to be met and the scope of the project, as well as the basic concepts required to understand the subject, together with the topics to be covered that are mentioned in the rest of the document.*

*Section two establishes the methodology, or the steps followed for the conformation of the project, describing in a linear, clear, and concise manner the steps followed for its achievement.*

*Section three describes how the technical requirements were established and through them the necessary components for the project were identified. Next, it is described how the software and hardware to be used were determined, and then, the design and subsequent implementation are described.*

*The final section contains the conclusions and recommendations obtained from the achievement of the objectives, implementation and testing of the project, as well as the bibliographic sources of the applied knowledge and related annexes.*

**KEY WORDS:** *load balancing, load, Internet, EVE-NG, Linux.*

# 1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

El propósito de este proyecto se enfoca en la implementación de un balanceador de carga de salida a Internet utilizando un servidor Linux. El balanceador de carga optimizará la distribución del tráfico de salida de la red, mejorando así el rendimiento y la disponibilidad de la conexión a Internet.

## 1.1 Objetivo general

Implementar un balanceador de carga de salida a Internet basado en un servidor Linux.

## 1.2 Objetivos específicos

- Identificar los requisitos técnicos y funcionales necesarios para la implementación del balanceador de carga.
- Seleccionar la distribución de Linux y las herramientas de software adecuadas para el balanceador.
- Configurar el servidor Linux como un balanceador de carga de salida a Internet.
- Implementar medidas de seguridad para proteger el servidor Linux y el balanceador contra amenazas.
- Realizar pruebas exhaustivas para evaluar el rendimiento y la eficiencia del balanceador.

## 1.3 Alcance

El alcance del proyecto comprende la implementación de un balanceador de carga de salida a Internet basado en un servidor Linux, que satisfaga las siguientes funcionalidades:

- Distribuir el tráfico de salida a Internet entre dos conexiones de salida.
- Detectar automáticamente la disponibilidad de las conexiones de Internet y gestionarlas en consecuencia.
- Garantizar la disponibilidad y redundancia de la conexión a Internet.
- Establecer medidas de seguridad básicas para proteger el servidor Linux y el balanceador contra posibles amenazas.

## 1.4 Marco Teórico

### Balanceador de carga

El balanceo de carga se refiere al empleo de una técnica de programación usada para distribuir de manera adecuada los recursos de tráfico en una red, a saber, de manera específica se refiere a la manera en que se maneja la “carga”, mediante aliviar el exceso de tráfico presente en un nodo al dirigirlo hacia otros menos usados [2], [3].

Para comprender el término de balanceo de carga hay que tomar en cuenta tres conceptos; el primero, el compartimiento de carga, el cual consiste en determinar de manera arbitraria si un recurso está ocupado; el segundo, balanceo de carga, el cual se obtiene al tratar de que la carga de trabajo este distribuida en un pequeño grado de equilibrio en todos los recursos; la tercera, la nivelación de carga, se trata de evitar la congestión de recursos a toda costa, mediante la búsqueda un punto intermedio entre las dos técnicas anteriores [2], [3].

Se pueden clasificar a los balanceadores de carga dependiendo de su aplicación, plataforma y métodos; para empezar, por su aplicación, de entre los principales se tiene:

El balanceador de carga de red o balanceador de capa 4 (L4), se basa en el uso de direcciones de protocolo de Internet (IP) y puertos para ejercer su decisión, por tanto, este tipo de balanceo se ejecuta en la capa cuatro del modelo de interconexión de sistemas abiertos (OSI), es decir a nivel de Protocolo de Control de Tráfico (TCP) y Protocolo de Datagramas de Usuario (UDP), este tipo de balanceadores solo se ocupan en aplicaciones de tráfico general [3], [4], [5].

El balanceador de aplicación, trabaja en nivel 7 del modelo OSI, este distribuye las peticiones de acuerdo con parámetros más amplios, como por ejemplo cabeceras http, sesiones SSL, etc. Este tipo de balanceador permite un control más granular del tráfico [3], [4], [5].

El balanceador de carga de servidor global (GSLB), trabaja tanto a nivel de capa 4 como a nivel de capa 7, este se implementa en entornos donde el tráfico sea extenso y localizado en múltiples sitios, mediante este tipo de balanceador se provee un mejor servicio en centros de datos en la nube, pues al existir errores en otros nodos, estos toman las funciones de los averiados [3], [4], [5].

De acuerdo con la plataforma usada, los balanceadores de carga son:

Balanceador de carga por *hardware*, consiste en un balanceador de carga que se implementa de manera física, por esto el equipo usado es costoso y menos flexible que otras opciones [5].

Balanceador de carga por *software*, este se basa en el uso de programación para su implementación, el programa usado puede ser licenciado o de código abierto, dados sus parámetros de funcionamiento, estos son flexibles, fáciles de implementar y de bajo costo [5].

Balanceador de carga virtual, este trabaja al simular o emular un balanceador de carga por *hardware* mediante el uso de contenedores o máquinas virtuales [5].

Los métodos más comunes de balanceo de carga son:

**Round Robin**, consiste en la repartición de carga con un sistema rotativo para distribución equitativa de datos, se usa al seguir un orden de cola, que se repite de manera eventual [3], [4], [5].

**Weighted Round Robin**, se basa en el uso del algoritmo de *round robin*, pero con asignación de “peso”, parámetro que especifica la cantidad de tráfico relacional en la distribución [3], [4], [5].

**Least Connections**, se basa en el uso de un algoritmo para identificar las conexiones de menor uso para asignar el tráfico de manera equitativa [3], [4], [5].

**Least Response Time**, al igual que en el algoritmo de *least connections*, este identifica las conexiones de menor uso, pero además asigna prioridad a las que tienen menor tiempo de respuesta [3], [4], [5].

**Hash**, este algoritmo basa su funcionamiento en la asignación de direcciones, puertos o url's a un servidor fijo para optimizar el tráfico, así mediante el tipo de hash usado se tiene *IP hash* o *url hash* [3], [4], [5].

### **Sistema operativo Linux**

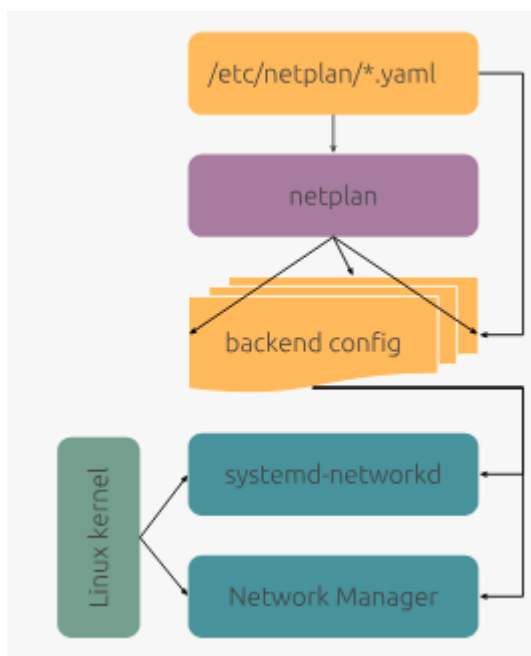
Un sistema operativo (SO), se define como una plataforma que contiene un conjunto de herramientas, orientadas a su vez a ofrecer un entorno de trabajo, medio de productividad o entretenimiento. Linux es un SO, lanzado en 1991, desarrollado en la Universidad de Helsinki, Finlandia, creado por Linus Torvalds, nace desde el punto de vista de software libre, con la finalidad de que, de este modelo se pueda aprender, distribuir y modificar el *software*. Este consta de un núcleo, denominado “kernel”, el cual es susceptible de modificaciones y optimización de parte de una comunidad muy activa,

de esta manera el kernel de Linux ha logrado convertirse en la base de miles de versiones de este, las cuales se denominan “distros” o distribuciones y se personalizan de acuerdo con la aplicación o preferencia del usuario [6].

## Lenguaje YAML

Su denominación viene del acrónimo de la palabra: YAML no es un lenguaje de marcado (YAML); es un lenguaje creado con el objetivo de serializar datos, enfoca su uso en archivos de configuración y programación; fue diseñado para ser sencillo de comprender y usar, su sintaxis posee rasgos de otros lenguajes de programación como perl, C, XML, JSON y HTML y su sangría usa el estilo Python (sin símbolos de formato). YAML no admite el carácter de tabulación y en su lugar, el espaciado se realiza con la tecla espacio, consta de una jerarquía basada en los niveles de sangría, las listas se realizan con el carácter “-” seguido de un espaciado y se pueden asignar valores denominados escalares, que son datos arbitrarios codificados en Unicode para utilizarse en forma de cadenas, números, fechas, o valores booleanos [7].

## Netplan



**Figura 1.1** Diagrama de bloque del módulo netplan [8]

Se describe como un “renderizador de abstracción de configuración de red”, es decir, se trata de una utilidad de Linux destinada a la configuración de redes, funciona mediante el uso de un archivo en lenguaje YAML, en el que se detalla las configuraciones deseadas, como se aprecia en la Figura 1.1, el documento de configuración genera una a su vez una instrucción de configuración de *backend*, la cual, se encarga de modificar el kernel de Linux mediante los renderizadores `systemd-`

networkd o networkmanager, al escoger uno en específico o a través de una combinación de estos [8].

### **Módulo iproute2**

Iproute2 es un módulo de Linux que contiene una colección de herramientas para el control de tráfico de tipo TCP/IP, de las cuales, de las más importantes son: la herramienta “ip”, que sirve para configurar parámetros relacionados con el protocolo de Internet 4 (ipv4), tanto como el protocolo de Internet versión 6 (ipv6); también la herramienta “tc” para control de tráfico. Entre sus capacidades se encuentran la posibilidad de añadir Calidad de Servicio (QoS) al tráfico de red, enrutamiento, entre otras; cabe añadir que de entre sus herramientas más conocidas destacan “ifconfig” y “route”, ambas discontinuadas y ligadas a versiones antiguas, se desaconseja su uso al conocerse escenarios de incompatibilidad en redes actuales y debido a poseer un repertorio limitado de características en comparación a sus nuevas iteraciones [9].

### **Lenguaje Pearl**

Desarrollado en la década de los 90 por Larry Wall, es un lenguaje de programación que recoge varios paradigmas, es dinámico y funciona en varias plataformas. Su enfoque inicial fue basado en la administración de sistemas y automatización de sus tareas. Debido a su facilidad de uso, se ha difundido como lenguaje de *scripting* y desarrollo web, posee una comunidad entusiasta, encargada de mantener una amplia biblioteca y cuenta con herramientas como la Perl Data Base Interface (Perl DBI) para programación, junto con módulos de optimización del código [10].

### ***Hardening***

El termino *hardening*, se refiere a la reducción de la superficie de ataque posible hacia un sistema informático, se ejecuta al seguir ciertas prácticas de seguridad como quitar paquetes de *software* innecesario, colocar *firewalls*, autenticar usuarios, colocar sistemas de prevención y ejecución de datos, sistemas de alarma, etc. Ya que la implementación de seguridad en un sistema añade restricciones operativas en relación incremental a su uso, el encargado del *hardening* de sistema debe incluir estos mecanismos de manera balanceada [11], [12].

### ***Firewall***

Un *firewall* o cortafuegos se establece como un componente fundamental en la seguridad de un sistema, ya que encarga de examinar el tráfico de red que entra y sale,

para luego determinar su legitimidad basándose en protocolos preestablecidos, los cuales determinan la permisión o denegación de paso [13].

Este dispositivo de seguridad se considera crítico y se ha mantenido como mecanismo de defensa primario durante más de dos décadas y media, mediante delinear una barrera protectora entre las redes internas (consideradas seguras) y las externas, consideradas de peligro potencial, de esta última, a la que pertenece la red mundial o vasta extensión de Internet, los *firewalls* se presentan de diversas formas, desde dispositivos de *hardware* hasta soluciones de *software*, incluidos modelos de *software* como servicio (SaaS), y pueden desplegarse en entornos de nube pública o privada [13].

### **Sistema de detección de intrusos**

Un sistema de detección de intrusos (IDS), sirve para supervisar de manera activa la actividad de la red, identificar y mitigar posibles amenazas o incumplimientos de las políticas establecidas, un IDS puede adoptar la forma de dispositivo de *hardware* o aplicación de *software* y sus resultados se centralizan para ser gestionados por medio de un sistema de gestión de eventos e información de seguridad (SIEM).

Los IDS son de varios tipos, lo cuales presentan variaciones como: sistemas de detección de intrusiones en red (NIDS), que analizan el tráfico de red entrante; los sistemas de detección de intrusiones basados en host (HIDS), que se centran en la supervisión de archivos cruciales del sistema operativo. Estos sistemas emplean diferentes métodos de detección, principalmente enfoques basados en firmas y en anomalías. Existen también los sistemas de prevención de intrusos (IPS), que a diferencia de los IDS, estos son capaces de responder a las intrusiones detectadas [14].

### **SSH**

Se denomina *Secure Shell* (SSH) a un protocolo que sirve como medio seguro para el inicio de sesión remoto y otros servicios de red a través de redes potencialmente inseguras, este se compone de 3 componentes principales: el protocolo de capa de transporte (SSH-Trans), responsable de la autenticación, confidencialidad e integridad del servidor y de la compresión opcional; el protocolo de autenticación de usuario (SSH-USERAUTH), que autentica a los usuarios ubicados como clientes del servidor y el protocolo de conexión (SSH-CONNECT), que facilita la multiplexación de túneles cifrados en canales lógicos.

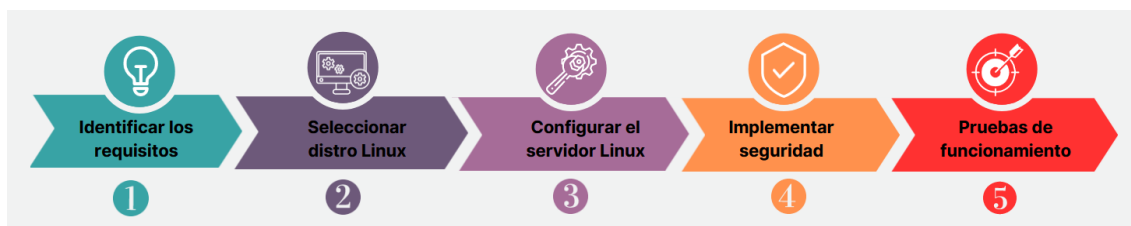
Estos componentes funcionan de manera jerárquica y cada protocolo basa sus decisiones en el anterior, estos funcionan tras establecer una conexión segura de capa de transporte, en la que luego el cliente inicia una solicitud de servicio, seguida de la



autenticación del usuario, lo que a su vez permite la coexistencia de nuevos protocolos con los ya existentes; el Protocolo de Conexión ofrece canales versátiles para varios propósitos, incluyendo sesiones *shell* interactivas seguras y reenvío de puertos TCP/IP [15].

## 2 METODOLOGÍA

En esta sección se detalla, en orden cronológico y desde el número uno al cinco el plan de objetivos necesarios para la consecución del proyecto, a continuación, como se demuestra en la Figura 2.1, se plantean los puntos clave que se necesitan para conseguir un avance bien definido, mediante su representación gráfica en un *roadmap* del avance lineal de los objetivos en secuencia lógica.



**Figura 2.1** Roadmap de objetivos

Como se ve en la Figura 2.1, el plan de objetivos consta de 5 puntos, los cuales se describen como:

Objetivo 1: Identificar los requisitos técnicos y funcionales, primero se realizó un análisis exhaustivo para identificar los requisitos técnicos y funcionales necesarios para la implementación exitosa del balanceador de carga.

Objetivo 2: Seleccionar la distribución de Linux y las herramientas de software, en esta sección se eligió la distribución de Linux más adecuada, a saber Ubuntu *Server*; continuación se seleccionaron las herramientas de software basadas en “*iproute 2*” necesarias para implementar el balanceador.

Objetivo 3: Configurar el servidor Linux como un balanceador de carga, aquí se llevó a cabo la configuración del servidor Linux mediante comandos para que funcione como un balanceador de carga de salida a Internet, se instaló y luego se configuró el *software*.

Objetivo 4: Implementar medidas de seguridad, para este apartado se realizaron las configuraciones en el siguiente orden : *firewall*, autenticación de usuarios, actualizaciones y parches del sistema operativo, remoción de *software*, puertos y

servicios no usados, implementación de IDS Suricata, implementación de tiempo NTP, servidor SSH, implementación de sistema de monitoreo centralizado.

Objetivo 5: Realizar pruebas de funcionamiento, al final, se ejecutarán pruebas para evaluar el trabajo del balanceador al menos en dos escenarios, para asegurar que cumpla con el alcance planteado.

### **3 RESULTADOS**

En la presente sección, se presenta el establecimiento secuencial de los requerimientos necesarios para la conformación del proyecto, se plantea el tipo de *software* y posible *hardware* necesarios, así como los elementos usados en la implementación de un balanceador de carga de salida a Internet basado en un servidor Linux, cuyo objetivo principal es gestionar de manera eficiente el tráfico de red peticionado desde una red local a través de dos o más salidas a Internet. El servidor con un sistema operativo Linux, configurado con un balanceador de carga es capaz de discriminar las peticiones de tráfico de Internet y dirigir las a la línea que más convenga de acuerdo con criterios establecidos de manera técnica.

#### **3.1 Identificación de los requerimientos para la implementación del prototipo**

Para la consecución del proyecto, después de un análisis exhaustivo, se identificaron todos los componentes necesarios para la implementación de un balanceador de carga de salida a Internet basado en un servidor Linux, el cual debe presentar las funcionalidades de control de cantidad de carga y conexión redundante, esto con el objetivo de presentar un uso eficiente de dos conexiones de salida Internet.

##### **Implementación de la plataforma**

Con la finalidad de implementar la plataforma, se necesita un sistema de emulación capaz de: primero emular dos proveedores de servicio de Internet (ISP), a continuación, el servidor al que llegan las conexiones, tanto de los ISP's como de la red local, sea una o varias máquinas; mediante un *switch* de capa 2, se requiere una computadora extra que cumplirá la función de administración del servidor mediante SSH, esta puede estar conectada en cualquier parte de la red, siempre que tenga acceso al servidor Linux.

## **Sistema operativo Linux**

Para comenzar, se necesita la elegir un sistema operativo basado en Linux, con el fin de que este trabaje como balanceador de carga de salida a Internet, el cual sea de código abierto y presente un manejo simple, eficiente y adecuado a las necesidades de una red local, la cual posea salida a Internet a través de dos conexiones. Además, debe cumplir también con que sea de fácil administración y que disponga de amplio soporte técnico y documental [16].

## **Módulo de enrutamiento**

El enrutador propuesto debe brindar un servicio de enrutamiento altamente flexible, con la capacidad de ser configurado según las determinaciones de la red en curso. En esta etapa inicial, se exige que el enrutador habilite la función de direccionar el tráfico desde una red local hacia dos conexiones *ethernet* que suministrarán acceso a Internet, procedente de distintos proveedores de manera simultánea. Además, en caso de falló, el enrutador debe permitir el acceso a Internet en cada conexión de manera independiente. Esta configuración implica una gestión eficiente de la tabla de enrutamiento y una capacidad de balanceo de carga para optimizar el uso de las conexiones disponibles, debe ser también escalable a nivel de pequeña o mediana empresa y que pueda ser adecuado a los cambios de red [17].

## **Módulo de balanceo**

Este debe ser seleccionado basado en la utilidad, escalabilidad (uso en pequeña y mediana empresa) y alcance de la implementación del prototipo, además de estar adecuado a estos parámetros, dicho algoritmo también debe permitir realizar un balanceo de carga con precisión media-alta, de rápida implementación y que sirva para todo tipo de tráfico o protocolo que alimente las necesidades de los usuarios. El módulo de balanceo debe ser capaz, cuando detecta un mal funcionamiento en cualquiera de las líneas, de dirigir el tráfico hacia aquella con mayor velocidad de descarga, disponibilidad o según el radio determinado por el administrador de red.

## **Sistema de monitoreo de tráfico de red**

A fin de comprobar el funcionamiento, se requiere de un sistema de medición que permita la captura de los paquetes enviados a través de las líneas de red, para esto, el sistema debe ejecutarse en las líneas de salida de los ISP de manera separada, la ejecución de captura debe ser temporizada y ejecutada al mismo tiempo en estas líneas. Al finalizar, el sistema de monitoreo debe crear un *log*, el cual ha de contener la información deseada de los paquetes, así como la cantidad de tráfico capturado en un

intervalo de tiempo determinado, todos los parámetros han de ser analizados y comparados para determinar la efectividad del funcionamiento del balanceador.

## **Seguridad**

En cuanto a la seguridad de *software*, el servidor debe ser asegurado con medidas básicas de *firewall*, configurado con seguridades como puertos cerrados por defecto, contraseñas y accesos de usuario bien definidos, comunicación SSH asegurada, aplicaciones y servicios bien definidos y controlados, sistema de detección de intrusos (IDS), un módulo de sincronización de tiempo de red preciso y un sistema de monitoreo fácil de acceso; de manera adicional, en caso de configurar el servidor de manera física, asegurar estos [11].

## **3.2 Determinación del hardware y software requeridos**

### **Selección del software**

#### **Selección de la plataforma de emulación del prototipo**

Con respecto al módulo de emulación del prototipo, se requiere que el sistema sea de estado de trabajo vigente, de preferencia con soporte documental, también debe presentar un estado de desarrollo activo y continuo, mediana o alta variedad de disponibilidad de dispositivos de emulación, alta eficiencia de utilización de recursos, buena precisión de simulación, buen nivel de realismo y facilidad de uso [18], [19]. A continuación, se presentan los criterios a ser estimados, descritos de manera más detallada, mediante considerar un sistema de calificación por puntaje, siendo mejor el más alto:

Selección inicial: Al tener en cuenta los *softwares* más usados, se eligen los 5 en común más recomendados, los cuales son: GNS-3, EVE-NG, Cisco Packet Tracer, Cisco VIRL, eNSP [18], [19], [20], [21], [22].

#### **Estado de actividad**

Se considera el estado de actividad como la vigencia de soporte que posee cada producto, el estado de la comunidad y la documentación; se asigna un número del 1 al 3 de acuerdo con los tres estados representativos de estas variables: inactivo (1), semiactivo (2) y activo (3), los mismos números representan estado bajo, medio y alto en sus respectivas categorías.

El *software* GNS-3 presenta un desarrollo activo con actualizaciones constantes y recientes (última versión 6 de noviembre de 2023), cuenta con una comunidad sólida, sin embargo su uso se ha visto desplazado por otras propuestas; su documentación es

amplia y detallada con implementaciones de nivel académico que son sencillas de encontrar [23], [24], [25], [26]; puntuación otorgada “2”. En cuanto a variedad de dispositivos, este presenta una variedad mediana, ya que no soporta la virtualización de imágenes más actuales. En cuanto a eficiencia de utilización de recursos, presenta mediana eficiencia pues desaprovecha los recursos de la máquina en un porcentaje considerable. Cuenta con alta precisión de simulación, alto nivel de realismo y su facilidad de uso es media [18], [19], [20], [22].

El *software* EVE-NG presenta un desarrollo activo, con actualizaciones constantes y recientes (última versión 16 de mayo de 2023); cuenta con una comunidad sólida, recomendado en uso corporativo; su documentación es amplia y detallada, con implementaciones de nivel académico sencillas de encontrar [27], [28], [29] [30]; puntuación otorgada “3”. En cuanto a variedad de dispositivos, presenta una variedad alta, ya que soporta la virtualización de imágenes más actuales. En cuanto a eficiencia de utilización de recursos, presenta alta eficiencia pues aprovecha bien los recursos de máquina, solo desaprovechando un pequeño porcentaje de esta. Cuenta con alta precisión de simulación, alto nivel de realismo y su facilidad de uso es media [18], [19], [20], [22].

El *software Packet Tracer* de Cisco presenta un desarrollo activo con actualizaciones constantes y recientes (última versión 2023); con una comunidad sólida, uso de nivel principiante, enfocado dispositivos Cisco; su documentación es amplia y detallada con implementaciones de nivel académico sencillas de encontrar [31], [32], [33]; puntuación otorgada “2”. En cuanto a variedad de dispositivos, presenta una variedad baja, ya que solo soporta la virtualización de imágenes de Cisco. En cuanto a eficiencia de utilización de recursos, presenta una eficiencia media, pues aprovecha los recursos de máquina al mismo tiempo que limita el número de nodos. Cuenta con mediana precisión de simulación, bajo nivel de realismo, y su facilidad de uso es media [18], [19], [20].

El *software VIRL* de Cisco, ahora migrado hacia la plataforma *Cisco Modeling Labs* (CML) presenta un desarrollo activo con actualizaciones constantes y recientes (última versión 20 de julio 2023); con una comunidad sólida, uso de nivel medio; su documentación es amplia y detallada [34], [35]; puntuación otorgada “3”. En cuanto a variedad de dispositivos, presenta una variedad media, ya que solo soporta de forma total la virtualización de imágenes de Cisco y un número básico de otras marcas. En cuanto a eficiencia de utilización de recursos, presenta una eficiencia media, pues aprovecha de manera bastante eficiente los recursos de máquina, sin embargo, limita el

número de nodos. Cuenta con mediana precisión de simulación, mediano nivel de realismo y su facilidad de uso es alta [18], [19], [20].

El *software* eNSP de Huawei, presenta un desarrollo detenido ya que no cuenta con actualizaciones nuevas (última versión 22 de junio 2021); su uso es limitado, enfocado en dispositivos Huawei; su documentación es amplia y detallada [36]; puntuación otorgada “1”. En cuanto a variedad de dispositivos, presenta una variedad baja, ya que solo soporta en su mayoría la virtualización de imágenes de Huawei y un número básico de otras marcas. En cuanto a eficiencia de utilización de recursos, presenta una eficiencia mediana pues aprovecha de manera bastante eficiente los recursos de máquina, sin embargo, limita el número de nodos. Cuenta con mediana precisión de simulación, bajo nivel de realismo, al no proponer escenarios tan reales y su facilidad de uso es alta [18], [19], [20].

**Tabla 3.1** Recopilación comparativa de emuladores recomendados [18], [19], [20], [22]

Emulador	Estado de actividad	Variedad de dispositivos	Eficiencia de máquina	Precisión de simulación	Real	Facilidad de uso	Suma Total
GNS3	2	2	2	3	3	2	14
EVE-NG	3	3	3	3	3	2	17
Packet Tracer	2	1	2	2	1	3	11
VIRL	3	2	2	2	2	3	14
eNSP	1	1	2	2	1	2	9

Como se observa en la Tabla 3.1, de acuerdo con la comparativa y basado en los seis parámetros descritos; el mejor *software* de emulación y por tanto el elegido para el presente proyecto es EVE-NG.

### Selección del sistema operativo

Para la elección del SO, se propone Ubuntu Server 22.04, debido a ser un sistema operativo Linux de baja utilización de recursos de máquina, este, especifica sus requisitos mínimos como 1 gigahercio (GHz) o más de frecuencia mono-núcleo de CPU, 1 gigabyte (GB) o más de memoria RAM y 2.5 gigabytes o más de disco duro [37]; está basado en Debian, el cual cuenta con una comunidad de uso muy extendida y por tanto soporte amplio (mayor que sus similares); soporta además varias arquitecturas: 32 y 64 bits, ARM v7, ARM64, POWER9 y POWER10, IBM zSystems, LinuxONE y RISC-V; tiene soporte de Canonical por 10 años, recibe actualizaciones frecuentes, posee

seguridad OpenSSL 3.0 para comunicación encriptada y mejoras con SmartNIC para el módulo de red Netplan [38], [39].

También se eligió este SO debido a su módulo de enrutamiento, denominado netplan, este viene incorporado en el kernel de Ubuntu y presenta la elección de *renderer* o manejador de red, de los disponibles: *systemd-networkd* y *NetworkManager*, se eligió *systemd-networkd*, ya que este es mejor en gestionar redes de tipo servidor, máquinas virtuales y entornos estáticos; mientras que el segundo es más enfocado a entornos dinámicos, como usuarios y dispositivos finales que requieren constante conexión y desconexión de cables y redes inalámbricas (WiFi) en diferentes dispositivos, cambios constantes de nombre del identificador del set de servicio (SSID), etc. [40], [8], [41], [42].

### **Selección de módulos de balanceo de carga**

Como módulo principal del balanceador, se usó el *iproute2*, este es un módulo de código abierto, el cual viene embebido en Ubuntu y en muchos SO's basados en Linux, este se eligió debido a que provee un repositorio de herramientas y utilidades que buscan añadir al kernel de Linux un módulo de manejo de red con calidad de servicio (QoS) [9]. De dichas herramientas, las principales que se escogieron son *iptables* y *fwmark*, para lograr que las sesiones NAT que vienen de la red de área local (LAN) interna, funcionen como conexiones de salida, para, al mismo tiempo asignar pesos al tráfico de red en cada una de estas; de manera adicional, se usó la herramienta *foolsm*, como sistema de comprobación de conectividad de red de con el propósito de lograr un método de *failover* en las conexiones de salida de Internet [43].

### **Selección de módulos de seguridad**

Para la seguridad del servidor basado en Ubuntu Server 22.04, se eligieron además de las medidas y módulos recomendados por los desarrolladores (en el caso de que el kernel contase con los indicados por defecto), algunas adiciones necesarias puntuales. De esta manera los módulos elegidos fueron: para la seguridad de aplicaciones, el módulo *AppArmor* de Ubuntu por defecto; *nmap* para pruebas de servicios; se eligió Suricata sobre Snort como IDS, debido a su facilidad de uso y mayor funcionalidad; el módulo *ntp* de Ubuntu para la sincronización horaria; el módulo *iptables* para la configuración del *firewall*; RSA para la encriptación de comunicación SSH, esto debido a su manejo local; el programa *logwatch*, como sistema de monitoreo centralizado, gracias a su flexibilidad, extensión de uso y facilidad de manejo [11], [12], [44].

## **Selección del hardware**

Ya que este proyecto se tiene como directiva primaria presentar una simulación de una implementación de servidor, no se considera el despliegue de un hardware en específico, sin embargo, como se observa en las especificaciones del SO utilizado [37]:

CPU: 1 (GHz) mínimo

RAM: 1 (GB) mínimo

Disco duro: mínimo de 2.5 (GB)

Se recomienda a futuro o si se tiene en mente un planteamiento físico, el uso de al menos un Raspberry Pi 3 Modelo B+ en adelante, en conjunto con un módulo de NIC PCIe o usb, para añadir más puertos de red Ethernet o WiFi [45], [46].

### **3.3 Diseño del prototipo de balanceador de carga de salida a Internet**

A continuación, se refiere de forma minuciosa el proceso seguido para la obtención del entorno funcional requerido para la implementación del proyecto.

#### **Preparativos del entorno de simulación**

Primero se dispuso del software simulador de redes EVE-NG virtualizado, en su versión 5.17.8 (Figura 3.1), a continuación, se dispuso de un espacio de 60 (GB) por defecto, del cual, como se observa se ocupó 2.6 GB para el sistema y del sobrante el 71,6 % de espacio total, para esto se ocuparon 2 imágenes de *routers*, 1 imagen del SO Ubuntu Desktop 22.04, una imagen del SO Ubuntu Server 22.04 y una imagen del SO Windows 10.

Debido a que tanto la virtualización como la configuración del simulador van más allá del alcance de este proyecto, los aspectos referentes se describirán de manera somera, lo suficiente para proveer el contexto necesario para comprender el funcionamiento del balanceador de carga.



```
Home x EVE-NG 2 x
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.17.8-eve-ng-uksm-wg+ x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Wed 17 Jan 2024 06:06:51 PM UTC

System load:  0.07          Processes:           294
Usage of /:   71.6% of 57.40GB Users logged in:      0
Memory usage: 4%          IPv4 address for pnet0: 192.168.152.153
Swap usage:   0%

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
  just raised the bar for easy, resilient and secure K8s cluster deployment.

  https://ubuntu.com/engage/secure-kubernetes-at-the-edge

* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

  https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Tue Jan 16 17:35:02 UTC 2024 on tty1
root@eve-ng:~#
```

**Figura 3.1** Eve-ng virtualizado

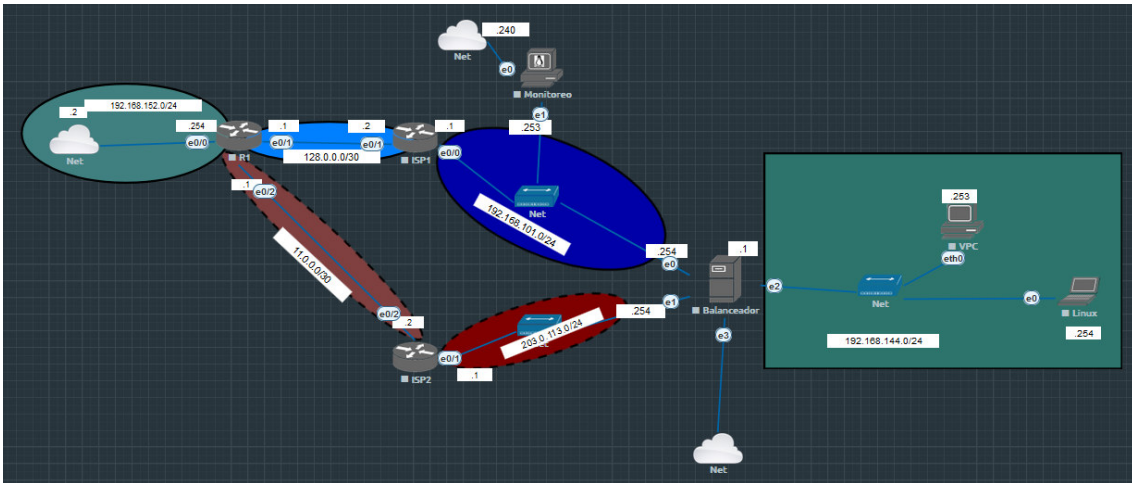
### **Diagrama de red**

A continuación, se dispuso la red con el objetivo de simular dos conexiones a Internet hacia el balanceador de carga, es decir colocar dos redes separadas que se manejen de manera independiente una de otra, de manera similar a un entorno real se contrata dos ISP, estos pueden llegar a las premisas del cliente con el equipo terminal, de manera usual un *Optical Node Terminal* (ONT) o solo la línea de suministro sea por cable, fibra o cobre.

Como se observa en la Figura 3.2, se consideró de izquierda a derecha una salida a Internet (Net) en forma de nube, esta provee el servicio al primer router (R1), el cual a su vez divide este en 2 conexiones hacia 2 routers distintos a manera de ISP's; el ISP1 se considera una red privada por lo tanto ejerce el rol de equipo terminal en las premisas del cliente; el router ISP2 se considera un equipo fuera de las premisas del cliente, por lo tanto la línea es una red pública; se asignaron las direcciones IP correspondientes tanto públicas como privadas.

En la red correspondiente al ISP1 (azul), se configuró un terminal de monitoreo y configuración del servidor, se lo hizo aquí ya que al ser una red privada y fuera de la red de los clientes, provee de cierta seguridad y conveniencia para las tareas mencionadas,

mientras que la red de la zona roja al ser pública, junto a la red de la zona verde de la derecha, al ser red de clientes, se consideraron como redes no aptas para el acceso de configuración y administración.

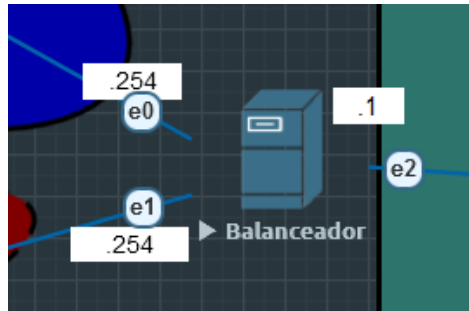


**Figura 3.2** Diagrama de configuración de la red

Cabe aclarar que tanto las nubes de acceso a Internet de la estación de monitoreo (Ubuntu Desktop), como la nube del servidor (Ubuntu Server), se añadieron para la configuración inicial de Ubuntu 22.04, puesto que en la configuración inicial se necesita un gran ancho de banda para la instalación y actualización de paquetes, requisito que no cumple la conexión de Internet al pasar por los routers ISP1 e ISP2, esto sucede porque, al ser la versión de prueba de EVE-NG, esta posee una restricción de ancho de banda, limitada primero a alrededor de 10 a 20 Megabits (Mbps) por segundo, tasa que varía aún más al estar ligada al rendimiento del procesador de la máquina anfitriona, por lo tanto, luego de las configuraciones iniciales, estas nubes pueden suprimirse y como se observa en la sección 3.4 más adelante, estas no constan, pues en la implementación se simula un ambiente de prueba real.

### **Diseño del modelo de balanceo de carga**

Una vez terminadas las consideraciones de trabajo iniciales, se diseñó el funcionamiento del balanceador de carga (Figura 3.3), para esto se consideró la instalación del SO Linux (Ubuntu Server 22.04) realizada y se procedió a configurar la red.



**Figura 3.3** Balanceador de carga

### Configuración de puertos de red

Los puertos de red de Ubuntu server 22.04 trabajan con el *software* de base incluido en el kernel, denominado “netplan”, este se maneja mediante lenguaje yaml en la ubicación “/etc/netplan/”, aquí, mediante un editor de texto como vim o nano se modifican y añaden las variables necesarias. Debido a el uso de la máquina (servidor), los puertos a usar se configuran con direcciones de red estáticas, por lo tanto, se omite la opción para protocolo de control de host dinámico (dhcp); como se puede ver en la Figura 3.3, primero, se necesita conectividad en las interfaces ethernet, primero en la Figura 3.4; para la interfaz e0, corresponde a la interfaz ens3; para la Figura 3.5, e1 corresponde a la interfaz ens4 y e3 corresponde a la interfaz ens5 [17]. Para el documento *00-installer-config.yaml*, se explica la plantilla de configuración de la siguiente manera:

- En el primer nivel se especifica el inicio de la configuración de red con la orden “network” [17], [47].
- En el segundo nivel se encuentran las configuraciones generales de la red; de estas se utilizó la línea “versión”, la cual debe ser la segunda (2), debido a que usa un archivo yaml diferente a la primera versión, la cual se enfoca en el despliegue de SO’s de tipo *metal as a service* (MaaS) mediante el uso del personalizador curtin de Canonical Ltd. [47]; la línea `renderer`, el cual se especificó como `networkd`, para trabajar con el manejador `systemd-networkd`, el cual está mejor adaptado a los eventos, configuraciones y necesidades de uso en servidores [17], [40], [42], [47]; `ethernets:`, el cual da inicio al tercer nivel de configuración, mediante especificar el tipo de puertos (estos pueden ser de otros tipos como por ejemplo wifis) [17], [42], [47].
- En el tercer nivel se especifican los nombres de los puertos ethernet, en este caso `ens3` (para e0), `ens4` (para e1), `ens5` (para e2).

El cuarto, quinto y sexto nivel están ligados, ya que estos especifican las opciones dedicadas a cada puerto ethernet, para la presente configuración, de arriba hacia abajo y por cada sección pertinente se usó:

- La línea “dhcp: no”, para especificar el dhcp apagado y asignar direcciones IP estáticas [17], [42], [47].
- La línea “addresses:” para indicar la creación del IP de *host*, en el subsiguiente nivel con la dirección deseada al iniciar con el carácter “-” [17], [42], [47].
- La línea “nameservers:” para indicar la entrada de servidores de nombre de dominio (DNS); la línea “addresses:” para indicar la creación del IP del DNS, en el subsiguiente nivel, la dirección deseada iniciando con el carácter “-”, en este caso se coloca la dirección IP del DNS del ISP [17], [42], [47].
- La línea “routes:”, marca el inicio de introducción de las rutas que se desea que siga el enrutado, en el subsiguiente nivel se introduce la línea “- to: <dirección de red>”, que indica la dirección de red hacia la cual el puerto va a tener acceso; en la siguiente línea “via: <dirección ip del gateway>”, indica la dirección del *gateway* de la red; la línea “metric: <número>”, indica la preferencia de ruta que se desea el tráfico siga, se toma en cuenta el menor número como la ruta preferida; la línea “table: <número>” indica la tabla de enrutamiento que se va a usar, esta puede ser una de las predefinidas o como en este caso una nueva, la cual se creará más adelante [17], [42], [47].
- La línea “routing-policy:”, indica el inicio de las políticas de enrutamiento que se va a aplicar a la interfaz; la subsiguiente línea “- from: <dirección de red>”, que indica la dirección de red desde la cual el tráfico será enrutado; la línea “table: <número>” indica la tabla de enrutamiento que se va a usar [17], [42], [47].

```
root@ubuntu:~# vi /etc/netplan/00-installer-config.yaml
```

```
# This is the network config written by 'subiquity'
network:
  version: 2
  renderer: networkd
  ethernets:
    ens3:
      dhcp4: no
      addresses:
        - 192.168.101.254/24
      nameservers:
        addresses:
          - 192.168.101.1
      routes:
        - to: 192.168.101.0/24
          via: 192.168.101.1
          metric: 200
          table: 10
      routing-policy:
        - from: 192.168.144.0/24
          table: 10
```

**Figura 3.4** Configuración de red, netplan (general, ens3)

```
ens4:
  dhcp4: no
  addresses:
    - 203.0.113.254/24
  nameservers:
    addresses:
      - 203.0.113.1
  routes:
    - to: 203.0.113.0/24
      via: 203.0.113.1
      metric: 300
      table: 11
  routing-policy:
    - from: 192.168.144.0/24
      table: 11

ens5:
  dhcp4: no
  addresses:
    - 192.168.144.1/24
  nameservers:
    addresses:
      - 8.8.8.8
```

**Figura 3.5** Configuración de red, netplan (ens4, ens5)

Como se muestra en la Figura 3.4, se usan dos tablas de enrutamiento, una por cada salida de Internet, además, con la línea “from”, en ambas se dirige el tráfico desde la red que se desea acceso de Internet (la red de clientes), hacia cada una de las tablas de enrutamiento; la última interfaz (ens5), se configura solo hasta la sección nameservers. Una vez terminada la configuración del archivo, se guarda y se sale de este, con el comando “netpan try”, se comprueba la correcta configuración y se aplica la presionar la tecla “enter” (Figura 3.6) [17], [42], [47].

```
root@ubuntu:~# netplan try
Press ENTER before the timeout to accept the new configuration

Changes will revert in 119 seconds
Configuration accepted.
root@ubuntu:~#
```

**Figura 3.6** Aplicación de la configuración de red

### Acceso dividido

Para continuar con el proceso de organización, para facilidad de configuración y administración se habilita SSH desde un terminal externo (monitor en la Figura 3.2), para continuar con la configuración del servidor desde el terminal monitor, el mismo que cuenta con el SO Ubuntu Desktop 22.04; primero, como se ve en la Figura 3.7, para instalar el servidor SSH en el servidor, se establece el comando “apt install openssh-server”, mientras que, para instalar el cliente SSH en el terminal monitor se lo realiza con el comando “apt install openssh-client”, para ingresar desde el terminal ingresar el comando “sudo ssh <usuario>@dirección\_ip” [48].

```
root@ubuntu:~# apt install openssh-server_
user@ubuntu22-desktop:~$ sudo -i
[sudo] password for user:
root@ubuntu22-desktop:~# apt install openssh-client
```

**Figura 3.7** Instalación del servicio SSH

Una vez definido el comportamiento de las interfaces de red, se necesita activar el acceso compartido de las conexiones de Internet, para esto se habilita el *IP forwarding*, es decir, para que los paquetes IP puedan pasar desde la red de clientes hacia las de servicio de Internet y viceversa [49], [50], [51]. Esto se hace de manera temporal con la orden “sysctl -w net.ipv4.ip\_forward=1” o de manera permanente, persistente en cada reinicio, mediante editar el archivo ubicado en /etc/sysctl.conf y descomentar la línea, para que se muestre como “net.ipv4.ip\_forward = 1”, tal como se representa en la Figura 3.8 [51].



```
root@ubuntu:~# ip rule add from 192.168.101.254 table ISP1
root@ubuntu:~# ip rule add from 203.0.113.254 table ISP2
```

**Figura 3.10** Asignación de IP's a tablas de enrutamiento

Para que la red cliente (LAN) pueda acceder a través de ambas salidas de Internet, se crean sus rutas respectivas, ligadas a sus tablas de ruteo, esto con los comandos de la Figura 3.11 [50], [53].

```
root@ubuntu:~# ip route add 192.168.144.0/24 via 192.168.101.1 table ISP1
root@ubuntu:~# ip route add 192.168.144.0/24 via 203.0.113.1 table ISP2
```

**Figura 3.11** Asignación de rutas

Ya que aún no existe salida a Internet, se necesita un método por el cual el tráfico alcance la red interna; mediante habilitar el modo de traducción de dirección de red (NAT) para las interfaces de salida Internet, se logra mantener el tráfico desde el exterior de la red hacia adentro (red LAN) sin que ningún puerto permanezca abierto, ya que solo el router Linux conoce las traducciones, mientras que el acceso de la red interna hacia el exterior se mantiene irrestricto, para esto se aplican los comandos de la Figura 3.12 [50], [53], [54].

```
root@ubuntu:~# iptables -t nat -A POSTROUTING -o ens3 -j MASQUERADE
root@ubuntu:~# iptables -t nat -A POSTROUTING -o ens4 -j MASQUERADE
```

**Figura 3.12** Habilitación de NAT

Donde “-t” se usa para indicar la tabla de correspondencia de paquetes y nat es la asignación; “-A” se usa para añadir una o varias reglas al final de la cadena, la cual usa la regla “POSTROUTING” para indicar que debe aplicarse en los paquetes de salida; “-o” indica *out* o salida de la interfaz, seguido del nombre de la misma; “-j” indica *jump* o salto y a continuación indicar la regla “MASQUERADE”, la cual especifica asignación a la dirección IP que pertenece a la interfaz de salida del paquete [55], [56].

Para salvar las reglas IP se usa el programa “iptables persistent”, mediante la instalación de este, se mantienen las reglas en cada reinicio; con los comandos “ip iptables-save -f /etc/iptables/rules.v4” y “netfilter-persistent save” se guardan las normas después de cada configuración (Figura 3.13) [57].

```
root@ubuntu:~# apt install iptables-persistent
root@ubuntu:~# iptables-save -f /etc/iptables/rules.v4
root@ubuntu:~# netfilter-persistent save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables save
root@ubuntu:~# █
```

**Figura 3.13** Guardar las normas



## Balancedor de carga

Para la instalación del balanceador de carga se usó el paquete “Net-ISP-Balance”, este permite automatizar las configuraciones del módulo *iproute2*, mediante scripts ejecutados en lenguaje Pearl, para empezar la configuración, nótese en la Figura 3.14, se ejecutan las órdenes; “apt install unzip” para descargar el descompresor unzip; “wget https://github.com/lstein/Net-ISP-Balance/archive/master.zip” para adquirir el archivo; “unzip master.zip” para descomprimir el paquete; “cd Net-ISP-Balance-master/” para ingresar a la ubicación que se acaba de descomprimir [50].

```
root@ubuntu:~# apt install unzip
root@ubuntu:~# wget https://github.com/lstein/Net-ISP-Balance/archive/master.zip
--2024-01-28 14:01:17-- https://github.com/lstein/Net-ISP-Balance/archive/master.zip
Resolving github.com (github.com)... 140.82.114.4
Connecting to github.com (github.com)[140.82.114.4]:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/lstein/Net-ISP-Balance/zip/refs/heads/master [following]
--2024-01-28 14:01:17-- https://codeload.github.com/lstein/Net-ISP-Balance/zip/refs/heads/master
Resolving codeload.github.com (codeload.github.com)... 140.82.112.10
Connecting to codeload.github.com (codeload.github.com)[140.82.112.10]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/zip]
Saving to: 'master.zip.1'

master.zip.1           [ <=> ] 104.40K  203KB/s  in 0.5s

2024-01-28 14:01:18 (203 KB/s) - 'master.zip.1' saved [106901]

root@ubuntu:~# unzip master.zip
Archive:  master.zip
  extracting: Net-ISP-Balance-master/etc/balance/routes/01.local_routes.conf
  extracting: Net-ISP-Balance-master/etc/balance/routes/02.local_routes.conf
  inflating: Net-ISP-Balance-master/etc/balance/routes/03.local_rules.pl
  inflating: Net-ISP-Balance-master/etc/balance/failover.conf
  inflating: Net-ISP-Balance-master/etc/interfaces
root@ubuntu:~#
root@ubuntu:~# cd Net-ISP-Balance-master/
root@ubuntu:~/Net-ISP-Balance-master#
```

Figura 3.14 Adquirir el paquete Net-ISP-Balance

A continuación, se adquieren herramientas como *C compiler* y *make*, necesarias para ejecutar el paquete en lenguaje Pearl. Estas se consiguen con el comando “apt-get install build-essential”; seguido del comando “apt install perl” para instalar las dependencias del lenguaje; con el comando “apt-get install libmodule-build-perl” se instalan las librerías necesarias de perl; de manera adicional se debe instalar la herramienta “cpanminus” con el comando “apt install cpanminus”, con la cual se ejecutan los *scripts* cpan, que sirven como repositorios de módulos perl; en este caso concreto se requiere adquirir el módulo Netmask con el comando “cpanm Net::Netmask” (Figura 3.15) [50], [58], [59], [60].

```
root@ubuntu:~/Net-ISP-Balance-master# apt-get install build-essential
root@ubuntu:~/Net-ISP-Balance-master# apt install perl
root@ubuntu:~/Net-ISP-Balance-master# apt-get install libmodule-build-perl
root@ubuntu:~/Net-ISP-Balance-master# apt install cpanminus
root@ubuntu:~/Net-ISP-Balance-master# cpanm Net::Netmask
```

Figura 3.15 Instalación de herramientas necesarias

Luego se necesitan las dependencias del balanceador de carga, las cuales se instalan con los comandos de la Figura 3.16, en su respectivo orden son: el archivo principal “Build.PL” el cual se describe a detalle en la sección *Configuración más adelante* (Anexo III), las dependencias y finalizar con prueba e instalación [50].

```
root@ubuntu:~/Net-ISP-Balance-master# perl Build.PL
WARNING: the following files are missing in your kit:
    META.json
    META.yml
Please inform the author.

Created MYMETA.yml and MYMETA.json
Creating new 'Build' script for 'Net-ISP-Balance' version '1.32'
root@ubuntu:~/Net-ISP-Balance-master# ./Build intalldeps
No action 'intalldeps' defined, try running the 'help' action.
root@ubuntu:~/Net-ISP-Balance-master# ./Build installdeps
All prerequisites satisfied
root@ubuntu:~/Net-ISP-Balance-master# ./Build test
skip /etc/network/balance.conf: already installed
root@ubuntu:~/Net-ISP-Balance-master# ./Build install
Building Net-ISP-Balance
skip /etc/network/balance.conf: already installed
Building Net-ISP-Balance
Installing /usr/local/man/man3/Net::ISP::Balance::ConfigData.3pm
root@ubuntu:~/Net-ISP-Balance-master#
```

**Figura 3.16** Comandos de dependencias del balanceador de carga

Una vez instalado, los parámetros de configuración se encuentran en la ubicación “/etc/network/balance.conf”, con un editor de texto se configura la red, tal como se presenta en la Figura 3.17, la configuración consta de una tabla de seis columnas, la primera especifica el servicio, la segunda la interfaz, la tercera el rol (isp en caso de ser salida a Internet, lan para la red interna), la cuarta, la dirección de ping a la cual el sistema de *failover* usa para detectar si existe una conexión activa (en este caso la dirección de *gateway*), en la quinta columna se especifica el peso de cada interfaz, es decir cuanto tráfico pasará a través de este en relación a los otros pesos y la sexta columna, se especifica la dirección de *gateway* de salida de Internet del balanceador de carga [50], [61].

```

root@ubuntu:/# vi /etc/network/balance.conf
# service device role ping-ip weight gateway
ISP1 ens4 isp 203.0.113.1 1 203.0.113.1
ISP2 ens3 isp 192.168.101.1 2 192.168.101.1
LAN1 ens5 lan
#LAN2 eth2 lan

# The mode controls whether to operate in "balanced" mode in which traffic
# is distributed among the isp services proportional to the weights, or "failover"
# mode in which all traffic is routed through the highest weighted service and
# switches to lower-weighted services only when the higher one(s) are unavailable.
mode=balanced
#mode=failover

## The "forwarding_group" option gives you fine control over how
## packets are forwarded. See the online docs for details. :lan
## means all interfaces marked as "lan" :isp means all interfaces
## marked as "isp" the default (shown below) allows forwarding among
## all lan and isp interfaces

forwarding_group=:lan :isp

## The options below are passed to lsm, among others.
## the defaults are shown. To change them, uncomment
## and edit.

#warn_email=root@localhost
interval_ms=250

```

**Figura 3.17** Parámetros del balanceador de carga

Una vez editada la configuración, se guarda el archivo y se ejecuta con el comando “load\_balance.pl”, este *script* en lenguaje Pearl se encarga de ejecutar las configuraciones, de ruteo multi-camino, por ejemplo, para el caso actual de configuración (Figura 3.17), este pasa a ejecutar los comandos: “nextthop via 192.168.101.254 dev ens3 weight 2” y “nextthop via 203.0.113.254 dev ens4 weight 1”, para asignar el peso de balanceo de las salidas de Internet; a la vez que define cinco cadenas de “iptables” destinadas al marcado de paquetes, una de inicio y una para finalizar; 3 intermedias para marcar los paquetes en forma de contador y dividir el tráfico en relación 2:1, de la manera que indica la **Figura 3.18**.

```

iptables -A PREROUTING -t mangle -j CONNMARK --restore-mark

iptables -A PREROUTING -t mangle -m mark --mark 0x0 -m statistic --mode nth --every
4 --packet 0 -j MARK --set-mark 1

iptables -A PREROUTING -t mangle -m mark --mark 0x0 -m statistic --mode nth --every
4 --packet 1 -j MARK --set-mark 2

iptables -A PREROUTING -t mangle -m mark --mark 0x0 -m statistic --mode nth --every
4 --packet 2 -j MARK --set-mark 1

iptables -A POSTROUTING -t mangle -j CONNMARK --save-mark

```

**Figura 3.18** Configuración de ruteo multicamino (marcado de paquetes) [54]

Donde “PREROUTING”, indica que se aplica a paquetes antes del ruteo; “mangle” para indicar la tabla correspondiente a alteración de paquetes especializada; “CONNMARK” para indicar el módulo de marcado de paquetes; “--restore-mark” y “--save-mark”, para

reiniciar el marcado y guardar de manera respectiva. Mientras que en las cadenas intermedias “-m” indica match o que coincida; “mark” indica marca de un paquete; “nth” módulo que indica que “n” número de paquetes tipo “th”, que en conjunto con “-every”, realiza un conteo, mientras trabaja para el módulo mark, para marcar estos paquetes; “-set-mark” se asigna dependiendo de cuantas interfaces existan, presente caso dos, en relación 2:1, entonces se marcan 2 paquetes con el numero 1 y uno con el número 2 [50], [54], [55], [56], [61].

De manera similar se utiliza el módulo IP tables para crear el failover, es decir con los comandos de la Figura 3.19.

```
iptables -N MARKING
iptables -A PREROUTING -t mangle -j CONNMARK --restore-mark
iptables -A PREROUTING -t mangle -m mark --mark 0x0 -j MARKING
iptables -A MARKING -t mangle -m condition --condition link1_up -m nth --counter 1 --
every 4 --packet 1 -j MARK --set-mark 1
iptables -A MARKING -t mangle -m condition ! --condition link1_up -m nth --counter 1 --
every 4 --packet 1 -j MARK --set-mark 1
iptables -A MARKING -t mangle -m condition --condition link2_up -m nth --counter 1 --
every 4 --packet 2 -j MARK --set-mark 2
iptables -A MARKING -t mangle -m condition ! --condition link2_up -m nth --counter 1 --
every 4 --packet 2 -j MARK --set-mark 1
iptables -A MARKING -t mangle -m condition --condition link1_up -m nth --counter 1 --
every 4 --packet 3 -j MARK --set-mark 1
iptables -A MARKING -t mangle -m condition ! --condition link1_up -m nth --counter 1 --
every 4 --packet 3 -j MARK --set-mark 2
iptables -A POSTROUTING -t mangle -j CONNMARK --save-mark

iptables -A POSTROUTING -t nat -m mark --mark 1 -j SNAT 192.168.101.254
iptables -A POSTROUTING -t nat -m mark --mark 2 -j SNAT 203.0.113.254
```

**Figura 3.19** Configuración de módulo de *failover* [54]

Donde “-N”, indica nueva cadena, “condition” indica que cumpla una condición o condición negada, la cual se indica si cumple con “link2\_up”, entonces se envíe por “mark 1” o “mark 2”; “SNAT” se usa para asignar de forma estática a las direcciones de salida de Internet con nateo de fuente; mediante el mismo sistema se asignan las reglas de firewall. La configuración detallada de enrutamiento, planteada en lenguaje Pearl, se observa en la sección Anexo II.II Código QR de la secuencia de código del archivo Balance.PM, a partir de la línea 1703 del código [50], [54], [55], [56], [61], [62].

## Seguridades generales

### App Armor

Se implementa App Armor como módulo de seguridad de bloqueo de aplicaciones, debido a que Ubuntu Server 22.04, viene instalado por defecto varios perfiles de aplicación que se debe comprobar y en caso de no ejecutarse, se debe configurar para que se coloquen en modo *enforce*, el cual bloquea la ejecución no autorizada de aplicaciones, a la vez que se debe evitar aplicaciones en modo *complain*, es decir que solo producen avisos. Como se observa en la Figura 3.20, existen 32 perfiles, todos en modo *enforce*, de entre los cuales destacan perfiles para *snapped* y *lxd*, que sirven para contenerización, perfiles de red por defecto para los servicios instalados: *ntp* y *tcpdump*, de estos dos últimos, solo el proceso de *ntp* se encuentra activado y en modo *enforcé*, *tcpdump* requiere acceso de usuario privilegiado [11], [12], [63].

```

root@ubuntu:/# aa-status
apparmor module is loaded.
32 profiles are loaded.
32 profiles are in enforce mode.
/snap/snapd/20290/usr/lib/snapd/snap-confine
/snap/snapd/20290/usr/lib/snapd/snap-confine//mount-namespace-capture-helper
/snap/snapd/20671/usr/lib/snapd/snap-confine
/snap/snapd/20671/usr/lib/snapd/snap-confine//mount-namespace-capture-helper
/usr/bin/man
/usr/lib/NetworkManager/nm-dhcp-client.action
/usr/lib/NetworkManager/nm-dhcp-helper
/usr/lib/connman/scripts/dhclient-script
/usr/lib/snapd/snap-confine
/usr/lib/snapd/snap-confine//mount-namespace-capture-helper
/usr/sbin/ntpd
/{usr}sbin/dhclient
lsb_release
man_filter
man_groff
nvidia_modprobe
nvidia_modprobe//kmod
snap-update-ns.lxd
snap.lxd.activate
snap.lxd.benchmark
snap.lxd.buginfo
snap.lxd.check-kernel
snap.lxd.daemon
snap.lxd.hook.configure
snap.lxd.hook.install
snap.lxd.hook.remove
snap.lxd.lxc
snap.lxd.lxc-to-lxd
snap.lxd.lxd
snap.lxd.migrate
snap.lxd.user-daemon
tcpdump
0 profiles are in complain mode.
0 profiles are in kill mode.
0 profiles are in unconfined mode.
1 processes have profiles defined.
1 processes are in enforce mode.
  /usr/sbin/ntpd (684)
0 processes are in complain mode.
0 processes are unconfined but have a profile defined.
0 processes are in mixed mode.
0 processes are in kill mode.
root@ubuntu:/#

```

**Figura 3.20** Revisión del estado de App Armor

Si se desea modificar perfiles, se buscan y se editan los archivos ubicados en la ubicación “/etc/apparmor.d/”, para revisar el historial se aplica el comando “sudo journalctl -fx” [11], [12], [63].

### Remover software inútil y servicios

Se empieza al listar los servicios escuchando de manera activa la red, estos se observan mediante la implementación de la orden en el shell “lsof”, como se observa en su Figura 3.21, existen 4 procesos, el primero “systemd-resolve”, embebido, relacionado con el manejo de red; el segundo, “ntpd” para el servidor ntp; el tercero “ssh” para este servicio y el último “master” para el servidor postfix de logwatch. Como todos son servicios utilizados de manera activa no se elimina ninguno [11], [12].

```

root@ubuntu:~# sudo lsof -i -P -n | grep -v ESTABLISHED
COMMAND  PID      USER     FD  TYPE DEVICE SIZE/OFF  NODE NAME
systemd-r 631    systemd-resolve 13u  IPv4 19449      0t0  UDP 127.0.0.53:53
systemd-r 631    systemd-resolve 14u  IPv4 19450      0t0  TCP 127.0.0.53:53 (LISTEN)
systemd-r 631    systemd-resolve 21u  IPv4 25038      0t0  UDP 203.0.113.254:42995->203.0.113.1:53
systemd-r 631    systemd-resolve 22u  IPv4 25040      0t0  UDP 192.168.101.254:50601->192.168.101.1:53
systemd-r 631    systemd-resolve 25u  IPv4 25973      0t0  UDP 203.0.113.254:53045->203.0.113.1:53
systemd-r 631    systemd-resolve 26u  IPv4 25974      0t0  UDP 192.168.101.254:33832->192.168.101.1:53
ntpd      711      ntp       4u  IPv4 25047      0t0  UDP 127.0.0.1:40482->127.0.0.53:53
ntpd      711      ntp      16u  IPv6 21255      0t0  UDP *:123
ntpd      711      ntp      17u  IPv4 21258      0t0  UDP *:123
ntpd      711      ntp      18u  IPv4 21262      0t0  UDP 127.0.0.1:123
ntpd      711      ntp      19u  IPv4 21264      0t0  UDP 192.168.101.254:123
ntpd      711      ntp      20u  IPv4 21266      0t0  UDP 203.0.113.254:123
ntpd      711      ntp      21u  IPv4 21268      0t0  UDP 192.168.144.1:123
ntpd      711      ntp      22u  IPv4 21270      0t0  UDP 192.168.152.244:123
ntpd      711      ntp      23u  IPv6 21272      0t0  UDP [::1]:123
ntpd      711      ntp      24u  IPv6 21274      0t0  UDP [fe80::250:ff:fe00:400]:123
ntpd      711      ntp      25u  IPv6 21285      0t0  UDP [fe80::250:ff:fe00:401]:123
ntpd      711      ntp      26u  IPv6 21287      0t0  UDP [fe80::250:ff:fe00:402]:123
ntpd      711      ntp      27u  IPv6 21290      0t0  UDP [fe80::250:ff:fe00:403]:123
sshd      723      root      3u  IPv4 21807      0t0  TCP *:22 (LISTEN)
sshd      723      root      4u  IPv6 21809      0t0  TCP *:22 (LISTEN)
master    1454     root     13u  IPv4 22995      0t0  TCP 127.0.0.1:25 (LISTEN)
master    1454     root     14u  IPv6 22996      0t0  TCP [::1]:25 (LISTEN)
root@ubuntu:~#

```

Figura 3.21 Comando lsof

## Protocolos y Servicios sin encriptación

Los protocolos y servicios que tienen acceso a la red y además no poseen encriptación. se pueden revisar con el comando “nmap” como en la Figura 3.22, donde “-v” indica incrementar verbosidad; “-sV” prueba puertos abiertos para determinar servicio y versión; aquí se observa como el único puerto abierto el 22 para SSH y 65534 puertos cerrados, con SSH encriptado para su uso como se observará más adelante [11], [12], [64].

```

root@ubuntu:~# nmap -v -sV -p- 192.168.101.254
Starting Nmap 7.80 ( https://nmap.org ) at 2024-01-29 01:58 -05
NSE: Loaded 45 scripts for scanning.
Initiating Parallel DNS resolution of 1 host. at 01:58
Completed Parallel DNS resolution of 1 host. at 01:59, 10.30s elapsed
Initiating SYN Stealth Scan at 01:59
Scanning 192.168.101.254 [65535 ports]
Discovered open port 22/tcp on 192.168.101.254
Completed SYN Stealth Scan at 01:59, 0.53s elapsed (65535 total ports)
Initiating Service scan at 01:59
Scanning 1 service on 192.168.101.254
Completed Service scan at 01:59, 0.02s elapsed (1 service on 1 host)
NSE: Script scanning 192.168.101.254.
Initiating NSE at 01:59
Completed NSE at 01:59, 0.00s elapsed
Initiating NSE at 01:59
Completed NSE at 01:59, 0.00s elapsed
Nmap scan report for 192.168.101.254
Host is up (0.0000030s latency).
Not shown: 65534 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (Ubuntu Linux; protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/./share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.22 seconds
Raw packets sent: 65535 (2.884MB) | Rcvd: 131071 (5.505MB)
root@ubuntu:~#

```

Figura 3.22 Comando nmap, revisión de puertos abiertos

## Seguridad preventiva

## IDS Suricata

Para instalar el IDS Suricata, como se observa en la Figura 3.23, primero se instala el repositorio Open Information Security Foundation's (OSIF), a continuación el programa suricata para de manera posterior habilitar el servicio "enable" y se detenerlo "stop" con los comandos respectivos [11], [12], [44], [65].

```
root@ubuntu:~# add-apt-repository ppa:oisf/suricata-stable
root@ubuntu:~# apt install suricata
root@ubuntu:~# systemctl enable suricata.service
root@ubuntu:~# systemctl stop suricata.service
```

**Figura 3.23** Instalación de suricata

A continuación, para habilitar el ID de Community low (en caso de uso de herramientas externas como zee, elasticserach), en la ubicación "/etc/suricata/suricata.yaml" se debe ir a la línea 127 y editar para que se muestre como "community-id: true" (Figura 3.24) [44], [65].

```
# Community Flow ID
# Adds a 'community_id' field to EVE records. These are meant to give
# records a predictable flow ID that can be used to match records to
# output of other tools such as Zeek (Bro).
#
# Takes a 'seed' that needs to be same across sensors and tools
# to make the id less predictable.
# enable/disable the community id feature.
community-id: true
# Seed value for the ID output. Valid values are 0-65535.
community-id-seed: 0
```

**Figura 3.24** Habilitación de importación de rulesets

Para mostrar las interfaces a usar se ejecuta el comando "ip -p -j route show default", con esta información, en la ubicación "/etc/suricata/" se edita el archivo "suricata.yaml", aquí se modifica la sentencia "af-packet" y en la línea 614 "interface: eth0" como "ens3", la otra interface "ens4", se añade al final antes de la línea "-interface: default" (Figura 3.25) [44], [65].

```
# Put default values here. These will be used for an interface that is not
# in the list above.
- interface: ens4
  cluster-id: 98

- interface: default
  #threads: auto
  #use-mmap: no
```

**Figura 3.25** Asignación de interfaces



Para configurar las reglas de suricata sin reiniciar el servicio, se debe activar el “Live rule reloading”, en este archivo “.yaml” hay que añadir las líneas “detect-engine:” y “-rule-reload: true”. Para exportar las rulesets externas se usa el comando “suricata-update” y se obtiene la Figura 3.26. En caso de querer usar otras rulesets, se puede revisar los proveedores de rulesets disponibles con el comando “suricata-update list-sources” y habilitarlas con el comando “suricata-update enable-source <nombre de ruleset>” y actualizar de nuevo [44], [65].

```
8/1/2024 -- 08:02:56 - <Info> -- Loading distribution rule file /usr/share/suricata/rules/stream-events.rules
8/1/2024 -- 08:02:56 - <Info> -- Loading distribution rule file /usr/share/suricata/rules/tls-events.rules
8/1/2024 -- 08:02:56 - <Info> -- Loaded 381 rules.
8/1/2024 -- 08:02:56 - <Info> -- Disabled 14 rules.
8/1/2024 -- 08:02:56 - <Info> -- Enabled 0 rules.
8/1/2024 -- 08:02:56 - <Info> -- Modified 0 rules.
8/1/2024 -- 08:02:56 - <Info> -- Dropped 0 rules.
8/1/2024 -- 08:02:56 - <Info> -- Enabled 0 rules for flowbit dependencies.
8/1/2024 -- 08:02:56 - <Info> -- Creating directory /var/lib/suricata/rules.
8/1/2024 -- 08:02:56 - <Info> -- Backing up current rules.
8/1/2024 -- 08:02:56 - <Info> -- Writing rules to /var/lib/suricata/rules/suricata.rules: total: 381; enabled: 321; added: 381; removed 0; modified: 0
8/1/2024 -- 08:02:56 - <Info> -- Writing /var/lib/suricata/rules/classification.config
8/1/2024 -- 08:02:56 - <Info> -- Testing with suricata -T.
8/1/2024 -- 08:02:56 - <Info> -- Done.
root@ubuntu:/#
```

**Figura 3.26** Actualización de rulesets de suricata

Para correr suricata se ejecuta el comando “systemctl start suricata.service” y la orden “systemctl status suricata.service” para observar el status. La revisión de eventos se realiza en el log con el comando “grep “ID” /var/log/suricata/fast.log”; para examinar el “eve.log” con el comando “apt install jq” una sola vez y luego “jq 'select(.alert.signature\_id=="ID")' /var/log/suricata/eve.json”, al ser el ID, el de la alerta de intrusión [44], [65].

### Tiempo NTP

Con el objetivo de tener una sincronización adecuada, se configuró el servicio de tiempo NTP, para esto se lo hizo mediante la ejecución del comando “dpkg-reconfigure tzdata”, siguiendo las instrucciones de configuración, al ejecutar “timedatectl status” se obtiene la Figura 3.27; si se desea personalizar el servidor NTP para un servicio más exacto, se lo hace mediante el comando “sudo apt-get install ntp”, a continuación, para elegir el servidor de tiempo más cercano, se crea un archivo en la ubicación “/etc/cron.daily/ntpdate” que contenga las líneas de la Figura 3.28 [12], [66].

```
Current default time zone: 'America/Bogota'
Local time is now:      Mon Jan  8 03:40:12 -05 2024.
Universal Time is now:  Mon Jan  8 08:40:12 UTC 2024.

root@ubuntu:/# timedatectl status
          Local time: Mon 2024-01-08 03:40:20 -05
          Universal time: Mon 2024-01-08 08:40:20 UTC
                RTC time: Mon 2024-01-08 08:40:20
                Time zone: America/Bogota (-05, -0500)
System clock synchronized: yes
              NTP service: active
          RTC in local TZ: no
root@ubuntu:/#
```

**Figura 3.27** Estadística de tiempo

```
#!/bin/sh
ntpdate ntp.ubuntu.com pool.ntp.org
~
~
~
~
:wq
```

**Figura 3.28** Archivo ntpdate

Al final, para hacerlo ejecutable, se ejecuta la orden “sudo chmod 755 /etc/cron.daily/ntpdate” y para indicar los servidores más cercanos, con el comando “ntpdate ntp.ubuntu.com pool.ntp.org” [12], [66].

## Hardening de Servicios

### Firewall:

El firewall de IP tables se configuró con el módulo Net-ISP-Balance, este comienza con todos los puertos cerrados y de manera añadida, como se verá en la Figura 3.29, este configura servicio anti-spoofing, anti-flooding, e informes; para terminar con la orden de abrir solo el puerto 22 para SSH, mediante el comando “iptables -A INPUT -p tcp --dport 22 -j ACCEPT” [12], [50], [61].

```

1959 $self->sh(<<END);
1960 iptables -P INPUT DROP
1961 iptables -P OUTPUT DROP
1962 iptables -P FORWARD DROP
1963
1964 iptables -N REJECTPERM
1965 iptables -A REJECTPERM -j LOG -m limit --limit 1/minute --log-level 4 --log-prefix "REJECTED: "
1966 iptables -A REJECTPERM -j REJECT --reject-with icmp-net-unreachable
1967
1968 iptables -N DROPGEN
1969 iptables -A DROPGEN -j LOG -m limit --limit 1/minute --log-level 4 --log-prefix "GENERAL: "
1970 iptables -A DROPGEN -j DROP
1971
1972 iptables -N DROPINVAL
1973 iptables -A DROPINVAL -j LOG -m limit --limit 1/minute --log-level 4 --log-prefix "INVALID: "
1974 iptables -A DROPINVAL -j DROP
1975
1976 iptables -N DROPPERM
1977 iptables -A DROPPERM -j LOG -m limit --limit 1/minute --log-level 4 --log-prefix "ACCESS-DENIED: "
1978 iptables -A DROPPERM -j DROP
1979
1980 iptables -N DROPSPOOF
1981 iptables -A DROPSPOOF -j LOG -m limit --limit 1/minute --log-level 4 --log-prefix "DROP-SPOOF: "
1982 iptables -A DROPSPOOF -j DROP
1983
1984 iptables -N DROPFLOOD
1985 iptables -A DROPFLOOD -m limit --limit 1/minute -j LOG --log-level 4 --log-prefix "DROP-FLOOD: "
1986 iptables -A DROPFLOOD -j DROP
1987
1988 iptables -N DEBUG
1989 iptables -A DEBUG -j LOG --log-level 3 --log-prefix "DEBUG: "
1990 -END

```

**Figura 3.29** Configuración del *firewall*

### Servidor SSH

Para encriptar la comunicación del servicio SSH se recomienda mínimo 64 bits de entropía, sin embargo, se usó Canonical recomienda un mínimo de  $rsa=3072$  bits, para crear las claves, por esto se consideró entonces 4096 bits; para generar la clave privada en la máquina cliente, se ejecuta el comando de la Figura 3.30 [12], [67].

```

root@ubuntu22-desktop:~# ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:pP8Y883zcDzcoYNlVtiiiLn4kQK0eTxnrGOVDDS8ibME root@ubuntu22-desktop
The key's randomart image is:
+----[RSA 4096]-----+
  .   +   o
  . . o *   o o
  + o O o . o
  . o B S . + .
    + B ...=0...
    * =.00.0= .
  . o.*.+o..
    ..+ 00.
+-----[SHA256]-----+
root@ubuntu22-desktop:~#

```

**Figura 3.30** Generación de clave rsa

A continuación, se copia la clave pública para servidor en la ubicación “~/.ssh/authorized\_keys” y como se observa en la Figura 3.31, con los comandos “ssh-copy-id lbadmín@192.168.101.254” y “ssh-copy-id carlos@192.168.101.254”, se copia una clave por cada usuario [12], [67].

```

root@ubuntu22-desktop:~# ssh-copy-id carlos@192.168.101.254
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
carlos@192.168.101.254's password:
Number of key(s) added: 1

```

**Figura 3.31** Copia de la clave pública

Para finalizar, solo los usuarios autenticados deben tener acceso a la ubicación “~/.ssh/authorized\_keys”, por lo tanto, se cambia los permisos de esta mediante la ejecución del comando “chmod 600 ~/.ssh/authorized\_keys” [12], [67].

## Seguridad operacional

### Manejo de Cuentas de usuario:

La primera medida de seguridad de usuarios es remover los permisos de lectura entre usuarios, con el comando “vi /etc/adduser.conf”, se modifica la línea para que quede como se puede comprobar en la Figura 3.32 [12], [68].

```

# If DIR_MODE is set, directories will be created with the specified
# mode. Otherwise the default mode 0755 will be used.
DIR_MODE=0750

```

**Figura 3.32** Permisos de cuentas

Para comprobar permisos removidos se ejecuta el comando de la Figura 3.33 [12], [68].

```
root@ubuntu:~# ls -ld /home/carlos
drwxr-x--- 4 carlos carlos 4096 Jan 10 19:45 /home/carlos
root@ubuntu:~#
```

**Figura 3.33** Comprobación de permisos

Para crear usuarios y grupos, se ejecutan en orden, los comandos “adduser carlos”, “addgroup técnicos”, “adduser carlos técnicos” para añadir un usuario, un grupo y vincular un usuario a un grupo de manera respectiva; se crea también un usuario con acceso root mediante el módulo sudo con los comandos “adduser lbadadmin” y “adduser lbadadmin sudo”. Para completar esta configuración se borra la contraseña de la cuenta root con el comando “passwd -d root” y se bloquea con “passwd -l root” [12], [68].

A continuación, se especifica un tiempo de vencimiento de contraseñas como se muestra en la Figura 3.34.

```
root@ubuntu:~# sudo chage -E 08/03/2024 -m 4 -M 90 -I 20 -W 14 carlos
root@ubuntu:~# sudo chage -l carlos
Last password change           : Jan 10, 2024
Password expires               : Apr 09, 2024
Password inactive              : Apr 29, 2024
Account expires                : Aug 03, 2024
Minimum number of days between password change : 4
Maximum number of days between password change : 90
Number of days of warning before password expires : 14
root@ubuntu:~#
```

**Figura 3.34** Configuración de expiración de contraseñas

Donde (-E) expresa la fecha de expiración, (-m) indica el tiempo mínimo de duración de la contraseña, (-M) el tiempo máximo, (-I) el tiempo de inactividad después de la expiración de la contraseña y (-W) señala el período de advertencia antes de la expiración de la contraseña [12], [68].

### **Desactivar acceso de root por SSH**

Para esto, en la ubicación “/etc/ssh/sshd\_config” se debe agregar o verificar que la línea “PermitRootLogin” se encuentre en “no”, tal como indica la Figura 3.35, a continuación reiniciar el servicio con el comando “systemctl restart ssh.service” [12], [67], [69].

```
# Authentication:
#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

Figura 3.35 Desactivar el acceso root de SSH

### Limitar IP's para SSH

De la manera que plantea la Figura 3.36, para las redes permitidas y en la ubicación “/etc/hosts.allow”, se debe busca o añade “sshd :” y modificar la línea a “sshd : 192.168.101.0/24 (se usa coma y escribir seguido para colocar varias redes); para bloquear todas las otras redes, se lo hace en la ubicación “/etc/hosts.deny”, aquí se debe buscar o añadir “sshd :” y modificar la línea a “sshd : ALL” [12], [67], [69].

```
root@ubuntu:~# nano /etc/hosts.allow      root@ubuntu:~# nano /etc/hosts.deny
GNU nano 6.2
# /etc/hosts.allow: list of
# See the
#
# Example:      ALL: LOCAL @som
#              ALL: .foobar.e
#
# If you're going to protect
# daemon name. See rpcbind(8)
#
sshd : 192.168.101.0/24

# The PARANOID wildcard mat
# address.
#
# You may wish to enable th
# validate looked up hostna
# versions of Debian this h
# ALL: PARANOID
sshd : ALL
```

Figura 3.36 Limitar IP's de acceso SSH

### Desactivar fingerprints de antiguos accesos SSH

En la ubicación “/home/user/.ssh/” borrar las claves guardadas en el cliente, esto con los comandos “ls -l “/home/user/.ssh/”; “rm /home/user/.ssh/known\_hosts”; “rm /home/user/.ssh/known\_hosts.old”. Para desactivar sesiones anteriores que podrían seguir activas, en el cliente ejecutar “who | grep <user>” y “pkill -f pts/###”. Al final para desactivar acceso por contraseña en el cliente, en la ubicación “/etc/ssh/ssh\_config”, se busca editar la línea para que quede de manera cómo se puede ver en la Figura 3.37 [12], [67], [69].

```
Include /etc/ssh/ssh_config.d/*.conf

Host *
# ForwardAgent no
# ForwardX11 no
# ForwardX11Trusted yes
PasswordAuthentication no
# HostbasedAuthentication no
# GSSAPIAuthentication no
# GSSAPIDelegateCredentials no
```

Figura 3.37 Desactivar la autenticación por contraseña

### Monitoreo de eventos local o mail:

Para tener un sistema centralizado de monitoreo, se instala el programa Logwatch, primero mediante la ejecución del comando “Apt update && apt upgrade && apt autoremove”, para tener las librerías preparadas; a continuación, con el comando “apt install postfix” se configura el servicio de correo para logwatch de manera “local”; con el comando “apt install logwatch” se instala el servicio y mediante el archivo ubicado en “/etc/logwatch/conf/logwatch.conf”, se configura. Para obtener el informe de eventos de logs se ejecuta el comando “logwatch --output stdout --format text --detail Low --range - 7 days” (Figura 3.38) [12], [70].

```
root@ubuntu:~# logwatch --output stdout --format text --detail Low --range -1 days
##### Logwatch 7.5.6 (07/23/21) #####
Processing Initiated: Mon Jan 29 05:14:49 2024
Date Range Processed: -1
                    ( 2024-Jan-29 )
                    Period is day.
Detail Level of Output: 0
Type of Output/Format: stdout / text
Logfiles for Host: ubuntu
#####

----- Kernel Begin -----

WARNING: Kernel Errors Present
Buffer I/O error on dev fd0, logi ...: 6 Time(s)
WARNING: See https://kernel.org/doc/html/latest/ ...: 3 Time(s)
blk_update_request: I/O error, dev fd0, sector ...: 12 Time(s)

----- Kernel End -----

----- pam_unix Begin -----

sudo:
Sessions Opened:
```

Figura 3.38 Muestra en pantalla del monitor logwatch

Donde “-output” especifica el tipo de salida del informe, este campo puede reemplazarse para servir la información por correo; stdout indica la opción de salida por defecto de la presentación en bash; “-format text” especifica que se muestre en texto

plano; "--detail Low" indica el nivel de detalle de la presentación que puede ser por tres niveles o por numeración; "--range -7 days" indica el reporte de tiempo atrás del que se desea el informe en este caso "7" días [12], [70].

### 3.4 Implementación del prototipo de balanceador de carga de salida a Internet

La implementación del prototipo se la realizó de manera simulada, mediante el programa simulador de redes EVE-ng, para esta sección se tomó en cuenta que, una vez realizada la instalación, el servidor provee de Internet a una red LAN, a la cual se puede añadir cualquier tipo de máquina cliente.

#### Instalación del servidor balanceador de carga

Se colocó el balanceador de carga de manera que sirva a la LAN, tal como se observa en la Figura 3.39, el balanceador de carga, de nombre Linux, se coloca al inicio de la red local a servir, esta con dirección de red 192.168.144.0/24; en donde, al lado izquierdo del balanceador se encuentra la salida de Internet y al lado derecho de este la red cliente, en esta se colocaron, una PC virtual, una máquina con SO Ubuntu Desktop 22.0 y Una máquina virtual con SO Windows 10.

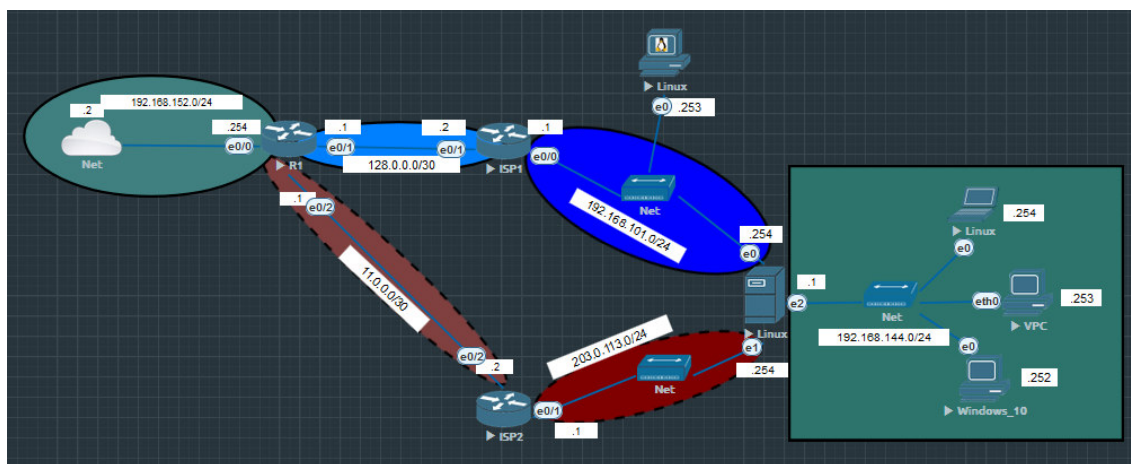
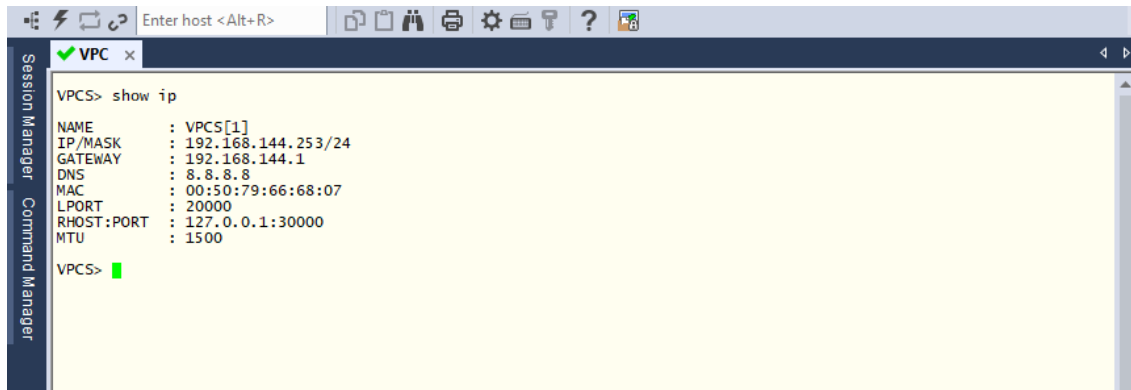


Figura 3.39 Red implementada

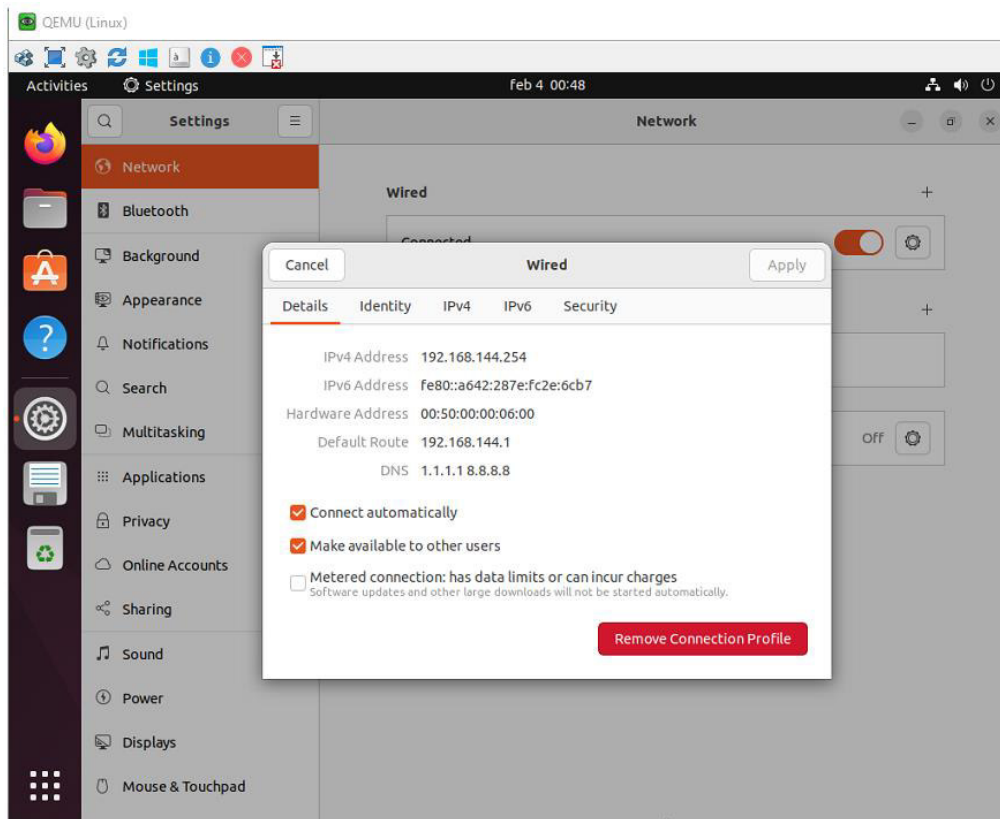
Primero, como se observa en la Figura 3.40, la PC virtual de nombre "VPC", se configuró con la dirección IP de *host* 192.168.144.253, la cual actúa como de terminal final de bajos requerimientos e interfaz de terminal de comandos, exclusiva para realizar pruebas específicas o comprobación de funcionamiento.





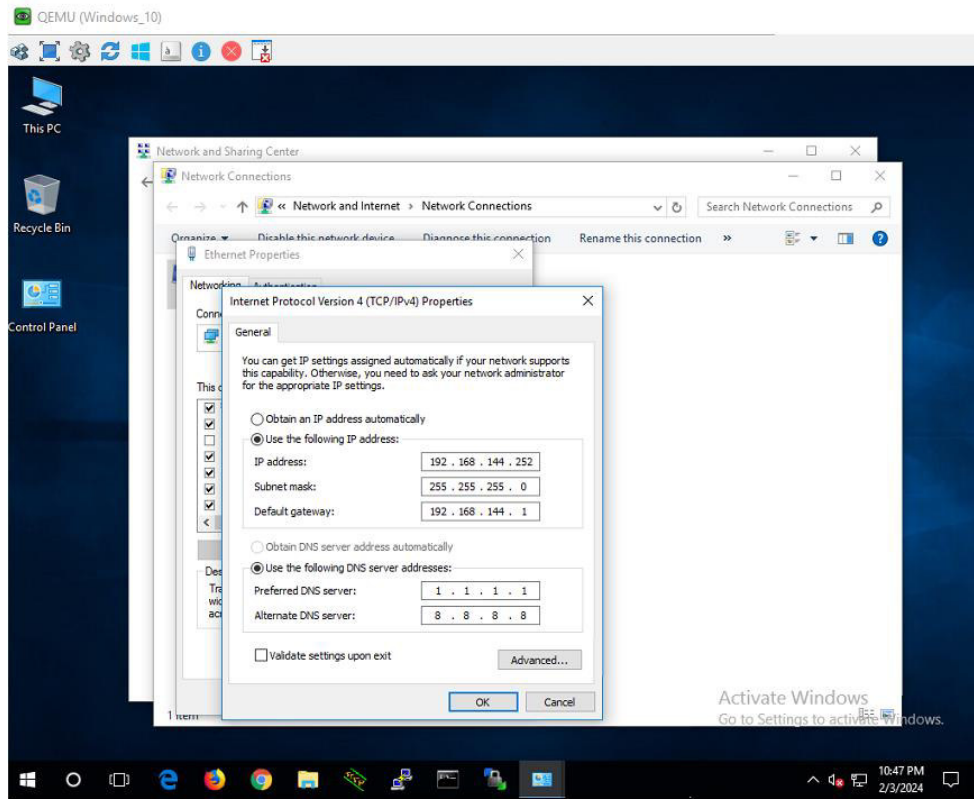
**Figura 3.40** Configuración de red (terminal final VPC)

La segunda PC de nombre "Linux", se configuró con la dirección IP de *host* 192.168.144.254, esta sirve de terminal final de pruebas completas, esta usa el SO Ubuntu 22.04 desktop (Figura 3.41).



**Figura 3.41** Configuración de red (terminal final Ubuntu desktop 22.04)

La tercera PC de nombre "Windows\_10", se configuró con la dirección IP de *host* 192.168.144.254, esta también sirve de terminal final de pruebas completas y usa el SO Windows 10 (Figura 3.42).



**Figura 3.42** Configuración de red (terminal final Windows 10)

Para finalizar la implementación, en el balanceador de carga se ejecuta el comando “load\_balance.pl”, el cual arroja los avisos de la Figura 3.43; el aviso “FIB table does not exist Flush terminated” significa que al iniciar, la secuencia de configuración del balanceador trata de vaciar la tabla de base de información de envío, la cual sirve para establecer órdenes de enrutamiento, como al iniciar no existe ninguna, no se realiza el vaciado (flush) y presenta la alerta mencionada. En segundo lugar, el aviso “starting foolsm link status monitoring daemon”, indica el inicio del daemon “foolsm”, el cual sirve para el monitoreo en tiempo real del estado de las conexiones de Internet para el módulo de *failover* [43].

```

root@ubuntu:~# load_balance.pl
Error: ipv4: FIB table does not exist.
Flush terminated
Error: ipv4: FIB table does not exist.
Flush terminated
Starting foolsm link status monitoring daemon
root@ubuntu:~#

```

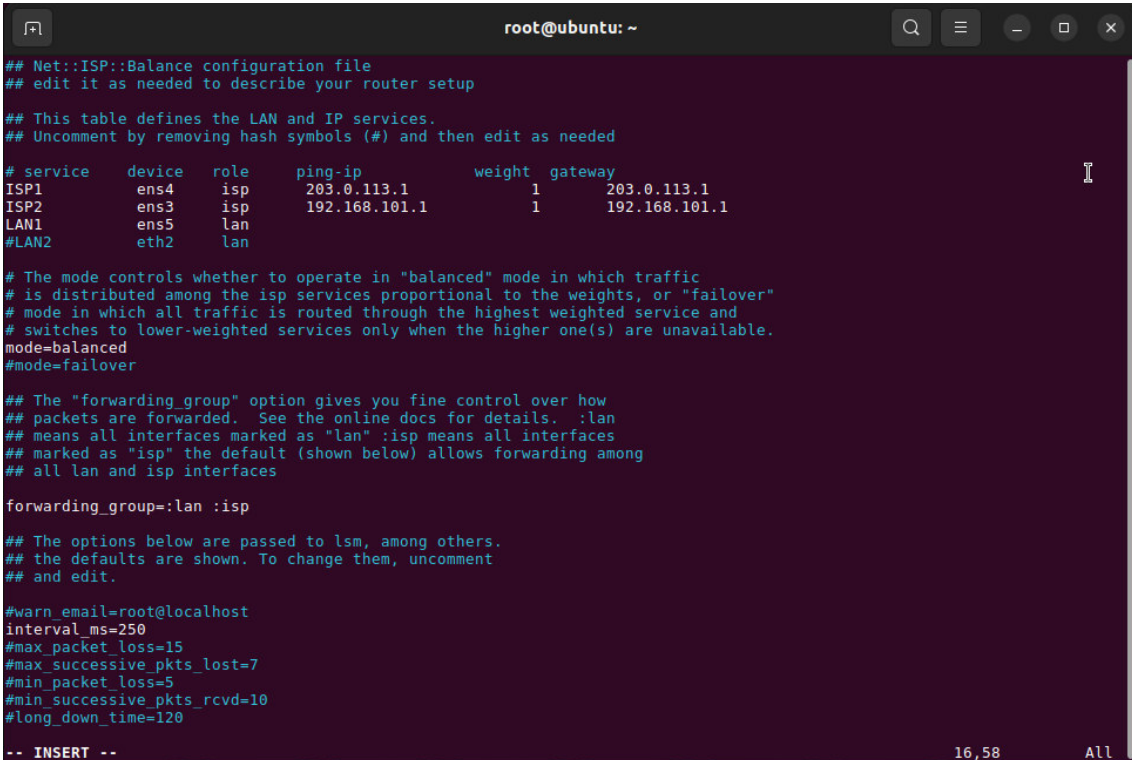
**Figura 3.43** Inicio del balanceador de carga

### 3.5 Realización exhaustiva de pruebas de funcionamiento

En esta sección se explica la metodología de prueba utilizada para verificar el correcto funcionamiento del balanceador de carga, esta comprende de dos secciones, la primera, se encarga de verificar el funcionamiento de manera general, con los sistemas operativos completos y carga ejecutada en tiempo real, la segunda, se encarga de verificar el funcionamiento de manera precisa mediante la VPC y el programa Wireshark.

#### Pruebas de trabajo del balanceador de carga

Para comprobar el correcto funcionamiento del balanceador de carga se procedió a realizar pruebas de salida de tráfico de Internet, de esta manera se planteó la siguiente prueba: con el balanceador de carga funcionando en configuración por defecto, es decir en relación de tráfico 1:1, ping de detección de detección de enlace 250ms, tal como se puede ver en la Figura 3.44.



```
## Net::ISP::Balance configuration file
## edit it as needed to describe your router setup

## This table defines the LAN and IP services.
## Uncomment by removing hash symbols (#) and then edit as needed

# service    device  role    ping-ip    weight  gateway
ISP1         ens4    isp     203.0.113.1  1       203.0.113.1
ISP2         ens3    isp     192.168.101.1  1       192.168.101.1
LAN1         ens5    lan
#LAN2        eth2    lan

# The mode controls whether to operate in "balanced" mode in which traffic
# is distributed among the isp services proportional to the weights, or "failover"
# mode in which all traffic is routed through the highest weighted service and
# switches to lower-weighted services only when the higher one(s) are unavailable.
mode=balanced
#mode=failover

## The "forwarding_group" option gives you fine control over how
## packets are forwarded. See the online docs for details. :lan
## means all interfaces marked as "lan" :isp means all interfaces
## marked as "isp" the default (shown below) allows forwarding among
## all lan and isp interfaces

forwarding_group=:lan :isp

## The options below are passed to lsm, among others.
## the defaults are shown. To change them, uncomment
## and edit.

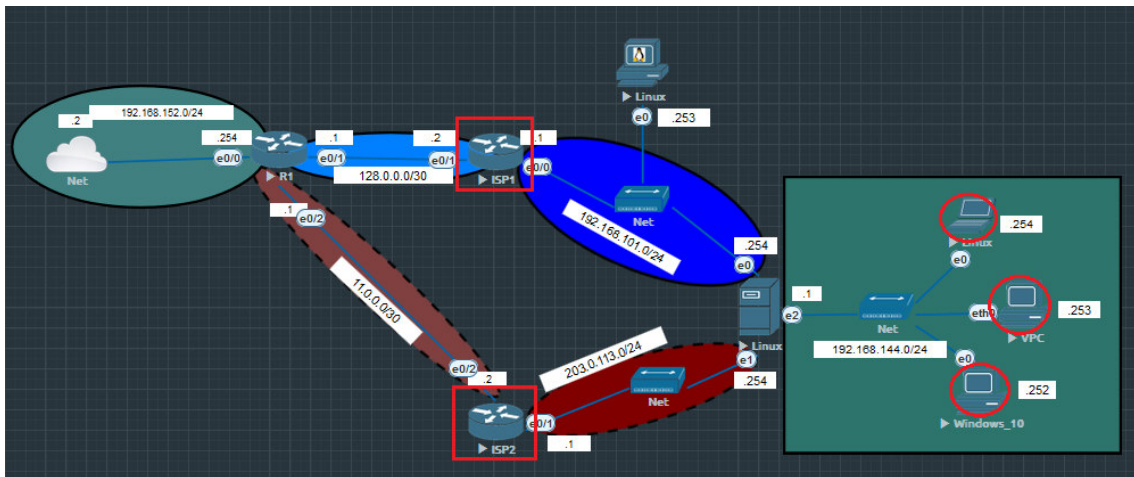
#warn_email=root@localhost
interval_ms=250
#max_packet_loss=15
#max_successive_pkts_lost=7
#min_packet_loss=5
#min_successive_pkts_rcvd=10
#long_down_time=120

-- INSERT --
```

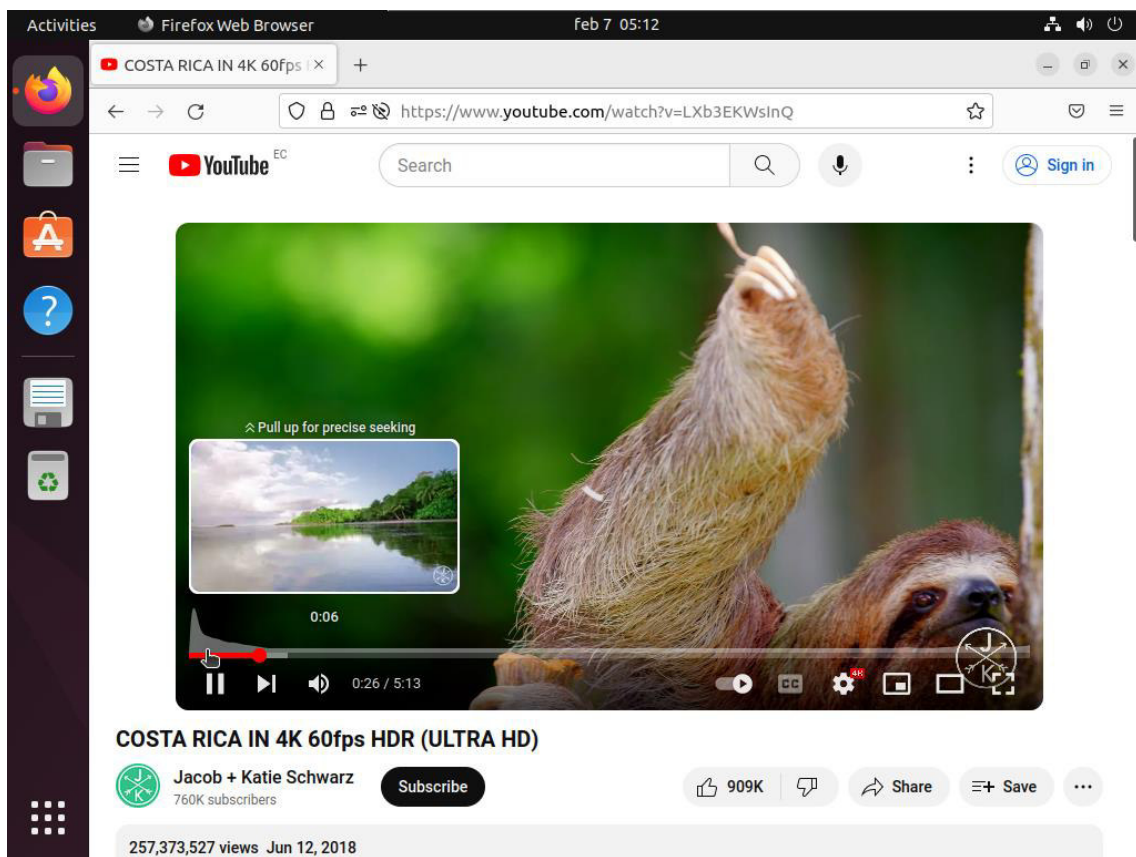
Figura 3.44 Balanceador de carga, configuración por defecto

Al aplicar la configuración con el comando “load\_balance.pl”, se procede a encender los terminales clientes. Con los routers ISP1 e ISP2 encendidos y los terminales clientes Linux y Windows en funcionamiento, como se puede ver en la Figura 3.45; primero, una vez ingresado al escritorio cliente del terminal cliente Linux, se abre el navegador Firefox, con este en ejecución se ingresa a Youtube, se elige un video de preferencia extenso, con calidad alta; a continuación, se procede a observar si el video carga y

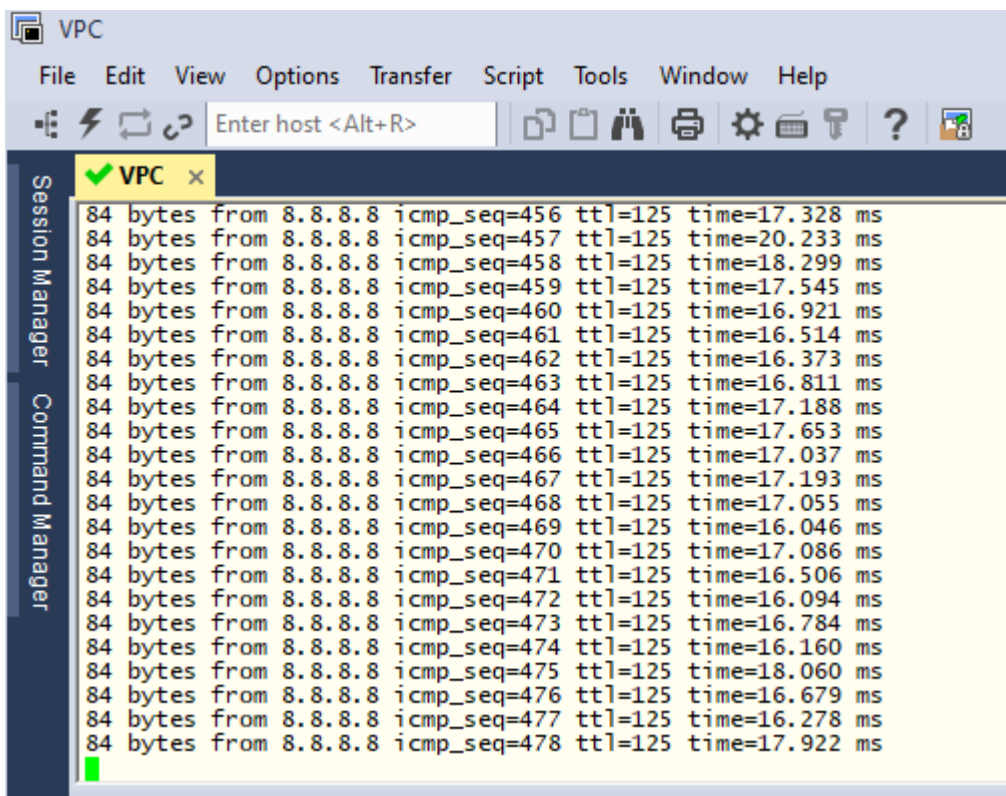
reproduce el contenido (Figura 3.45), se observa también en la PC virtual, mediante ejecutar interrupciones en el ping indefinido (Figura 3.47).



**Figura 3.45** Prueba con ambos ISP

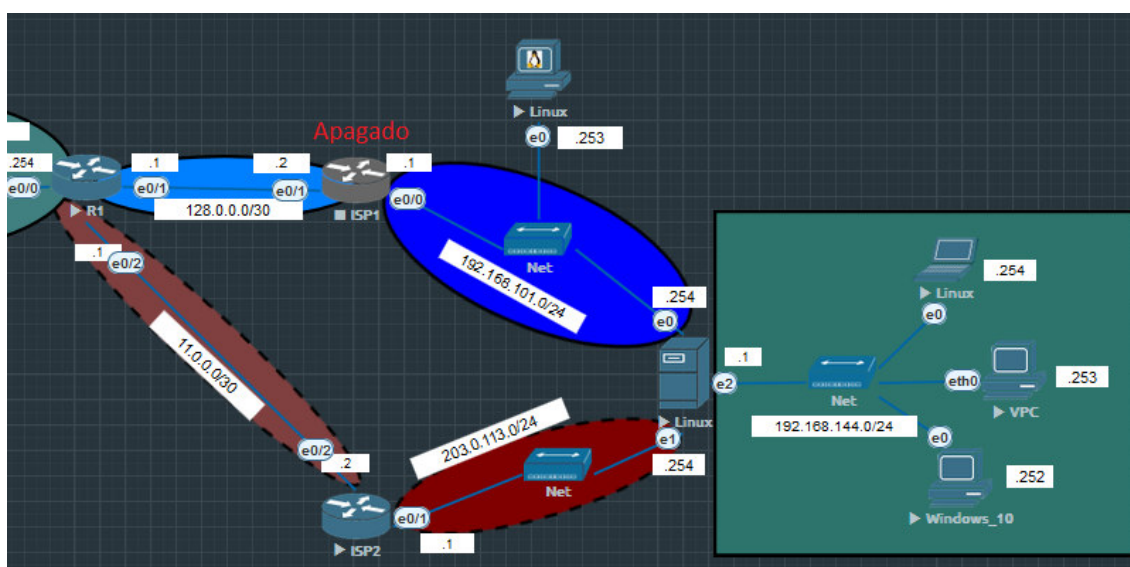


**Figura 3.46** Observación de carga de video (2 ISP)

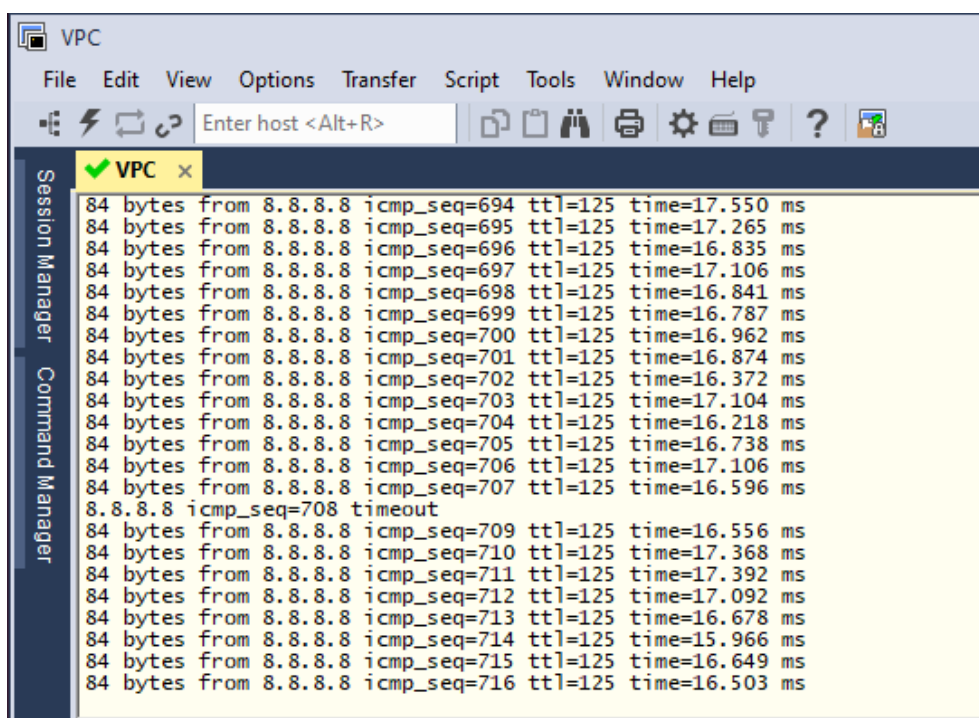


**Figura 3.47** Ping indefinido (sin interrupción)

A continuación, como el video si se reproduce y en el ping indefinido no se observa interrupción, se procede a comprobar si existe tráfico mediante el corte de una de las líneas al apagar el router “ISP1” (Figura 3.48); de manera inmediata se revisa si existe interrupción en el ping indefinido Figura 3.49.

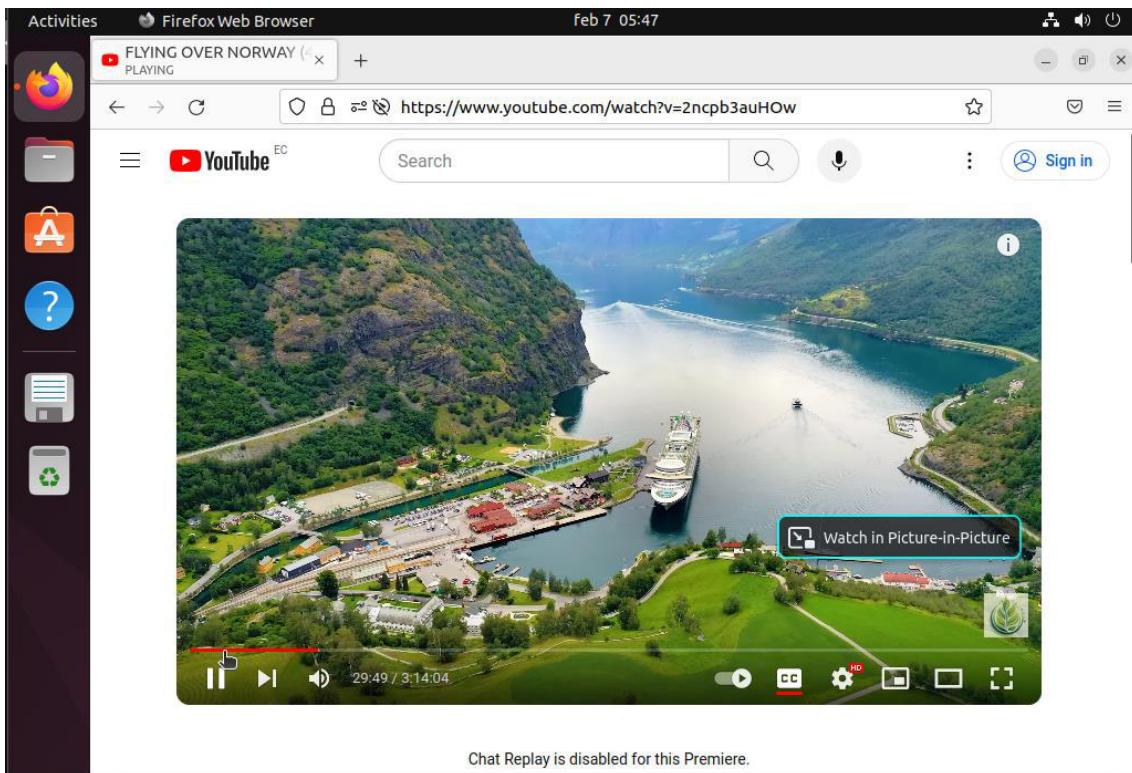


**Figura 3.48** Router ISP1 apagado



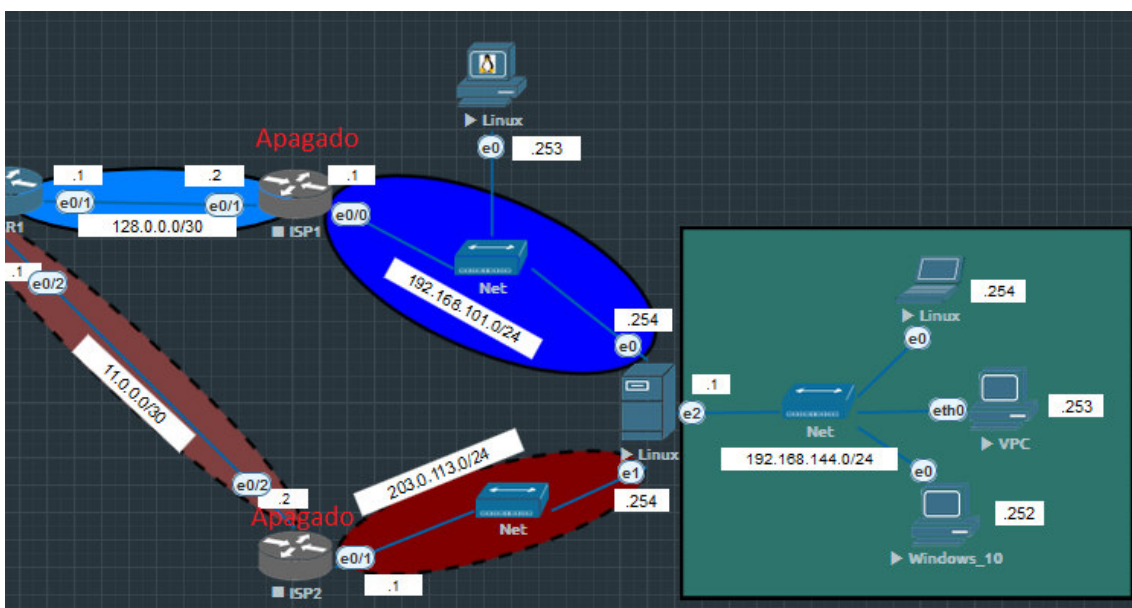
**Figura 3.49** Interrupción del ping indefinido (breve)

Como se observa en la Figura 3.49, existe una interrupción en la secuencia ICMP de número 708, pero, la secuencia de ping retorna a su flujo regular sin cortes, lo que indica que el sistema de failover funcionó y redirigió el tráfico a través de la línea sobrante. A continuación, se observa que el video reproduzca contenido; como se comprueba en la Figura 3.50, el video continua su reproducción.

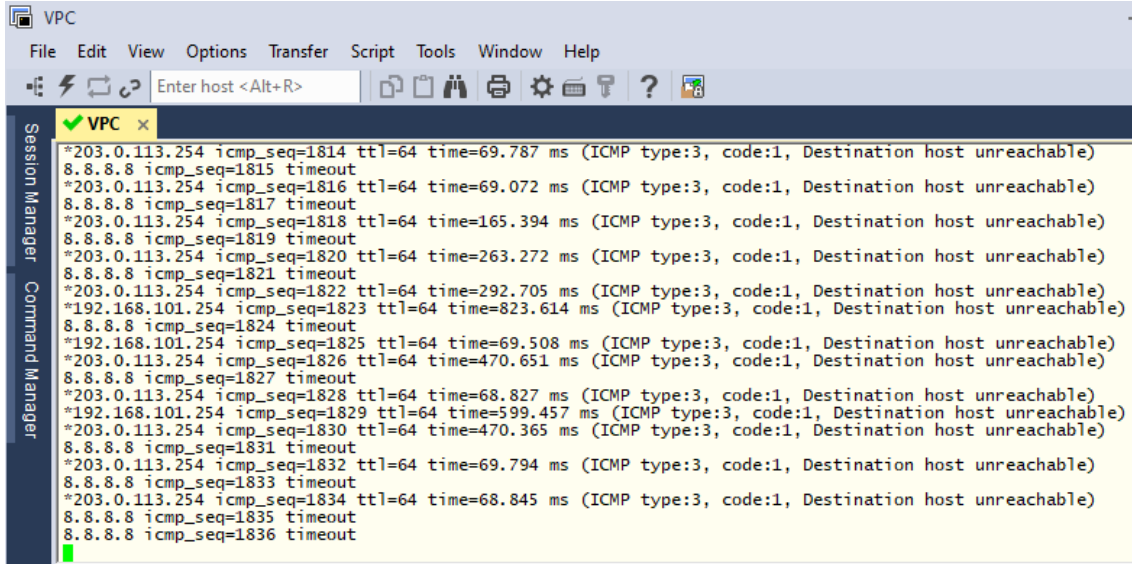


**Figura 3.50** Observación de carga de video (solo ISP2 encendido)

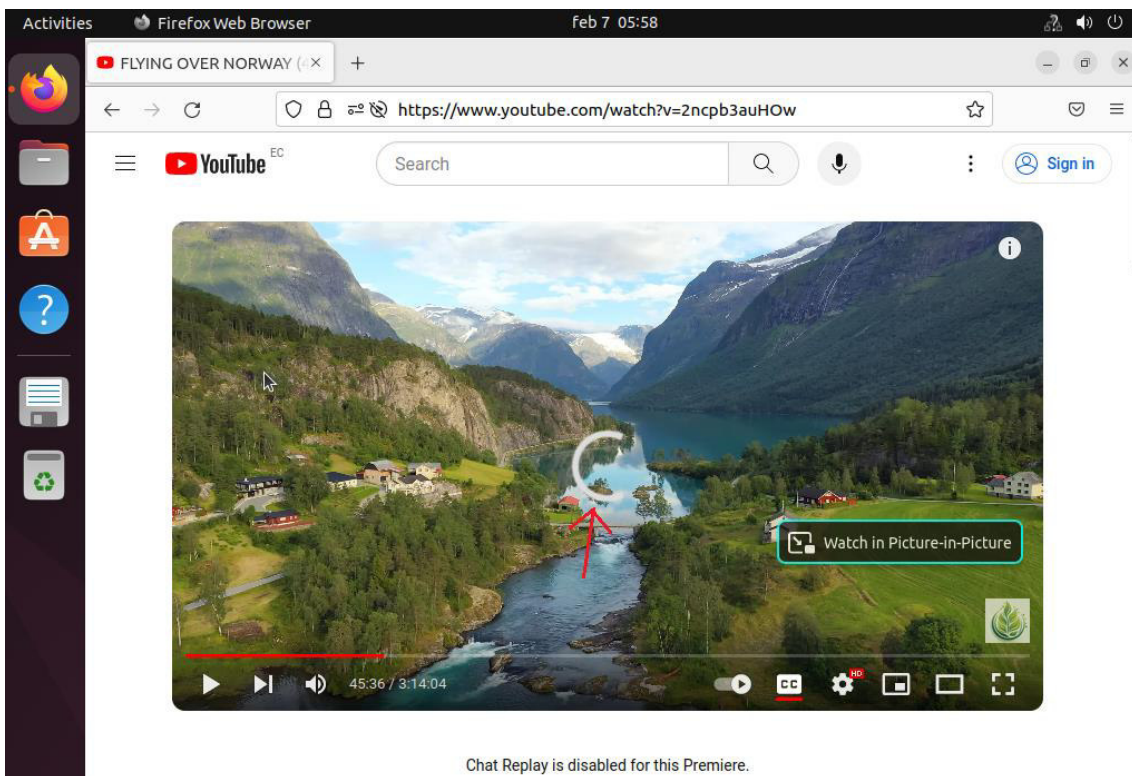
Luego se procede a realizar el corte de la única línea encendida (Figura 3.51), y se comprueba de nuevo el flujo del ping indefinido, y la reproducción del video (Figura 3.52 y Figura 3.53).



**Figura 3.51** Routers ISP1 e ISP2 apagados



**Figura 3.52** Interrupción del ping indefinido (permanente)

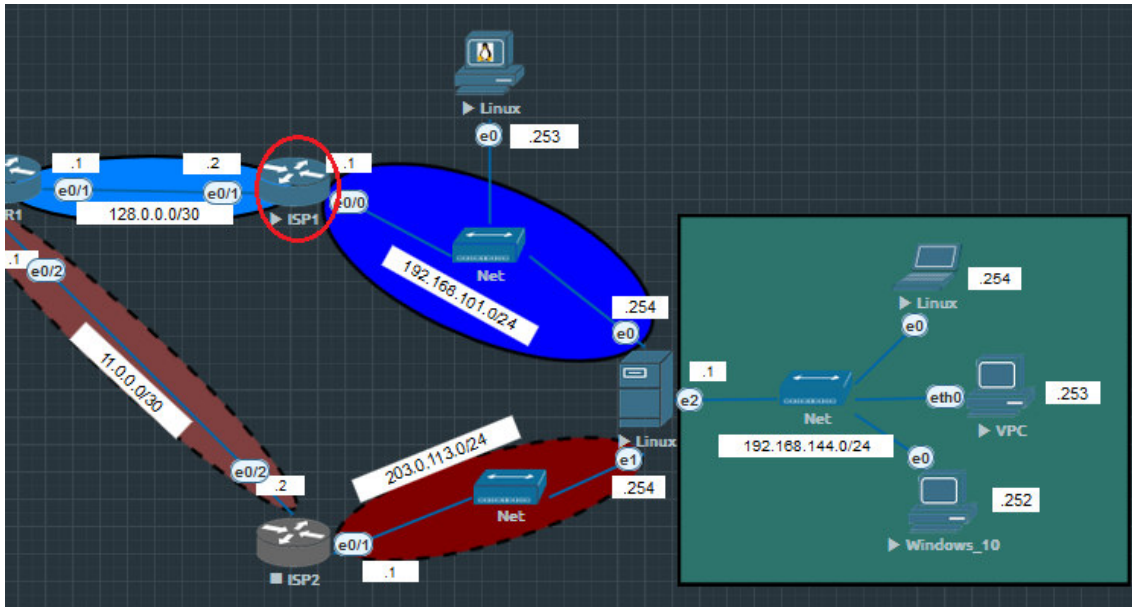


**Figura 3.53** Observación de carga de video (Ningún ISP encendido)

Como se observa arriba, en la Figura 3.51, el ping indefinido se ha detenido y no alcanza ningún host y como se observa en la Figura 3.52, el video de Youtube tampoco carga, ambos indicadores de que el tráfico de red ha sido detenido por la inactividad de ambos routers ISP.

A continuación, se enciende solo el ISP1 (Figura 3.54).





**Figura 3.54** Routers ISP2 encendido

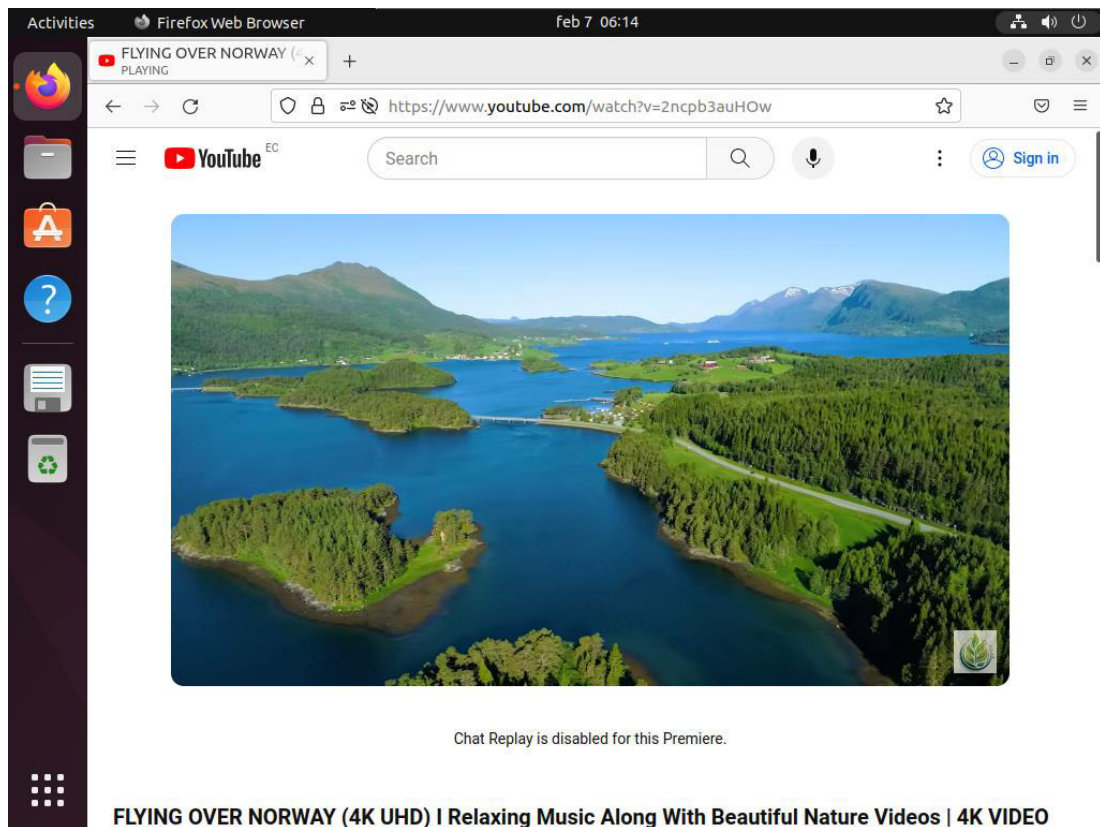
Para comprobar que exista tráfico de red solo por esta línea, de manera inmediata se analiza el ping indefinido de VPC. En la Figura 3.55, el ping indefinido se restaura luego de entrar a funcionar la detección de failover en una sola línea. A continuación, se comprueba también la carga del video, que, como se ve en la Figura 3.56, también ha vuelto a su funcionamiento normal.

```

VPC
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
Session Manager
Command Manager
VPC x
84 bytes from 8.8.8.8 icmp_seq=2211 ttl=125 time=16.826 ms
84 bytes from 8.8.8.8 icmp_seq=2212 ttl=125 time=17.838 ms
8.8.8.8 icmp_seq=2213 timeout
84 bytes from 8.8.8.8 icmp_seq=2214 ttl=125 time=16.632 ms
84 bytes from 8.8.8.8 icmp_seq=2215 ttl=125 time=16.683 ms
*203.0.113.254 icmp_seq=2216 ttl=64 time=159.604 ms (ICMP type:3, code:1, Destination host unreachable)
84 bytes from 8.8.8.8 icmp_seq=2217 ttl=125 time=16.263 ms
8.8.8.8 icmp_seq=2218 timeout
8.8.8.8 icmp_seq=2219 timeout
84 bytes from 8.8.8.8 icmp_seq=2220 ttl=125 time=16.750 ms
84 bytes from 8.8.8.8 icmp_seq=2221 ttl=125 time=16.628 ms
84 bytes from 8.8.8.8 icmp_seq=2222 ttl=125 time=17.152 ms
*203.0.113.254 icmp_seq=2223 ttl=64 time=395.270 ms (ICMP type:3, code:1, Destination host unreachable)
8.8.8.8 icmp_seq=2224 timeout
84 bytes from 8.8.8.8 icmp_seq=2225 ttl=125 time=17.038 ms
84 bytes from 8.8.8.8 icmp_seq=2226 ttl=125 time=17.092 ms
84 bytes from 8.8.8.8 icmp_seq=2227 ttl=125 time=16.620 ms
84 bytes from 8.8.8.8 icmp_seq=2228 ttl=125 time=16.720 ms
84 bytes from 8.8.8.8 icmp_seq=2229 ttl=125 time=16.760 ms
84 bytes from 8.8.8.8 icmp_seq=2230 ttl=125 time=16.155 ms
84 bytes from 8.8.8.8 icmp_seq=2231 ttl=125 time=16.874 ms
84 bytes from 8.8.8.8 icmp_seq=2232 ttl=125 time=16.617 ms
84 bytes from 8.8.8.8 icmp_seq=2233 ttl=125 time=15.893 ms

```

**Figura 3.55** Restauración del ping indefinido (solo ISP2)



**Figura 3.56** Observación de carga de video (Solo ISP2)

Como prueba de trabajo final, se encienden ambos routers, igual que en el escenario de la Figura 3.45, para comprobar que exista tráfico de Internet en ambas líneas se ejecuta Wireshark (que viene en la instalación por defecto de EVE-ng) para monitorear ambas líneas, se verifica que exista tráfico UDP en ambas líneas, tal como se ve en la Figura 3.57. Una vez finalizado el proceso de esta sección, se repite para el cliente de Windows. Así se comprueba que el balanceador de carga se encuentra en estado operativo.

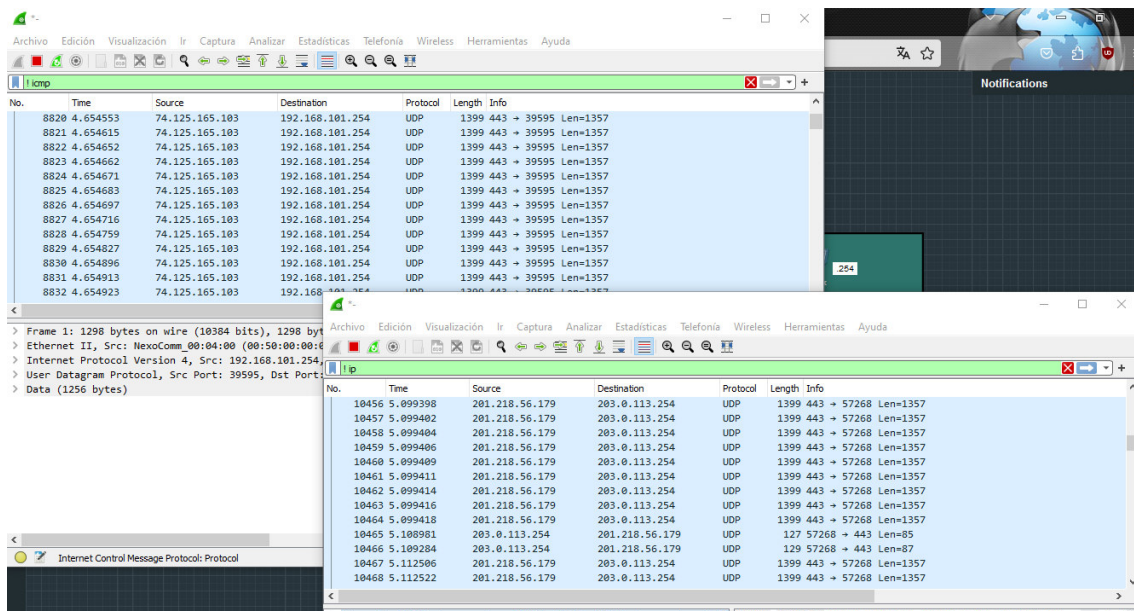


Figura 3.57 Observación de la operación del balanceador de carga en Wireshark

### Mediciones de efectividad del balanceador de carga

Para la realización de las pruebas se eligió la instalación del programa de monitoreo de TCPdump, el cual permite hacer capturas de tráfico de red presente en las interfaces del sistema, para la instalación se ejecuta el comando “apt install tcpdump” y una vez instalado, se puede ejecutar el comando “tcpdump -G 30 -W 1 -w /home/<user>/capturas/ens3.pcap -i ens3” con la interjección “&” junto a nueva iteración de la orden (Figura 3.58), para realizar una captura del tráfico de salida de Internet, en las interfaces “ens3” y “ens4”; de esta última orden “-G” representa el tiempo en segundos, “-W” el número de capturas, “-w” la ubicación donde guardar el archivo de extensión “.pcap”, “-i” indica la interfaz [71].

```

root@ubuntu:~# apt install tcpdump
root@ubuntu:~# tcpdump -G 15 -W 1 -w /home/lbadmin/capturas/15seg_1a2_ens3.pcap -i ens3
&
tcpdump -G 15 -W 1 -w /home/lbadmin/capturas/15seg_1a2_ens4.pcap -i ens4

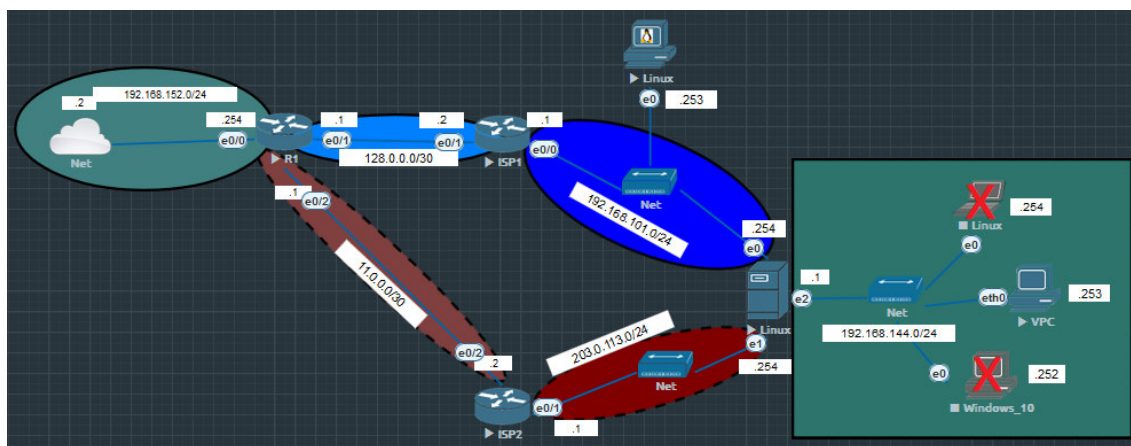
tcpdump: listening on ens4, link-type EN10MB (Ethernet), snapshot length 262144 bytes
tcpdump: listening on ens3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
Maximum file limit reached: 1
116 packets captured
123 packets received by filter
0 packets dropped by kernel
Maximum file limit reached: 1
133 packets captured
140 packets received by filter
0 packets dropped by kernel
root@ubuntu:~#
[1]+  Done                  tcpdump -G 15 -W 1 -w /home/lbadmin/capturas/15seg_1a2_
ens3.pcap -i ens3
root@ubuntu:~#

```

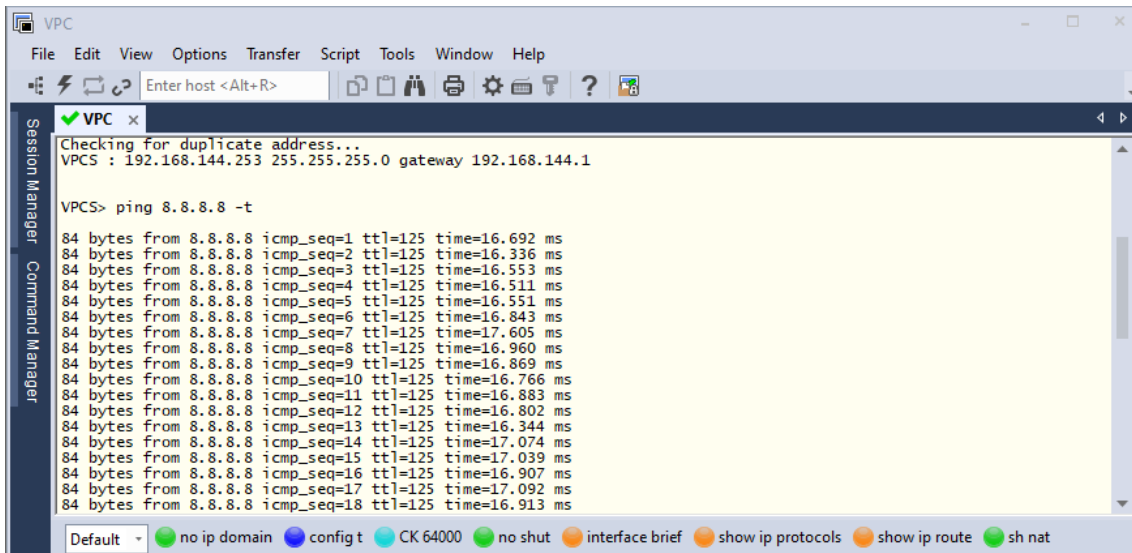
**Figura 3.58** Comandos de captura de tráfico

Para el resultado de la ejecución del comando de captura, en este ejemplo, se creó el directorio “/home/lbadmin/capturas”, para colocar todas las pruebas realizadas, se asignó una prueba de 15 segundos, la cual arroja 116 y 133 paquetes capturados en su respectiva interfaz, también 123 y 140 paquetes recibidos por el filtro en su respectiva interfaz, hace una sola captura y muestra un aviso de su terminación [71].

De esta manera se plantearon varios escenarios de prueba, primero se consideró que, para evitar la mayor cantidad de ruido (entiéndase como paquetes imprevistos en el capturador), ya sea por descargas automáticas, actualizaciones, servicios y procesos que hacen peticiones a Internet, etc., se propuso usar el PC virtual incluido en el simulador EVE-ng; este terminal posee un sistema operativo con funcionalidad reducida, enfocado a realizar pruebas de este tipo.



**Figura 3.59** Esquema de medición de paquetes



**Figura 3.60** Comando ping desde VPC

De esta manera, como se observa en Figura 3.59 se planteó que desde aquí se usará el comando “ping” para enviar y recibir peticiones ICMP a la dirección 8.8.8.8 (servidor de Google.com) tal como se observa en la Figura 3.60, las cuales se registran con el capturador tcpdump y mediante el terminal monitor Linux, se importan los archivos “.pcap” resultantes a este, para ser analizados con el programa Wireshark, a continuación se procede contar los paquetes icmp que tengan petición o respuesta a la dirección 8.8.8.8 y de manera final se comprara el volumen de datos para determinar la efectividad de la razón de tráfico asignado en la configuración, que para esta prueba es 1:1, por defecto del balanceador.

### Proceso de las pruebas estadísticas del balanceador

Como se observa en la Figura 3.61, primero se modificó el documento de configuración ubicado en “/etc/network/balance.conf”, cambiando el parámetro *weight* de ens4 al valor 1 y de ens3 al valor 1, se aplicó con el comando “load\_balance.pl”.

```
## This table defines the LAN and IP services.
## Uncomment by removing hash symbols (#) and then edit as needed

# service    device    role    ping-ip    weight    gateway
ISP1         ens4     isp     203.0.113.1    1        203.0.113.1
ISP2         ens3     isp     192.168.101.1    1        192.168.101.1
LAN1         ens5     lan
#LAN2        eth2     lan
```

**Figura 3.61** Configuración relación 1:1

A continuación, con el comando ping infinito en ejecución desde la VPC, se planteó la primera prueba con relación 1:1, tiempo de ejecución 2 minutos en “tcpdump”, con el comando de la Figura 3.62 (primera captura), una vez finalizada la captura se procedió

a importar los archivos “.pcap” al escritorio, mediante en una instancia de terminal ejecutado en este con acceso privilegiado, con los comandos que se observan en la Figura 3.62 (segunda captura). Para realizar una prueba de 5 minutos, se ejecuta “tcpdump”, con el comando de la Figura 3.62 (tercera captura), una vez finalizada la captura se procede a importar de nuevo los archivos “.pcap” al escritorio, mediante en la instancia de terminal, con los comandos que se observan en la Figura 3.62 (cuarta captura).

```

root@ubuntu:~# tcpdump -G 120 -W 1 -w /home/lbadmin/capturas/2min_1a1_ens3.pcap -i ens3 & tcpdump -G 120 -W 1 -w /home/lbadmin/capturas/2min_1a1_ens4.pcap -i ens4

root@ubuntu22-desktop:~# scp lbadmin@192.168.101.254:/home/lbadmin/capturas/2min_1a1_ens4.pcap /home/user/Desktop
2min_1a1_ens4.pcap          100% 117KB 19.9MB/s  00:00
root@ubuntu22-desktop:~# scp lbadmin@192.168.101.254:/home/lbadmin/capturas/2min_1a1_ens3.pcap /home/user/Desktop
2min_1a1_ens3.pcap          100% 118KB 14.7MB/s  00:00

root@ubuntu:~# tcpdump -G 300 -W 1 -w /home/lbadmin/capturas/5min_1a1_ens3.pcap -i ens3 & tcpdump -G 300 -W 1 -w /home/lbadmin/capturas/5min_1a1_ens4.pcap -i ens4

root@ubuntu22-desktop:~# scp lbadmin@192.168.101.254:/home/lbadmin/capturas/5min_1a1_ens3.pcap /home/user/Desktop
5min_1a1_ens3.pcap          100% 294KB 30.9MB/s  00:00
root@ubuntu22-desktop:~# scp lbadmin@192.168.101.254:/home/lbadmin/capturas/5min_1a1_ens4.pcap /home/user/Desktop
5min_1a1_ens4.pcap          100% 294KB 38.6MB/s  00:00

```

**Figura 3.62** Importe de archivos “.pcap” (relación 1:1)

Para realizar la comparativa, en el terminal de monitoreo se abren los archivos de medición que se encuentran en el escritorio del terminal monitor mediante el programa “Wireshark”, se examinan como se observan en Figura 3.63 y Figura 3.64, correspondientes a las capturas del tráfico de red de las interfaces de salida a Internet “ens3” y “ens4” de manera respectiva, primero entre los paquetes ICMP que se observan, existen varios con la dirección de *request* o *reply* 192.168.101.1 y 192.168.101.254, nótese que estos son del tipo ping, que usa el módulo de *failover* “foolsm” del balanceador de carga para la detección de estado de conexión, por esto se busca en el tráfico los paquetes ICMP de tipo ping, con dirección 8.8.8.8 en *reply* y *request*.

No.	Time	Source	Destination	Protocol	Length	Info
25	1.793125	192.168.101.254	192.168.101.1	ICMP	94	Echo (ping) request i
26	1.793439	192.168.101.1	192.168.101.254	ICMP	94	Echo (ping) reply i
27	2.044859	192.168.101.254	192.168.101.1	ICMP	94	Echo (ping) request i
28	2.045262	192.168.101.1	192.168.101.254	ICMP	94	Echo (ping) reply i
29	2.296192	192.168.101.254	192.168.101.1	ICMP	94	Echo (ping) request i
30	2.296569	192.168.101.1	192.168.101.254	ICMP	94	Echo (ping) reply i
31	2.460974	192.168.101.254	8.8.8.8	ICMP	98	Echo (ping) request i
32	2.477262	8.8.8.8	192.168.101.254	ICMP	98	Echo (ping) reply i
33	2.547230	192.168.101.254	192.168.101.1	ICMP	94	Echo (ping) request i

▶ Frame 4: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)  
 ▶ Ethernet II, Src: NexComm\_00:05:00 (00:50:00:00:05:00), Dst: NexComm\_00:04:00 (00:50:00:00:04:00)  
 ▶ Internet Protocol Version 4, Src: 192.168.101.253, Dst: 192.168.101.254  
 ▶ Transmission Control Protocol, Src Port: 59856, Dst Port: 22, Seq: 1, Ack: 61, Len: 0

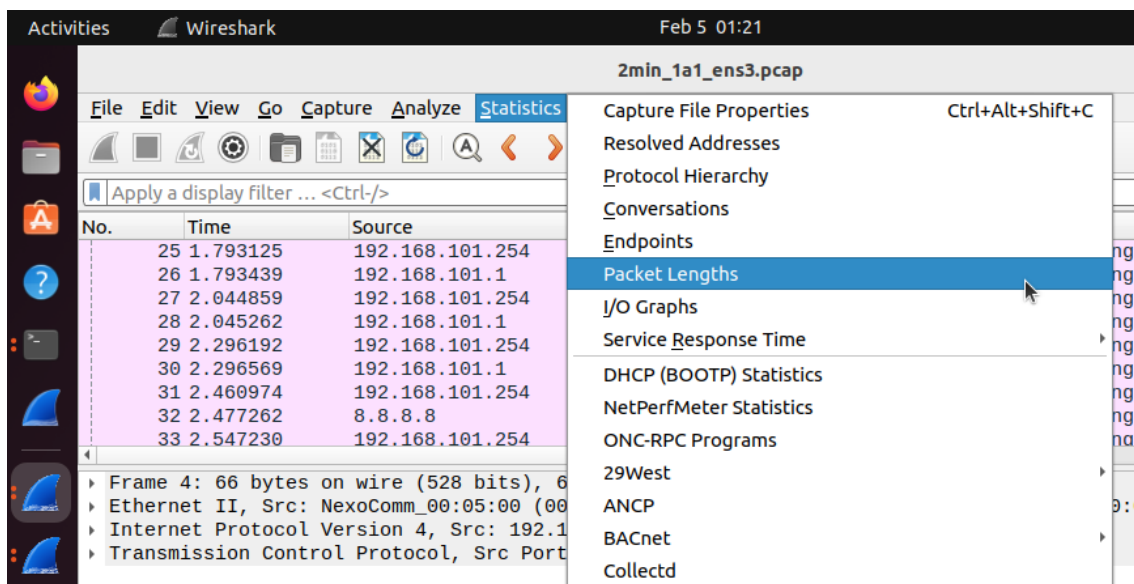
**Figura 3.63** Interfaz de salida a Internet ens3 (captura wireshark)

No.	Time	Source	Destination	Protocol	Length	Info
10	1.003912	203.0.113.254	203.0.113.1	ICMP	94	Echo (ping) request i
11	1.004203	203.0.113.1	203.0.113.254	ICMP	94	Echo (ping) reply i
12	1.256313	203.0.113.254	203.0.113.1	ICMP	94	Echo (ping) request i
13	1.256627	203.0.113.1	203.0.113.254	ICMP	94	Echo (ping) reply i
14	1.293038	203.0.113.254	8.8.8.8	ICMP	98	Echo (ping) request i
15	1.309287	8.8.8.8	203.0.113.254	ICMP	98	Echo (ping) reply i
16	1.506881	203.0.113.254	203.0.113.1	ICMP	94	Echo (ping) request i
17	1.507343	203.0.113.1	203.0.113.254	ICMP	94	Echo (ping) reply i
18	1.758663	203.0.113.254	203.0.113.1	ICMP	94	Echo (ping) request i

▶ Frame 1: 94 bytes on wire (752 bits), 94 bytes captured (752 bits)  
 ▶ Ethernet II, Src: NexComm\_00:04:01 (00:50:00:00:04:01), Dst: aa:bb:cc:00:30:10 (aa:bb:cc:00:30:10)  
 ▶ Internet Protocol Version 4, Src: 203.0.113.254, Dst: 203.0.113.1  
 ▶ Internet Control Message Protocol

**Figura 3.64** Interfaz de salida a Internet ens4 (captura wireshark)

Una vez ubicada la existencia de este tipo de paquetes, en Wireshark se dirige al menú “statistics/Package Lengths” como precisa la Figura 3.65.



**Figura 3.65** Apartado de estadísticas (Wireshark)

Dentro de “Packet Lengths”, se aplica el filtro “ip,addr == 8.8.8.8” para que se muestren solo los paquetes que tengan esa dirección en sus campos *reply* o *request*. Como se precisa en la Figura 3.66 y Figura 3.67, en la interfaz ens3 existen 118 paquetes, de longitud de 80 a 159 Bytes, correspondientes a tráfico ICMP, con el filtro en el campo “Display filter” (verde) de dirección 8.8.8.8; en la interfaz ens4, existen 116 paquetes, de longitud de 80 a 159 Bytes, correspondientes a tráfico ICMP, con el filtro en el campo “Display filter” (verde) de dirección 8.8.8.8.

The screenshot shows the 'Packet Lengths' statistics window in Wireshark. The display filter is set to 'ip.addr == 8.8.8.8'. The table below shows the distribution of packet lengths.

Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
Packet Lengths	118	98.00	98	98	0.0010	100%	0.0200	0.425
0-19	0	-	-	-	0.0000	0.00%	-	-
20-39	0	-	-	-	0.0000	0.00%	-	-
40-79	0	-	-	-	0.0000	0.00%	-	-
80-159	118	98.00	98	98	0.0010	100.00%	0.0200	0.425
160-319	0	-	-	-	0.0000	0.00%	-	-
320-639	0	-	-	-	0.0000	0.00%	-	-
640-1279	0	-	-	-	0.0000	0.00%	-	-
1280-2559	0	-	-	-	0.0000	0.00%	-	-
2560-5119	0	-	-	-	0.0000	0.00%	-	-
5120 and greater	0	-	-	-	0.0000	0.00%	-	-

**Figura 3.66** Conteo de paquetes 2 minutos (ens3)



Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
Packet Lengths	116	98.00	98	98	0.0010	100%	0.0200	1.293
0-19	0	-	-	-	0.0000	0.00%	-	-
20-39	0	-	-	-	0.0000	0.00%	-	-
40-79	0	-	-	-	0.0000	0.00%	-	-
80-159	116	98.00	98	98	0.0010	100.00%	0.0200	1.293
160-319	0	-	-	-	0.0000	0.00%	-	-
320-639	0	-	-	-	0.0000	0.00%	-	-
640-1279	0	-	-	-	0.0000	0.00%	-	-
1280-2559	0	-	-	-	0.0000	0.00%	-	-
2560-5119	0	-	-	-	0.0000	0.00%	-	-
5120 and greater	0	-	-	-	0.0000	0.00%	-	-

Display filter:

**Figura 3.67** Conteo de paquetes 2 minutos (ens4)

Debido a que la relación de tráfico configurada en este caso es de 1:1, se puede obtener estadísticas de efectividad, así para mediante la relación del número de paquetes ens3 y ens4 se tiene que:

En la relación de tráfico 1:1, de manera ideal, el resultado de la división del número de paquetes es igual a 1 o 50% y 50%, sin embargo, en la prueba se obtiene el resultado en la Ecuación 3.1.

$$\frac{n^{\circ} \text{ de paquetes en ens4}}{n^{\circ} \text{ de paquetes en ens3}} = \frac{116}{118} = 0.9831$$

**Ecuación 3.1** Relación de tráfico 1:1 (2min)

En porcentaje:

$$118 + 116 = 234 \text{ paquetes}$$

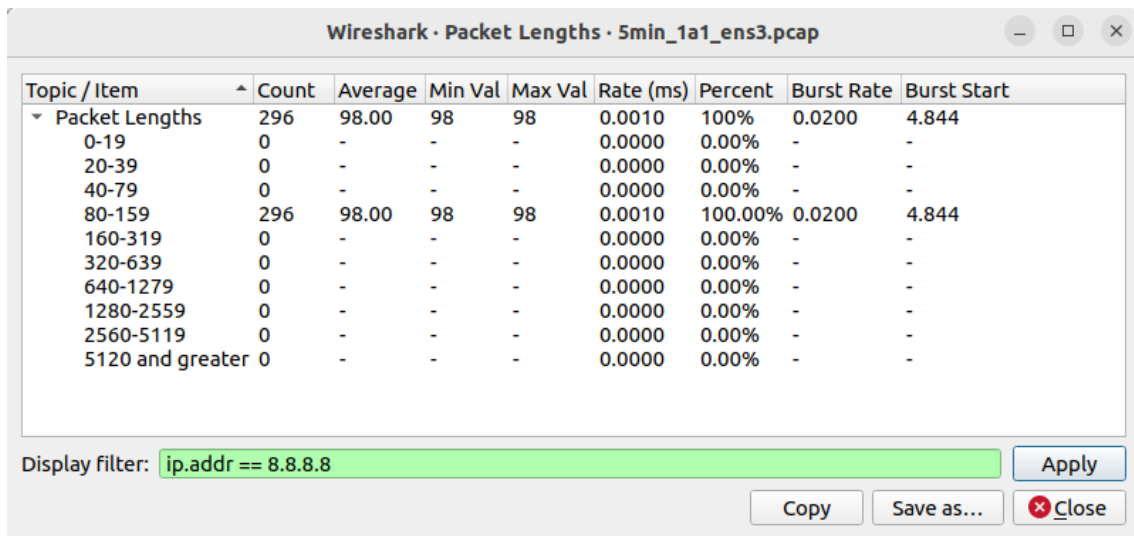
$$\rightarrow \text{Para ens3: } \frac{118}{234} * 100 = 50.43\% \quad \wedge \quad \text{Para ens4 } \frac{116}{234} * 100 = 49.57\%$$

**Ecuación 3.2** Porcentaje de tráfico en las interfaces (1:1, 2min)

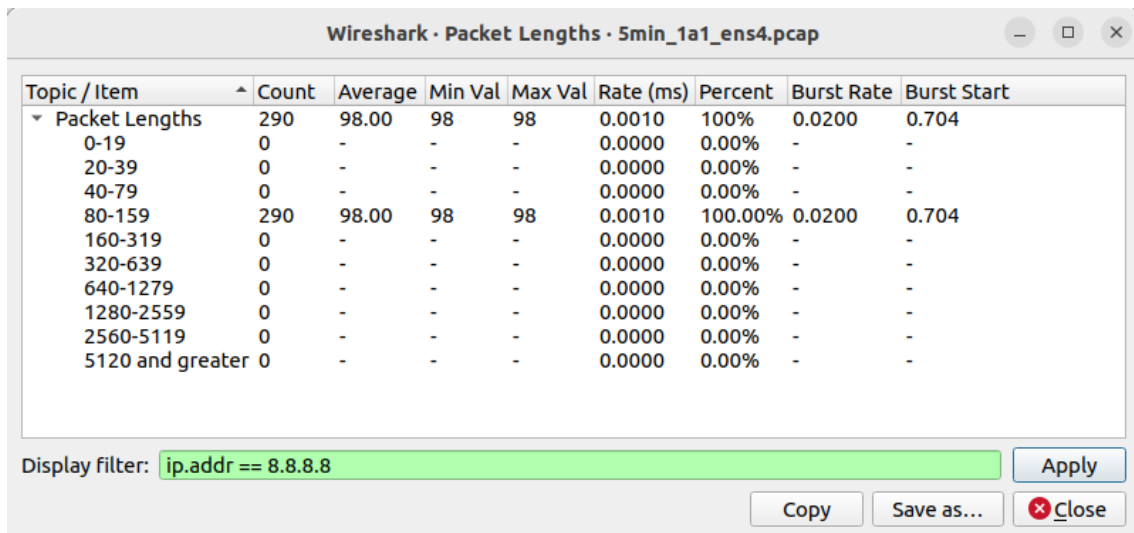
Con en la Ecuación 3.2, en la prueba de 2 minutos y relación de tráfico 1:1, se obtiene 50.43% de tráfico a través de la interfaz ens3 y 49.57% de tráfico por la interfaz ens4.

Para continuar con la segunda medida de tráfico de red, de relación 1:1 y 5 minutos de captura; se abren los archivos “.pcap” correspondientes, en la sección “Packet Lengths” aplica vuelve aplicar el filtro “ip.addr == 8.8.8.8”. Tal como se precisa en la Figura 3.68 y Figura 3.69, en la interfaz ens3 existen 296 paquetes, de longitud de 80 a 159 Bytes,

correspondientes a tráfico ICMP, con el filtro en el campo “Display filter” (verde) de dirección 8.8.8.8; en la interfaz ens4, existen 290 paquetes, de longitud de 80 a 159 Bytes, correspondientes a tráfico ICMP, con el filtro en el campo “Display filter” (verde) de dirección 8.8.8.8.



**Figura 3.68** Conteo de paquetes (Packet Lengths, ens3)



**Figura 3.69** Conteo de paquetes (Packet Lengths, ens4)

Debido a que la relación de tráfico configurada en este caso es de 1:1, se puede obtener estadísticas de efectividad, así para mediante la relación del número de paquetes ens3 y ens4 se tiene que:

En la relación de tráfico 1:1, de manera ideal, el resultado de la división del número de paquetes es igual a 1 o 50% y 50%, sin embargo, en la prueba se obtiene el resultado en la Ecuación 3.3.

$$\frac{n^{\circ} \text{ de paquetes en ens4}}{n^{\circ} \text{ de paquetes en ens3}} = \frac{290}{296} = 0.9797$$

**Ecuación 3.3** Relación de tráfico 1:1 (5min)

En porcentaje:

$$296 + 290 = 586 \text{ paquetes}$$

$$\rightarrow \text{Para ens3: } \frac{296}{586} * 100 = 50.51\% \quad \wedge \quad \text{Para ens4 } \frac{290}{586} * 100 = 49.49\%$$

**Ecuación 3.4** Porcentaje de tráfico en las interfaces (1:1, 5min)

Con en la Ecuación 3.2, en la prueba de 2 minutos y relación de tráfico 1:1, se obtiene 50.43% de tráfico a través de la interfaz ens3 y 49.57% de tráfico por la interfaz ens4. De lo anterior se calcula el error mediante la Ecuación 3.5, Ecuación 3.6 y Ecuación 3.7 [72], [73].

$$\text{Error absoluto} = (\text{medido} - \text{calculado})$$

$$\text{Error absoluto} = 0.9831 - 1 = -0.0169$$

**Ecuación 3.5** Cálculo del error absoluto [72], [73]

$$\text{Error relativo} = \left( \left( \frac{\text{error}_{\text{absoluto}}}{\text{calculado}} \right) x 100 \right) \%$$

$$\text{Error relativo} = \left( \left( \frac{-0.0169}{1} \right) x 100 \right) \% = -1.69\%$$

**Ecuación 3.6** Cálculo del error relativo [72], [73]

$$\text{corrección} = (-(\text{error absoluto}))$$

$$\text{corrección} = 0.0169$$

**Ecuación 3.7** Corrección [72], [73]

Con en la Ecuación 3.4, en la prueba de 5 minutos y relación de tráfico 1:1, se obtiene 50.51% de tráfico a través de la interfaz ens3 y 49.49% de tráfico por la interfaz ens4. De lo anterior se calcula el error mediante la Ecuación 3.8, Ecuación 3.9 y Ecuación 3.10 [72], [73].

$$\text{Error absoluto} = (\text{medido} - \text{calculado})$$

$$\text{Error absoluto} = 0.9797 - 1 = -0.0203$$

**Ecuación 3.8** Cálculo del error absoluto [72], [73]

$$\text{Error relativo} = \left( \frac{\text{error}_{\text{absoluto}}}{\text{calculado}} \times 100 \right) \%$$

$$\text{Error relativo} = \left( \frac{-0.0203}{1} \times 100 \right) \% = -2.03\%$$

**Ecuación 3.9** Cálculo del error relativo [72], [73]

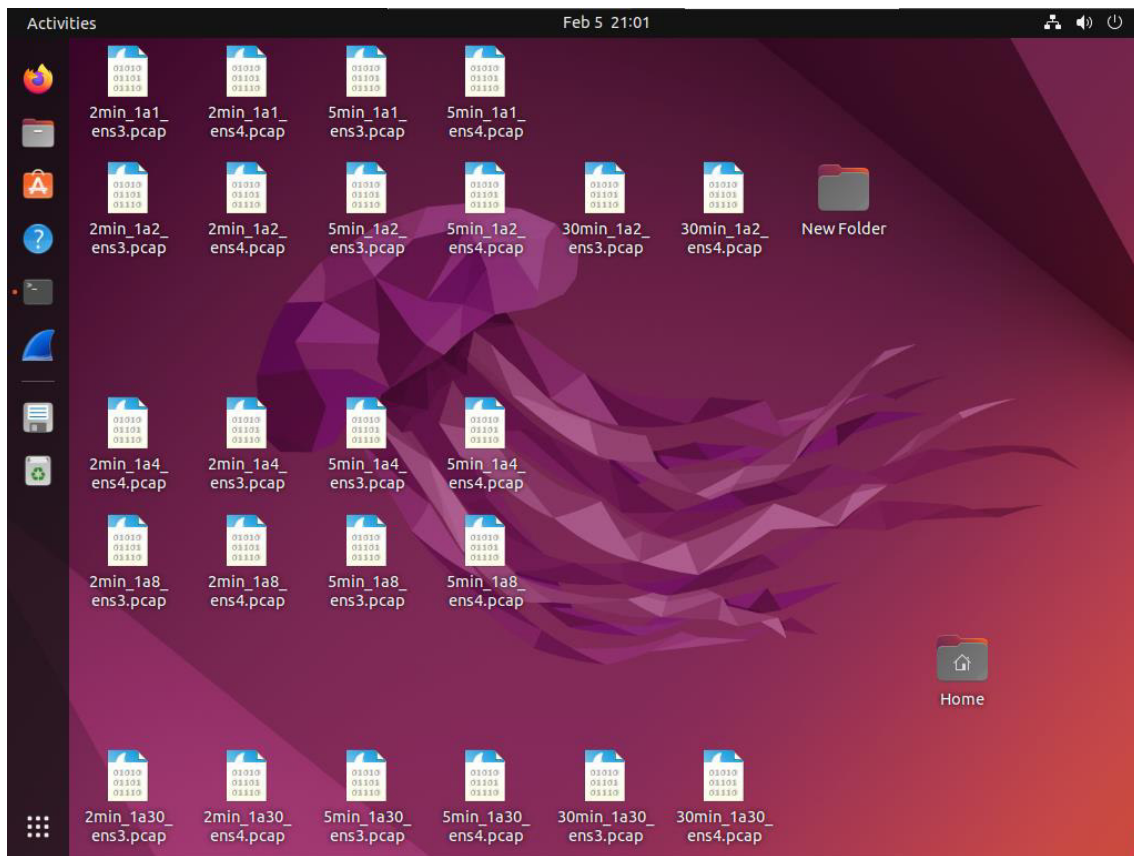
$$\text{corrección} = (-(\text{error absoluto}))$$

$$\text{corrección} = 0.0203$$

**Ecuación 3.10** Corrección [72], [73]

### **Pruebas estadísticas del balanceador de carga**

Como se observa en la Figura 3.70, después de seguir los pasos de la sección anterior, se obtuvieron 14 archivos “.pcap” adicionales en el escritorio del terminal monitor, para ser analizados de la misma manera; estos corresponden a pruebas de 2 y 5 minutos de funcionamiento con tráfico ICMP del comando ping, en relaciones de 1:2, 1:4, 1:8 y 1:30, de manera respectiva.



**Figura 3.70** Archivos de tráfico para análisis

Si repite el análisis de tráfico de la sección anterior (Proceso de las pruebas estadísticas del balanceador), para cada uno de los archivos capturados, se obtiene como resultado los datos recopilados en la Tabla 3.2 y Tabla 3.3, donde se presenta un análisis porcentual del tráfico de red un porcentajes de tráfico asignados y cálculo de error de manera respectiva [72], [73].

**Tabla 3.2** Análisis porcentual de tráfico de red [72], [73]

	Tiempo	Relación	# de paquetes en ens3	# de paquetes en ens4	# de paquetes total	Porcentaje de tráfico en ens3 (calculado)	Porcentaje de tráfico en ens3	Porcentaje de tráfico en ens4 (calculado)	Porcentaje de tráfico en ens4
Caso 1	2 min	1:1	118	116	234	50.00%	50.43%	50.00%	49.57%
	5 min	1:1	296	290	586	50.00%	50.51%	50.00%	49.49%
Caso 2	2 min	1:2	82	35	117	66.67%	70.09%	33.33%	29.91%
	5 min	1:2	198	97	295	66.67%	67.12%	33.33%	32.88%
Caso 3	2 min	1:4	190	44	234	80.00%	81.20%	20.00%	18.80%
	5 min	1:4	480	106	586	80.00%	81.91%	20.00%	18.09%
Caso 4	2 min	1:8	178	54	232	88.89%	76.72%	11.11%	23.28%
	5 min	1:8	516	98	614	88.89%	84.04%	11.11%	15.96%
Caso 5	2 min	1:30	216	18	234	96.77%	92.31%	3.23%	7.69%
	5 min	1:30	562	24	586	96.77%	95.90%	3.23%	4.10%

Caso 6	30 min	1:2	1644	847	2491	66.67%	66.00%	33.33%	34.00%
	30 min	1:30	1711	64	1775	96.77%	96.39%	3.23%	3.61%
	120 min	1:30	13644	480	14124	96.77%	96.60%	3.23%	3.40%

**Tabla 3.3** Cálculo de error del tráfico de red [72], [73]

	Tiempo	Relación	Relación ens4:ens3 (teórica)	Relación ens4:ens3 (medida)	Error absoluto	Error relativo	Corrección
Caso 1	2 min	1:1	1.0000	0.9831	-0.0169	-1.69%	0.0169
	5 min	1:1	1.0000	0.9797	-0.0203	-2.03%	0.0203
Caso 2	2 min	1:2	0.5000	0.4268	-0.0732	-14.63%	0.0732
	5 min	1:2	0.5000	0.4899	-0.0101	-2.02%	0.0101
Caso 3	2 min	1:4	0.2500	0.2316	-0.0184	-7.37%	0.0184
	5 min	1:4	0.2500	0.2208	-0.0292	-11.67%	0.0292
Caso 4	2 min	1:8	0.1250	0.3034	0.1784	142.70%	-0.1784
	5 min	1:8	0.1250	0.1899	0.0649	51.94%	-0.0649
Caso 5	2 min	1:30	0.0330	0.0833	0.0503	152.53%	-0.0503
	5 min	1:30	0.0330	0.0427	0.0097	29.41%	-0.0097
Caso 6	30 min	1:2	0.5000	0.5152	0.0152	3.04%	-0.0152
	30 min	1:30	0.0330	0.0374	0.0044	13.35%	-0.0044
	120 min	1:30	0.0330	0.0352	0.0022	6.61%	-0.0022

Estos resultados indican, en el primer caso, en general, el caso 1 con distribución de tráfico 1:1, un error relativo inferior al 2.03% ; en el segundo caso, con distribución de tráfico 1:2, en la prueba de dos minutos, un error relativo de 14.63%, al ser mayor al 7% indica que existe error debido al tiempo de ejecución, es decir en 2 minutos y con dicho radio de compartición se necesita de mayor tiempo para poder obtener un resultado correcto, el que si se observa cuando la prueba corre durante cinco minutos, con un error relativo de 2.02%; en el tercer, cuarto y quinto caso, los tiempos de prueba son insuficientes para arrojar resultados de error certeros, es decir los errores son mayores al 7%, debido a esto se planteó un sexto caso, donde se prueba de manera adecuada el mayor radio de compartición (1:30), es decir con tiempo extendido.

Para el caso 6, se hizo primero una prueba de 20 minutos, con relación 1:2, esta arrojó resultados consistentes con un error relativo inferior a 3.04%; a continuación, se realizó una prueba de 30 minutos, con una relación 1:30, esta arrojó que el tiempo sigue siendo insuficiente para apreciar el correcto funcionamiento en esta relación de tráfico, ya que el error relativo es de 13.35%; por último, se realizó una prueba de 120 minutos, con una relación 1:30, esta arrojó como resultado que el tiempo cumple con las expectativas

de resolución de muestreo para esta relación de tráfico, demostrando un error relativo 6.61%, debajo de del límite de error de 7%.

Queda demostrado que mientras más grande sea la relación de tráfico, más tiempo se necesita para realizar el muestreo, es decir la efectividad del algoritmo (*statistic*) causa que la resolución sea menor cada vez que se incrementan las líneas de tráfico o la relación de estas, por esto, el comportamiento del balanceador será mejor observable en relaciones de tráfico inferiores a 1:4, de esta manera se cuantiza la efectividad del balanceador de carga.

## 4 CONCLUSIONES

- Se concluye que, en la etapa de identificación de los requerimientos del balanceador de carga, fue un factor determinante la limitación de la escalabilidad del proyecto, pues está al ser enfocada en pequeña y mediana empresa, sienta la base de la elección del módulo *iproute2* el cual permite establecer políticas de enrutamiento avanzado, que junto al algoritmo *statistic* del módulo, permiten la clasificación del tráfico en forma de paquetes marcados.
- Se concluye que el componente crucial durante la implementación del prototipo fue la plataforma de simulación de redes EVE-NG, debido a su fidelidad de funcionamiento en la simulación de los equipos, sumado a su ágil esquematización y presentación, permite crear un escenario de configuración física observable, cuyas pruebas preliminares son de fácil implementación y observación con el programa Wireshark que se encuentra embebido en la instalación de EVE-NG, lo cual a su vez permitió el desarrollo del concepto de las pruebas de medición de tráfico con el programa *tcpdump* en el servidor.
- Durante la determinación del sistema operativo se comprobó que la configuración del balanceador, al estar basada kernel de Linux, es aplicable a cualquiera de las distribuciones que contengan el módulo *iproute2*, sin embargo, se estableció la elección de Ubuntu Server debido al amplio repositorio documental, variedad de arquitecturas, apoyo de desarrollador y comunidad, factores clave para el desarrollo de las funciones de marcado, enrutamiento y asignación de peso de tráfico usadas en el desarrollo del balanceador.
- Durante el diseño del proyecto se determinó que antes de la implementación del balanceo de carga, se deben primero implementar dos tablas de enrutamiento, cada una correspondiente a una conexión de Internet, en éstas se relaciona la

interfaz de servicio de la red LAN, con cada una de las interfaces de conexión a Internet respectivas, de esta manera se especifican las rutas de tráfico que el balanceador usará después en su configuración.

- Se concluye que la funcionalidad primordial del balanceador de carga está dada por la herramienta de marcado *fwmark* de *iptables*, ya que esta funciona primero al dividir el tráfico en lotes de determinado número de paquetes, después establece una etiqueta o marca que se asigna a cada lote y con ayuda de un contador clasifica el flujo de cada lote hacia una de las interfaces de conexión a Internet.
- Durante el diseño del prototipo se observó que las medidas de seguridad o *hardening* del servidor deben ser aplicadas con mesura, es decir que no entorpezcan las actividades de servicio del cliente como tampoco las actividades de monitoreo, de esta manera, mediante el uso de *apparmor* para el manejo de ejecución de aplicaciones, suricata configurado como IDS, el firewall de *iptables* que permite solo el uso de SSH encriptado y con autenticación además de las protecciones esenciales varias, un correcto manejo de usuarios, todo esto sumado al uso de *logwatch* como sistema centralizado de monitoreo de eventos, consolidan un panorama de *hardening* equilibrado, que no compromete la funcionalidad del servicio de balanceo de carga ni las actividades de administración y monitoreo de sistema.
- Durante la implementación del balanceador de carga, se determinó que los clientes tanto de Windows como de Linux presentan una conectividad total, en línea con el funcionamiento esperado. Se destaca que este comportamiento se debe a que la implementación es de tipo virtualizada, lo que garantiza que los parámetros de funcionamiento sean ideales. Por consiguiente, no existen variables físicas que introduzcan inestabilidad al sistema.
- En la etapa de pruebas del balanceador de carga se determinó que el módulo *foolsm*, encargado del sistema de *failover*, requiere un ping configurable y constante hacia los *gateways* de los ISP, razón por la cual, el tiempo de restablecimiento del servicio de Internet depende del ajuste de tiempo entre intervalos de ping configurado. Por esto, en caso de corte de cualquiera de las conexiones, el intervalo de tiempo colocado de 250 milisegundos permite obtener una respuesta de restablecimiento óptima en el módulo de *failover* para las conexiones de Internet.
- Durante las pruebas de efectividad se evidenció que, para la correcta medición del funcionamiento de la asignación de peso del balanceador de carga, se deben



hacer pruebas extensas (de más de 40 minutos), ya que el balanceador solo arrojará resultados satisfactorios en pruebas de 5 a 2 minutos con ratios de tráfico inferiores a 1:4, para ratios de asignación de tráfico superiores, las pruebas de tiempo estarán relacionadas de manera incremental a mayor orden de división en la fracción de asignación de tráfico designada.

## 5 RECOMENDACIONES

- Dado que el prototipo fue diseñado para admitir dos conexiones de Internet y su efectividad fue comprobada durante la implementación actual, se sugiere realizar pruebas adicionales en el futuro para determinar la cantidad máxima de conexiones de Internet que el diseño puede soportar. Esto proporcionará una comprensión precisa del límite superior de escalabilidad del prototipo.
- Se recomienda encarecidamente revisar exhaustivamente la documentación y manuales relacionados con el módulo *iproute2* antes de iniciar cualquier configuración. Esto permitirá una comprensión completa de las capacidades que ofrece este módulo.
- Para garantizar una implementación de seguridad correcta, se recomienda mantener un enfoque equilibrado al aplicar políticas de endurecimiento (*hardening*) al servidor. Esto asegurará una mayor accesibilidad funcional tanto para el usuario como para el administrador.
- Dado que la configuración presentada en este documento implica una implementación simulada del prototipo, se sugiere considerar una implementación física en el futuro. Esto permitirá observar el funcionamiento del prototipo en un entorno real.
- En caso de considerar una implementación física del proyecto, se recomienda utilizar como plataforma el módulo Raspberry Pi 3 modelo B+ o cualquier equipo que cumpla con especificaciones de hardware equivalentes o superiores.
- Para mejorar la eficiencia en el uso de la red en la detección del estado de conectividad en el módulo de *failover*, se sugiere investigar la posibilidad de implementar un método de sensado de estado de conexión tipo *link state* en el futuro, similar al protocolo "ospf" que se encuentra presente en el módulo *iproute2*.

## 6 REFERENCIAS BIBLIOGRÁFICAS

- [1] Emulated Virtual Environment Next Generation, «eve-ng,» 2023. [En línea]. Available: <https://www.eve-ng.net/index.php/documentation/release-notes/>. [Último acceso: 16 Diciembre 2023].
- [2] J. G. H. Thomas L. Casavant, «A Taxonomy of Scheduling in General-Purpose Distributed Computing Systems,» *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, vol. XIV, nº 2, pp. 1-9, 1988.
- [3] N. Rathore, «A REVIEW TOWARDS: LOAD BALANCING TECHNIQUES,» *i-manager's Journal on Power Systems Engineering*, vol. IV, nº 4, pp. 47-60, 2017.
- [4] appviewx, «Load Balancer and Types,» appviewX, 2023. [En línea]. Available: <https://www.appviewx.com/education-center/load-balancer-and-types/>. [Último acceso: 20 Noviembre 2023].
- [5] Byte, «Balanceo de carga,» Redestelco, 17 Abril 2019. [En línea]. Available: [https://redesteleco.com/balanceo\\_de\\_carga/](https://redesteleco.com/balanceo_de_carga/). [Último acceso: 20 Noviembre 2023].
- [6] A. W. S. F. R. L. A. R. Ellen Siever, «Intoduction,» de *Linux in a Nutshell*, Quinta ed., A. Oram, Ed., Sebastopol, O'Reilly Media, Inc., 2005, pp. 1-12.
- [7] Red Hat, Inc., «¿Qué es YAML?,» 03 Marzo 2023. [En línea]. Available: <https://www.redhat.com/es/topics/automation/what-is-yaml>. [Último acceso: 21 Noviembre 2023].
- [8] Canonical Ltd., «Netplan The network configuration abstraction renderer,» 01 Enero 2024. [En línea]. Available: <https://netplan.io/>. [Último acceso: 12 Enero 2024].
- [9] The Linux Foundation, «iproute2,» 04 Noviembre 2023. [En línea]. Available: <https://wiki.linuxfoundation.org/networking/iproute2#documentation>. [Último acceso: 12 Enero 2024].
- [10] Arimetrics, «<https://www.arimetrics.com/glosario-digital/perl>,» 2022. [En línea]. Available: <https://www.arimetrics.com/glosario-digital/perl>. [Último acceso: 01 Diciembre 2023].

- [11] H. Coggill, «What is System Hardening? Essential Checklists from OS to Applications,» Canonical, Ubuntu, 27 Marzo 2023. [En línea]. Available: <https://ubuntu.com/blog/what-is-system-hardening-definition-and-best-practices>. [Último acceso: 18 Diciembre 2023].
- [12] Canonical Ltd., «A guide to infrastructure hardening Best practices to improve the security posture of your Linux-based infrastructure deployments,» 15 Enero 2023. [En línea]. Available: <https://ubuntu.com/engage/a-guide-to-infrastructure-hardening>. [Último acceso: 16 Diciembre 2023].
- [13] Cisco Systems, Inc., «What Is a Firewall?,» 2023. [En línea]. Available: <https://www.cisco.com/c/en/us/products/security/firewalls/what-is-a-firewall.html>. [Último acceso: 05 12 2023].
- [14] Barracuda Networks, Inc., «Intrusion Detection System,» 2023. [En línea]. Available: <https://www.barracuda.com/support/glossary/intrusion-detection-system>. [Último acceso: 15 Diciembre 2023].
- [15] C. L. T. Ylonen, «The Secure Shell (SSH) Protocol Architecture,» Request for Comments: 4251, Enero 2006. [En línea]. Available: <https://www.rfc-editor.org/rfc/rfc4251>. [Último acceso: 20 Diciembre 2023].
- [16] T. O'Connor, Interviewee, *Best Linux distro for load-balancers?*. [Entrevista]. 03 Abril 2013.
- [17] P. Gedam, «Netplan documentation,» Netplan team, 2022. [En línea]. Available: <https://netplan.readthedocs.io/en/stable/examples/>. [Último acceso: 23 Noviembre 2023].
- [18] T. Hill, «ituonline,» 28 Septiembre 2023. [En línea]. Available: <https://www.ituonline.com/blogs/best-network-simulator-for-cisco/>. [Último acceso: 16 Diciembre 2023].
- [19] H. Andrea, «Networks Training,» 17 Noviembre 2020. [En línea]. Available: <https://www.networkstraining.com/network-simulation-software-tools/>. [Último acceso: 15 Diciembre 2023].
- [20] Y. C. Yuen, «infostyle,» 13 Marzo 2023. [En línea]. Available: <https://infosyte.com/network-simulators/>. [Último acceso: 16 Diciembre 2023].

- [21] T. O'Connor y A. Chauhan, «serverfault,» 02 Abril 2010. [En línea]. Available: <https://serverfault.com/questions/128869/best-linux-distro-for-load-balancers>. [Último acceso: 16 Diciembre 2023].
- [22] theLAN Tamer, «youtube,» 15 Marzo 2018. [En línea]. Available: <https://www.youtube.com/watch?v=pwNKuu1eMSc&t=1857s>. [Último acceso: 15 Diciembre 2023].
- [23] J. Grossmann, «gns3,» 06 Noviembre 2023. [En línea]. Available: <https://www.gns3.com/community/blog/gns3-2-2-44-released>. [Último acceso: 16 Diciembre 2023].
- [24] «Universidad Complutense de Madrid,» 2023. [En línea]. Available: <https://www.ucm.es/pimcd2014-free-software/gns3>. [Último acceso: 16 Diciembre 2023].
- [25] «Institut Puig Castellar,» Fundació Plone, 2023. [En línea]. Available: <https://elpuig.xeill.net/Members/juanmorote/articulos/verificar-practicas-de-gns3-con-teuton>. [Último acceso: 16 Diciembre 2023].
- [26] L. Golightly, P. Modesti y V. Chang, «MDPI,» 06 Junio 2023. [En línea]. Available: <https://www.mdpi.com/2624-800X/3/3/24>. [Último acceso: 16 Diciembre 2023].
- [27] H. Y. C., «EVE-NG & GNS3 are Delicious Bites for Network-Engineers,» 18 Diciembre 2022. [En línea]. Available: <https://www.linkedin.com/pulse/eve-ng-gns3-delicious-bites-network-engineers-ccie-sp-msc/>. [Último acceso: 16 Diciembre 2023].
- [28] S. Mukherjee, «Deploying and Using vJunos in a Bare Metal EVE-NG server,» 05 Noviembre 2023. [En línea]. Available: <https://community.juniper.net/blogs/shalini-mukherjee/2023/05/11/deploying-vjunos-in-a-bare-metal-eve-ng-server>. [Último acceso: 16 Diciembre 2023].
- [29] M. Rafati, «Scribd,» 2023. [En línea]. Available: <https://www.scribd.com/document/487603859/Deployment-Cisco-SD-WAN-LAB-on-EVE-NG-pdf>. [Último acceso: 16 Diciembre 2023].
- [30] D. Cheung, «MPLS VPLS Interop — Nokia 7750 and Juniper MX on EVE-NG,» 19 Enero 2023. [En línea]. Available: <https://derekcheung.medium.com/mpls-vpls->

interop-nokia-7750-and-juniper-mx-on-eve-ng-3d419c7c52d5. [Último acceso: 16 Diciembre 2023].

[31] CISCO, «netacad,» 2023. [En línea]. Available: <https://www.netacad.com/courses/packet-tracer>. [Último acceso: 16 Diciembre 2023].

[32] packettracernetwork, «Packet Tracer 8.2.1 labs,» 25 Abril 2023. [En línea]. Available: <https://www.packettracernetwork.com/labs/packettracerlabs.html>. [Último acceso: 16 Diciembre 2023].

[33] J. Rinehart, «ICND1 Lab Exercises,» 02 Mayo 2023. [En línea]. Available: [https://www.academia.edu/35210353/ICND1\\_Lab\\_Exercises](https://www.academia.edu/35210353/ICND1_Lab_Exercises). [Último acceso: 16 Diciembre 2023].

[34] Cisco, «The Cisco Learning Network,» 08 Abril 2020. [En línea]. Available: <https://learningnetwork.cisco.com/s/question/0D53i00000S9IQbCAN/virl-release-date-branding-change-announcement>. [Último acceso: 16 Diciembre 2023].

[35] Cisco, «Cisco DevNet,» 20 Julio 2023. [En línea]. Available: <https://developer.cisco.com/docs/modeling-labs/#!cml-release-notes>. [Último acceso: 16 Diciembre 2023].

[36] Huawei, «Huawei Enterprise Support Community,» 30 Junio 2023. [En línea]. Available: <https://forum.huawei.com/enterprise/en/hcie-storage/thread/674841195592957952-667213863596208128>. [Último acceso: 16 Diciembre 2023].

[37] Canonical Ltd., «Basic installation,» 2023. [En línea]. Available: <https://ubuntu.com/server/docs/installation>. [Último acceso: 16 Diciembre 2023].

[38] Canonical Ltd, «released, Canonical Ubuntu 22.04 LTS is,» 21 Abril 2022. [En línea]. Available: <https://canonical.com/blog/ubuntu-22-04-lts-released>. [Último acceso: 16 Diciembre 2023].

[39] Ł. Zemczak, «Jammy Jellyfish Release Notes,» 10 2021. [En línea]. Available: <https://discourse.ubuntu.com/t/jammy-jellyfish-release-notes/24668>. [Último acceso: 16 Diciembre 2023].

- [40] MAFoElffen, «Thread: nmcli vs netplan,» 02 Enero 2022. [En línea]. Available: <https://ubuntuforums.org/showthread.php?t=2470516>. [Último acceso: 17 Diciembre 2023].
- [41] Ubuntu documentation, «NetworkManager,» 30 Junio 2023. [En línea]. Available: <https://help.ubuntu.com/community/NetworkManager>. [Último acceso: 12 Enero 2024].
- [42] Ubuntu Manuals, «systemd.network - Network configuration,» 01 Enero 2019. [En línea]. Available: <https://manpages.ubuntu.com/manpages/noble/man5/systemd.network.5.html>. [Último acceso: 11 Enero 2024].
- [43] L. D. Stein, «Net-ISP-Balance,» 2021. [En línea]. Available: <https://lstein.github.io/Net-ISP-Balance/>. [Último acceso: 15 Noviembre 2023].
- [44] B. Lee, «Suricata vs Snort: Which is the best IDS?,» virtualizationhowto.com, 19 Octubre 2023. [En línea]. Available: <https://www.virtualizationhowto.com/2023/10/suricata-vs-snort-which-is-the-best-ids/>. [Último acceso: 12 Enero 2024].
- [45] RS, «RASPBERRY PI MODEL COMPARISON TABLE,» 2023. [En línea]. Available: <https://docs.rs-online.com/3669/0900766b816c34e5.pdf>. [Último acceso: Diciembre Enero 2024].
- [46] E. Brown, «Raspberry Pi based computer offers Real-Time Ethernet,» 28 Marzo 2017. [En línea]. Available: <https://linuxgizmos.com/raspberry-pi-based-computer-offers-real-time-ethernet/>. [Último acceso: 12 Enero 2023].
- [47] Canonical Ltd., «Netplan design,» 01 Enero 2023. [En línea]. Available: <https://netplan.io/design>. [Último acceso: 16 Diciembre 2023].
- [48] Canonical Ltd., «OpenSSH Server,» 2023. [En línea]. Available: <https://ubuntu.com/server/docs/service-openssh>. [Último acceso: 10 Diciembre 202].
- [49] J. Chege, 15 Septiembre 2023. [En línea]. Available: <https://thriveread.com/enable-and-disable-linux-ubuntu-ip-forwarding/>. [Último acceso: 18 Diciembre 2023].

- [50] L. D. Stein, «Net-ISP-Balance,» 25 Marzo 2022. [En línea]. Available: <https://lstein.github.io/Net-ISP-Balance/>. [Último acceso: 01 Diciembre 2023].
- [51] wiki.archlinux.org, «Internet sharing,» 16 Enero 2024. [En línea]. Available: [https://wiki.archlinux.org/title/Internet\\_sharing#Enable\\_packet\\_forwarding](https://wiki.archlinux.org/title/Internet_sharing#Enable_packet_forwarding). [Último acceso: 24 Enero 2024].
- [52] T. G. G. M. R. v. M. M. v. O. P. B. S. J. S. P. L. Bert Hubert, «4.8. Routing Tables,» 2023. [En línea]. Available: <http://linux-ip.net/html/routing-tables.html>. [Último acceso: 28 Noviembre 2023].
- [53] T. G. G. M. R. v. M. M. v. O. P. B. S. J. S. P. L. Bert Hubert, «4.2. Routing for multiple uplinks/providers,» 2023. [En línea]. Available: <https://lartc.org/howto/lartc.rpdb.multiple-links.html>. [Último acceso: 10 Diciembre 2023].
- [54] Stamus Networks, «Links Load balancing,» 2023. [En línea]. Available: <https://home.regit.org/netfilter-en/links-load-balancing/>. [Último acceso: 15 Diciembre 2023].
- [55] F. Bar, «Iptables para torpes,» 18 Marzo 2019. [En línea]. Available: <https://elbinario.net/2019/03/18/iptables-para-torpes/>. [Último acceso: 18 Diciembre 2023].
- [56] die.net, «iptables(8) - Linux man page,» 2023. [En línea]. Available: <https://linux.die.net/man/8/iptables>. [Último acceso: 18 Diciembre 2023].
- [57] user213088, «How to save rules of the iptables? [duplicate],» 09 Noviembre 2013. [En línea]. Available: <https://askubuntu.com/questions/119393/how-to-save-rules-of-the-iptables>. [Último acceso: 18 Diciembre 2023].
- [58] A. Prakash, «What is Build Essential Package in Ubuntu? How to Install it?,» 04 Julio 2023. [En línea]. Available: <https://itsfoss.com/build-essential-ubuntu/>. [Último acceso: 18 Diciembre 2023].
- [59] Comprehensive Perl Archive Network, «How to install CPAN modules,» 2023. [En línea]. Available: <https://www.cpan.org/modules/INSTALL.html>. [Último acceso: 19 Diciembre 2023].

- [60] T. Miyagawa, «App::cpanminus,» 30 Julio 2023. [En línea]. Available: <https://metacpan.org/pod/App::cpanminus>. [Último acceso: 19 Diciembre 2023].
- [61] L. D. Stein, «Net::ISP::Balance,» 20 Mayo 2021. [En línea]. Available: <https://metacpan.org/dist/Net-ISP-Balance/view/lib/Net/ISP/Balance.pm>. [Último acceso: 18 Diciembre 2023].
- [62] Stamus Network, «Netfilter Connmark,» 2023. [En línea]. Available: <https://home.regit.org/netfilter-en/netfilter-connmark/>. [Último acceso: 19 Diciembre 2024].
- [63] Ubuntu Documentation, «AppArmor,» 05 Julio 2020. [En línea]. Available: <https://help.ubuntu.com/community/AppArmor>. [Último acceso: 05 Enero 2023].
- [64] nmap.org, «Options Summary,» 2023. [En línea]. Available: <https://nmap.org/book/man-briefoptions.html>. [Último acceso: 05 Enero 2024].
- [65] 22 Octubre 2021. [En línea]. Available: <https://www.digitalocean.com/community/tutorials/how-to-install-suricata-on-ubuntu-20-04>. [Último acceso: 23 Diciembre 2023].
- [66] Ubuntu Documentation, «UbuntuTime,» 13 Julio 2016. [En línea]. Available: <https://help.ubuntu.com/community/UbuntuTime>. [Último acceso: 26 Diciembre 2023].
- [67] Canonical Ltd., «OpenSSH Server,» 2023. [En línea]. Available: <https://ubuntu.com/server/docs/service-openssh>. [Último acceso: 05 Enero 2024].
- [68] Canonical Ltd., «User Management,» 2023. [En línea]. Available: <https://ubuntu.com/server/docs/security-users>. [Último acceso: 08 Enero 2023].
- [69] R. Red, «How to Secure an SSH Server in Ubuntu,» 01 Noviembre 2022. [En línea]. Available: <https://www.maketecheasier.com/secure-ssh-server-ubuntu/>. [Último acceso: 08 Enero 2024].
- [70] CAnonical Ltd., «How to install and configure Logwatch,» Marzo 2023. [En línea]. Available: <https://ubuntu.com/server/docs/logwatch>. [Último acceso: 09 Enero 2024].



- [71] Tcpdump Group, «TCPDUMP(1) MAN PAGE,» 28 Diciembre 2023. [En línea]. Available: <https://www.tcpdump.org/manpages/tcpdump.1.html>. [Último acceso: 12 Enero 2024].
- [72] E. E. M. S. P. Félix Ortiz, «El proceso de medición Análisis y comunicación de datos experimentales,» Universidad Nacional de Río Cuarto, 14 Septiembre 2005. [En línea]. Available: [https://www.unrc.edu.ar/unrc/digital/El\\_proceso\\_de\\_med.pdf](https://www.unrc.edu.ar/unrc/digital/El_proceso_de_med.pdf). [Último acceso: 14 Enero 2023].
- [73] S. M. Rivera, «BREVE INTRODUCCIÓN A LA TEORÍA DE ERRORES Y LA GRAFICACIÓN,» Universidad Autónoma de Aguascalientes, 2017. [En línea]. Available: [https://editorial.uaa.mx/docs/breve\\_introduccion\\_teoriasErrores.pdf](https://editorial.uaa.mx/docs/breve_introduccion_teoriasErrores.pdf). [Último acceso: 14 Enero 2023].

## 7 ANEXOS

La lista de los **Anexos** se muestra a continuación:

ANEXO I. Certificado de originalidad

ANEXO II. Enlaces

## ANEXO III. Conjunto de datos extensos

## ANEXO I: Certificado de Originalidad

F\_AA\_236

### CERTIFICADO DE ORIGINALIDAD TRABAJO DE INTEGRACIÓN CURRICULAR

Quito, D.M. 20 de febrero de 2024

De mi consideración:

Yo, **LEANDRO ANTONIO PAZMIÑO ORTIZ**, en calidad de Director del Trabajo de Integración Curricular titulado **IMPLEMENTACIÓN DE UN BALANCEADOR DE CARGA DE SALIDA A INTERNET BASADO EN UN SERVIDOR LINUX** asociado al proyecto **IMPLEMENTACIÓN DE UN BALANCEADOR DE CARGA DE SALIDA A INTERNET BASADO EN UN SERVIDOR LINUX** elaborado por el estudiante **CARLOS ANDRÉS FLORES ORTIZ** de la carrera en **TECNOLOGÍA SUPERIOR EN REDES Y TELECOMUNICACIONES**, certifico que he empleado la herramienta antiplagio "TURNITIN" para la revisión de originalidad del documento escrito producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 8%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el informe generado por la herramienta Turnitin.

Atentamente,



Leandro Pazmiño Ortiz

Docente

Escuela de Formación de Tecnólogos

## ANEXO II: Enlaces



**Anexo II.I** Código QR de la implementación y pruebas de funcionamiento del prototipo de balanceador de carga de salida a Internet basado en un servidor Linux

Link: <https://n9.cl/lug8i>



**Anexo II.II** Código QR de la secuencia de código del archivo Balance.PM

Link: <https://n9.cl/i0uxh>

## ANEXO III: Códigos Fuente

### Configuración

```
# Configurar la red
```

```
vi /etc/netplan/
```

```
network:
```

```
version: 2
```

```
renderer: networkd
```

```
ethernets:
```

```
ens3:
```

```
dhcp4: no
```

```
addresses:
```

```
- 192.168.101.254/24
```

```
nameservers:
```

```
addresses:
```

```
- 192.168.101.1
```

```
routes:
```

```
- to: 192.168.101.0/24
```

```
via: 192.168.101.1
```

```
metric: 200
```

```
table: 10
```

```
routing-policy:
```

```
- from: 192.168.144.0/24
```

```
table: 10
```

```
ens4:
```

```
dhcp4: no
```

addresses:

- 203.0.113.254/24

nameservers:

addresses:

- 203.0.113.1

routes:

- to: 203.0.113.0/24

via: 203.0.113.1

metric: 300

table: 11

routing-policy:

- from: 192.168.144.0/24

table: 11

ens5:

dhcp4: no

addresses:

- 192.168.144.1/24

nameservers:

addresses:

- 8.8.8.8

ens6:

dhcp4: no

addresses:

- 192.168.152.244/24

nameservers:

addresses:

- 192.168.152.2

routes:

- to: 192.168.152.0/24

via: 192.168.152.2

metric: 100

netplan try

# Habilitar IP Forwarding:

sysctl -w net.ipv4.ip\_forward=1

ir a /etc/sysctl.conf

# Descomentar

net.ipv4.ip\_forward = 1

sysctl -p #para aplicar cambios

# Ruteo basado en políticas:

# añadir en la tabla ubicada en "/etc/iproute2/rt\_tables" los valores:

10 ISP1

11 ISP2

# Reglas para el tráfico

ip rule add from 192.168.101.254 table ISP1

ip rule add from 203.0.113.254 table ISP2

# Reglas para la lan

ip route add 192.168.144.0/24 via 192.168.101.1 table ISP1



```
ip route add 192.168.144.0/24 via 203.0.113.1 table ISP2
```

```
# Habilitar nateo para las interfaces de salida Internet
```

```
iptables -t nat -A POSTROUTING -o ens3 -j MASQUERADE
```

```
iptables -t nat -A POSTROUTING -o ens4 -j MASQUERADE
```

```
#salvar las reglas ip
```

```
apt install iptables-persistent
```

```
iptables-save -f /etc/iptables/rules.v4
```

```
netfilter-persistent save
```

```
# Descargar Net ISP balance
```

```
apt install unzip
```

```
wget https://github.com/lstein/Net-ISP-Balance/archive/master.zip
```

```
unzip master.zip
```

```
cd Net-ISP-Balance-master/
```

```
# Instalar las herramientas necesarias
```

```
apt-get install build-essential
```

```
apt install perl
```

```
apt-get install libmodule-build-perl
```

```
apt install cpanminus
```

```
cpanm Net::Netmask
```

```
# Ejecutar los archivos en el directorio Netispbalance
```

```
perl Build.PL
```

```
./Build installdeps
```

```
./Build test
```

```
./Build install
```

```
# configuración e implementación
```

```
apt install netmask
```

```
cpanm Net::netmask
```

```
reboot
```

```
# en la ubicación: "/etc/network/balance.conf", editar o descomentar los campos:
```

service	device	role	ping-ip	weight	gateway
ISP2	ens4	isp	203.0.113.1	1	203.0.113.1
ISP1	ens3	isp	192.168.101.1	1	192.168.1.1
LAN	ens5	lan			
#LAN2	eth2	lan			

```
mode=balanced
```

```
forwarding_group=:lan :isp
```

```
# Iniciar los servicios
```

```
load_balance.pl
```

```
comandos utiles para observar configuración
```

```
ip rule show
```

```
ip route show
```

```
ip tables -t nat -L
```

```
ip tables -L
```

```
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

```
# MONITOREO
```

```
apt install tcpdump
```

```
# tcpdump -i eth0 'port 8080' -w balance.pcap ///ejemplo
```

```
# -G tiempo en segundos, -W 1 número de capturas, -w ubicacion donde guardar el .pcap para poder abrirse con wireshark, -i indica interface
```

```
tcpdump -G 30 -W 1 -w /home/user/capturas/ens3.pcap -i ens3 & tcpdump -G 30 -W 1 -w /home/user/capturas/ens4.pcap -i ens4
```

```
tcpdump -G 30 -W 1 -w /home/user/capturas/ens3.pcap -i ens3 no arp no icmp & tcpdump -G 30 -W 1 -w /home/user/capturas/ens4.pcap -i ens4 no arp no icmp
```

```
# scp /home/user/capturas/ens3.pcap user@192.168.1.253:/home/user/Desktop //copiar desde el server al analizador escritorio
```

```
# scp username@remote:/file/to/send /where/to/put
```

```
scp root@192.168.1.254:/home/user/capturas/ens3.pcap /home/user/Desktop //copiar desde el host (no abrir ssh) terminal diferente
```

```
####SEGURIDAD
```

```
#GENERALES
```

```
App Armor
```

```
Complain mode: avisa
```

```
Enforce mode: bloquea apps
```

```
aa-status // chequear que los perfiles estén activados
```

```
Directorio de perfiles activos, revisión:
```

```
cd /etc/apparmor.d/
```

Se observan tipos de archivo

ls -l

vi ..... //para abrir

comando para historial:

sudo journalctl -fx

#Remover software inútil y servicios

#Para listar servicios escuchando de manera activa la red:

sudo lsof -i -P -n | grep -v ESTABLISHED

#Protocolos y Servicios sin encriptación

#Revisar:

nmap -v -sV -p- <address>

#PREVENTIVOS

#IDS Suricata

#Se instala el repositorio Open information Scurity Foundation's (OSIF):

add-apt-repository ppa:oisf/suricata-stable

#Se instala suricata con:

sudo apt install suricata

#Se habilita el servicio:

systemctl enable suricata.service

#Para configurar se detiene el servicio:

systemctl stop suricata.service

#Habilitar el ID de Community low (en caso de herramientas externas como zee, elasticserach):

#Ir a la línea 127 y configurar la línea "community-id: true"

```
nano /etc/suricata/suricata.yaml
```

#Para mostrar las interfaces a usar

```
ip -p -j route show default
```

```
nano /etc/suricata/suricata.yaml
```

#Modificar "af-packet" línea 614 "interface: eth0" como "ens3", la otra interface "ens4", se añade al final antes de la línea "-interface: default"

#Para configurar las reglas de suricata sin reiniciar el servicio activar el "Live rule reloading", en el yaml anterior al final añadir:

```
detect-engine:
```

```
- rule-reload: true
```

#Para exportar las rulesets externas:

```
suricata-update
```

#En caso de querer usar otras rulesets, revisar los proveedores de rulesets disponibles:

```
suricata-update list-sources
```

#Habilitarlas con:

```
suricata-update enable-source "nombre de ruleset"
```

```
suricata-update
```

#Para validar la configuración, `_t test mode -c ubicación -v info:`

```
suricata -T -c /etc/suricata/suricata.yaml -v
```

#Para correr suricata:

```
systemctl start suricata.service
```

```
systemctl status suricata.service
```

#Para examinar fast.log:

```
grep "ID" /var/log/suricata/fast.log
```

#Para examinar eve.log:

```
apt install jq //una vez
```

```
jq 'select(.alert .signature_id=="ID")' /var/log/suricata/eve.json
```

###Tiempo NTP

```
dpkg-reconfigure tzdata
```

#Instalar NTP

```
sudo apt-get install ntp
```

##Cambiar el servidor ntp a cercano:

Crear un archivo en /etc/cron.daily/ntpdate, que contenga las líneas:

```
#!/bin/sh
```

```
ntpdate ntp.ubuntu.com
```

#Para hacerlo ejecutable:

```
sudo chmod 755 /etc/cron.daily/ntpdate
```

#Para indicar los servers:

```
ntpdate ntp.ubuntu.com pool.ntp.org
```

##HARDENING SERVICIOS

#Firewall:

IP tables

#Probar:

sudo nmap -v -p- <dirección>

sudo nmap -v -sU -p- <dirección>

#Servidor SSH

Mínimo 64 bits de entropía, rsa=3072, para crear las claves, en la máquina cliente:

ssh-keygen -t rsa -b 4096

#Copiar la clave pública al servidor en la ubicación "~/.ssh/authorized\_keys":

ssh-copy-id lbadmin@192.168.101.254

ssh-copy-id carlos@192.168.101.254

#Checar permisos:

chmod 600 .ssh/authorized\_keys

###OPERACIONAL

#Manejo de Cuentas de usuario:

#Remover los permisos de lectura entre usuarios:

vi /etc/adduser.conf, modificar DIR\_MODE=750

#Comprobar permisos removidos:

ls -ld /home/username

#Crear usuarios y grupos:

Adduser carlos

Addgroup tecnicos

adduser carlos tecnicos

Adduser admin

adduser admin sudo

#Deshabilitar el usuario root, borrar contraseña y bloquear cuenta:

passwd -d root

passwd -l root

#Vencimiento de contraseñas:

sudo chage usuario

sudo chage -E 08/03/2024 -m 4 -M 90 -I 20 -W 14 carlos

#Revisión:

sudo chage -l <usuario>

#Desactivar acceso de root por SSH

En la ubicación “/etc/ssh/sshd\_config” agregar o verificar:

PermitRootLogin no

#Reiniciar el servicio:

Systemctl restart ssh.service

#Limitar ip's para ssh:



Para las redes permitidas, en la ubicación “/etc/hosts.allow” buscar o añadir “sshd :” y modificar:

```
sshd : 192.168.101.0/24
```

#\*se usa coma y escribir seguido para colocar varias redes

Para bloquear toda otra red, en la ubicación “/etc/hosts.deny” buscar o añadir “sshd :” y modificar:

```
sshd : ALL
```

#Desactivar fingerprints de antiguos accesos ssh

En la ubicación “/home/user/.ssh/” borrar las claves guardadas, en el cliente:

```
ls -l "/home/user/.ssh/
```

```
rm /home/user/.ssh/known_hosts
```

```
rm /home/user/.ssh/known_hosts.old
```

#Desactivar sesiones anteriores que podrían seguir activas, en el cliente:

```
who | grep <user>
```

```
pkill -f pts/##
```

#Desactivar acceso por contraseña, en el cliente:

```
vi /etc/ssh/ssh_config
```

#Editar la línea para que quede como:

```
PasswordAuthentication no
```

#Monitoreo de eventos local o mail:

Logwatch para revisión:

```
Apt update && apt upgrade && apt autoremove
```

```
apt install postfix
```

```
#, elegir "local", dar una dirección fqdn "real"
```

```
apt install logwatch
```

```
#Configuración:
```

```
/etc/logwatch/conf/logwatch.conf
```

```
#monitoreo:
```

```
logwatch --output stdout --format text --detail Low --range -7 days
```

## **Build.PL**

```
#!/usr/bin/perl
```

```
use strict;
```

```
use warnings;
```

```
use Module::Build;
```

```
use lib './lib';
```

```
use Net::ISP::Balance;
```

```
my $class = Module::Build->subclass(code=><<'EOF');
```

```
use File::Find 'find';
```

```
use File::Basename 'basename';
```

```
use lib './lib';
```

```
use Net::ISP::Balance;
```

```

sub ACTION_clean {

    my $self = shift;

    my $dir = $self->base_dir();

    system "cd $dir; make -s clean; cd $dir";

    $self->SUPER::ACTION_clean();

}

```

```

sub ACTION_install {

    my $self = shift;

    my $dir = $self->base_dir();

    $self->SUPER::ACTION_install();

    mkdir '/var/lib/lsm';

    my $path = $self->install_destination('script');

    my $install_etc = Net::ISP::Balance->install_etc;

    unlink("$install_etc/load_balance.pl");

    symlink("$path/load_balance.pl"=>"$install_etc/load_balance.pl");

    # fix permissions of balance.conf, to encourage root to edit

    chmod 0644,"$install_etc/balance.conf";

    # and rules files

    system "chmod u+w $install_etc/balance/firewall/*";

    system "chmod u+w $install_etc/balance/routes/*";

    system "chmod u+w $install_etc/balance/pre-run/*";

    system "chmod u+w $install_etc/balance/post-run/*";

}

```

```

sub process_lsm_files {
    my $self = shift;
    my $dir = $self->base_dir();
    $self->SUPER::ACTION_build();
    chomp(my $make = `which make 2>/dev/null`);
    chomp(my $nmake = `which nmake 2>/dev/null`);
    $make ||= $nmake;
    system "cd lsm; $make install; cd $dir";
}

```

```

sub process_etc_files {
    my $self = shift;
    find({no_chdir=>1,
        wanted => sub { return unless -f $File::Find::name;
            my $bn = basename($_);
            return if $bn =~ /~$/ || $bn =~ /^#/;
            (my $strip = $File::Find::name) =~ s!^\.etc/!!;
            my $install_etc = Net::ISP::Balance->install_etc;
            if (-e "$install_etc/$strip") {
                print STDERR "skip $install_etc/$strip: already installed\n";
                unlink ".bilib/$File::Find::name";
                return;
            }
            $self->copy_if_modified(from => $File::Find::name,
                to_dir => '.bilib');
        }},
    './etc'
}

```

```
    );  
}
```

EOF

```
my $build = $class->new(  
    module_name    => 'Net::ISP::Balance',  
    license        => 'perl',  
    dist_version_from => 'lib/Net/ISP/Balance.pm',  
    dist_abstract  => 'Load balance among two or more ISP connections',  
    dist_author    => 'Lincoln Stein <lincoln.stein@gmail.com>',  
    configure_requires => { 'Module::Build' => 0 },  
    requires       => {  
        'perl'      => '5.8.0',  
        'Net::Netmask' => 0,  
        'Pod::Usage' => 0,  
        'DB_File'   => 0,  
    },  
);  
  
my $install_etc = Net::ISP::Balance->install_etc;  
$build->add_build_element('etc');  
$build->add_build_element('lsm');  
$build->install_path('etc'    => $install_etc);  
$build->config_data('lsm_path'=>$build->install_destination('script').'/foolsm');  
$build->create_build_script;  
exit 0;
```