

INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

ING. EN SISTEMAS COMPUTACIONALES



PRÁCTICA 8: “BALANCEO DE CARGAS”

22 - MAYO - 2024

ASIGNATURA: ADMINISTRACIÓN DE SERVICIOS EN RED

PROFESOR: TENORIO MARRON MARCO ANTONIO

GRUPO: 7CM3

EQUIPO “ROUTERS Y RISAS” :

- ALVARADO ROMERO LUIS MANUEL
- ROMERO HERNÁNDEZ OSCAR DAVID
- OLMOS VERDIN DIEGO

PRACTICA 8 Balanceo de cargas

Introducción

El balanceo de cargas es una técnica esencial en las redes de computadoras que distribuye de manera eficiente el tráfico de red y las solicitudes de servicios entre varios servidores o recursos. Este mecanismo es fundamental para garantizar la disponibilidad, escalabilidad y rendimiento óptimo de aplicaciones y servicios en entornos de red, especialmente en la era actual donde la demanda de servicios en línea es cada vez mayor.

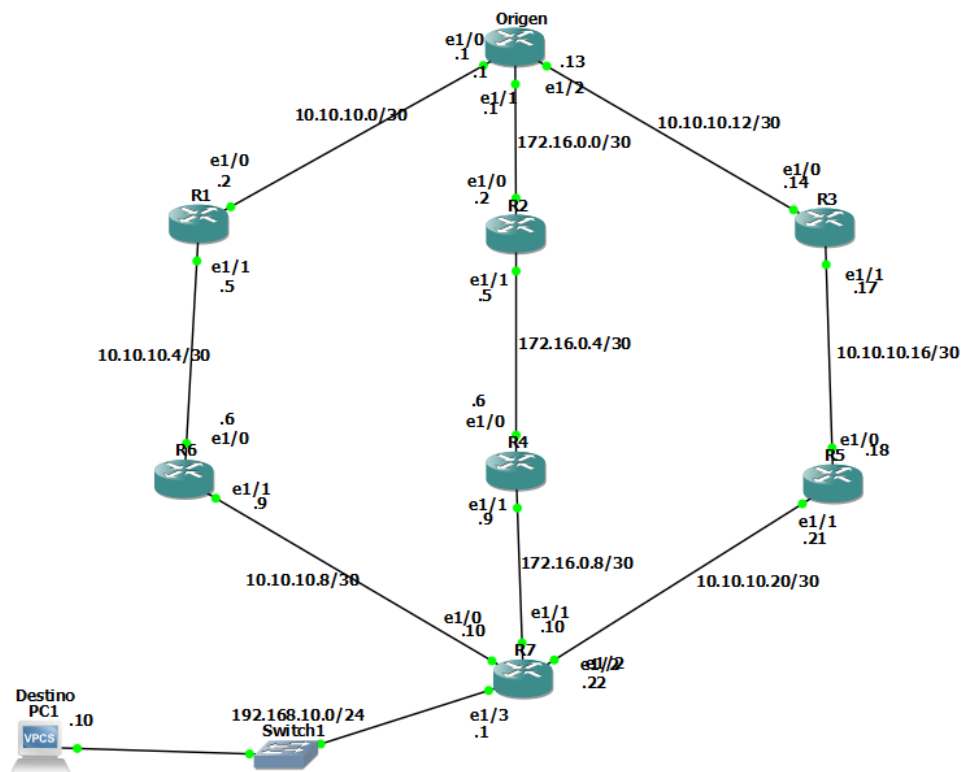
Unos de los factores clave en los que ayuda el balanceo de cargas son:

Disponibilidad y Redundancia: El balanceo de cargas asegura que los servicios permanezcan disponibles incluso si uno o más servidores fallan. Al distribuir el tráfico entre múltiples servidores, el sistema puede redirigir automáticamente las solicitudes a servidores funcionales, mejorando la tolerancia a fallos.

Escalabilidad: Permite que los sistemas crezcan de manera horizontal al añadir más servidores al clúster de balanceo de carga. Esto es crucial para manejar incrementos en la demanda de usuarios sin degradar el rendimiento del servicio.

Rendimiento: Al distribuir las solicitudes de manera equitativa entre varios servidores, el balanceo de cargas evita la sobrecarga de cualquier servidor individual, optimizando así el tiempo de respuesta y el rendimiento general del sistema.

Topología Propuesta

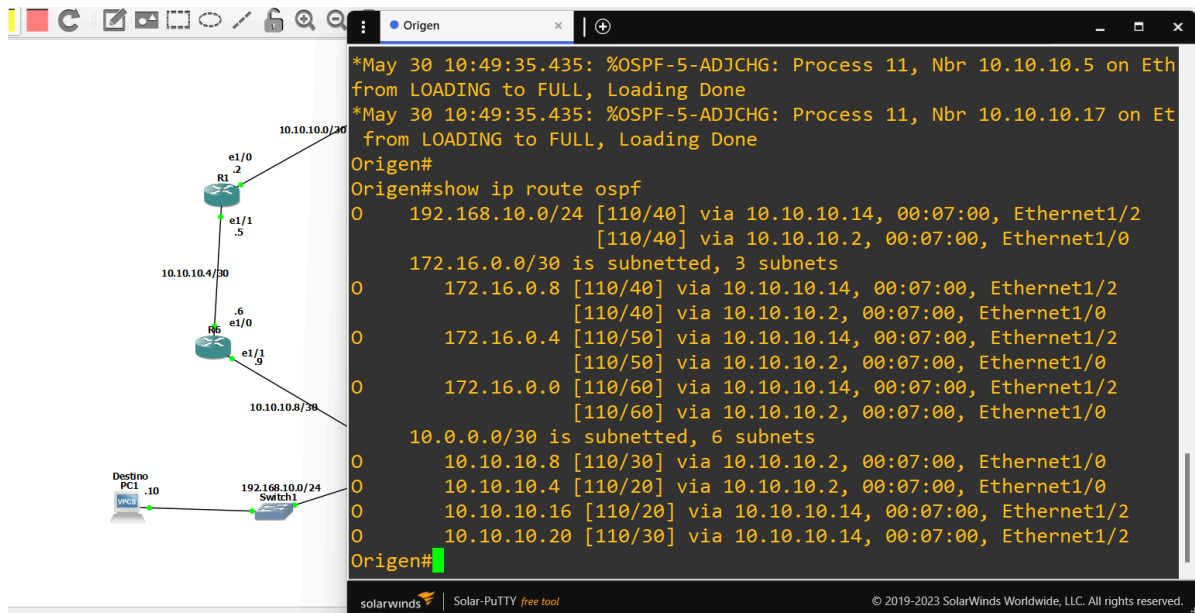


Configuración de dispositivos:

Se configura la topología con las redes propuestas y con el protocolo OSPF con todos los routers en la misma área

```
show ip route ospf
```

Mostramos las rutas creadas con el siguiente comando:



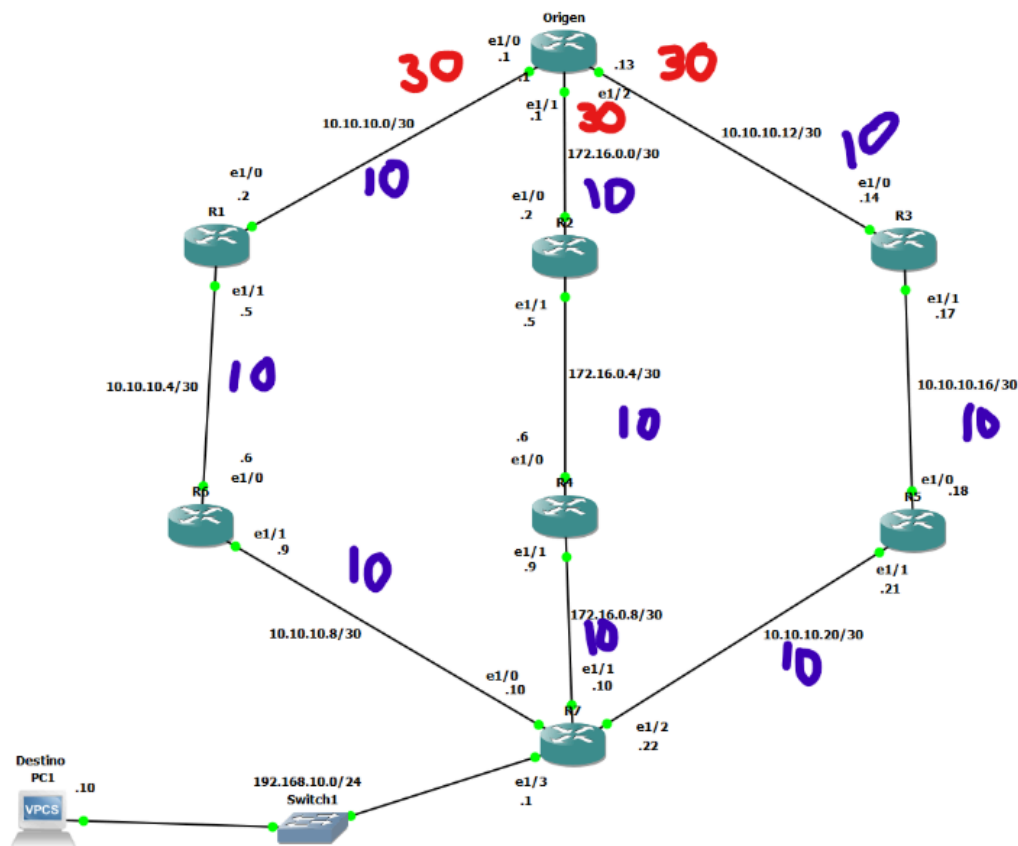
The screenshot shows a network diagram on the left and a terminal window on the right. The terminal window displays the following commands and output:

```

*May 30 10:49:35.435: %OSPF-5-ADJCHG: Process 11, Nbr 10.10.10.5 on Eth
from LOADING to FULL, Loading Done
*May 30 10:49:35.435: %OSPF-5-ADJCHG: Process 11, Nbr 10.10.10.17 on Et
from LOADING to FULL, Loading Done
Origen#
Origen#show ip route ospf
0    192.168.10.0/24 [110/40] via 10.10.10.14, 00:07:00, Ethernet1/2
    [110/40] via 10.10.10.2, 00:07:00, Ethernet1/0
    172.16.0.0/30 is subnetted, 3 subnets
0    172.16.0.8 [110/40] via 10.10.10.14, 00:07:00, Ethernet1/2
    [110/40] via 10.10.10.2, 00:07:00, Ethernet1/0
0    172.16.0.4 [110/50] via 10.10.10.14, 00:07:00, Ethernet1/2
    [110/50] via 10.10.10.2, 00:07:00, Ethernet1/0
0    172.16.0.0 [110/60] via 10.10.10.14, 00:07:00, Ethernet1/2
    [110/60] via 10.10.10.2, 00:07:00, Ethernet1/0
    10.0.0.0/30 is subnetted, 6 subnets
0    10.10.10.8 [110/30] via 10.10.10.2, 00:07:00, Ethernet1/0
0    10.10.10.4 [110/20] via 10.10.10.2, 00:07:00, Ethernet1/0
0    10.10.10.16 [110/20] via 10.10.10.14, 00:07:00, Ethernet1/2
0    10.10.10.20 [110/30] via 10.10.10.14, 00:07:00, Ethernet1/2
Origen#

```

Cambiamos los costos de las interfaces:

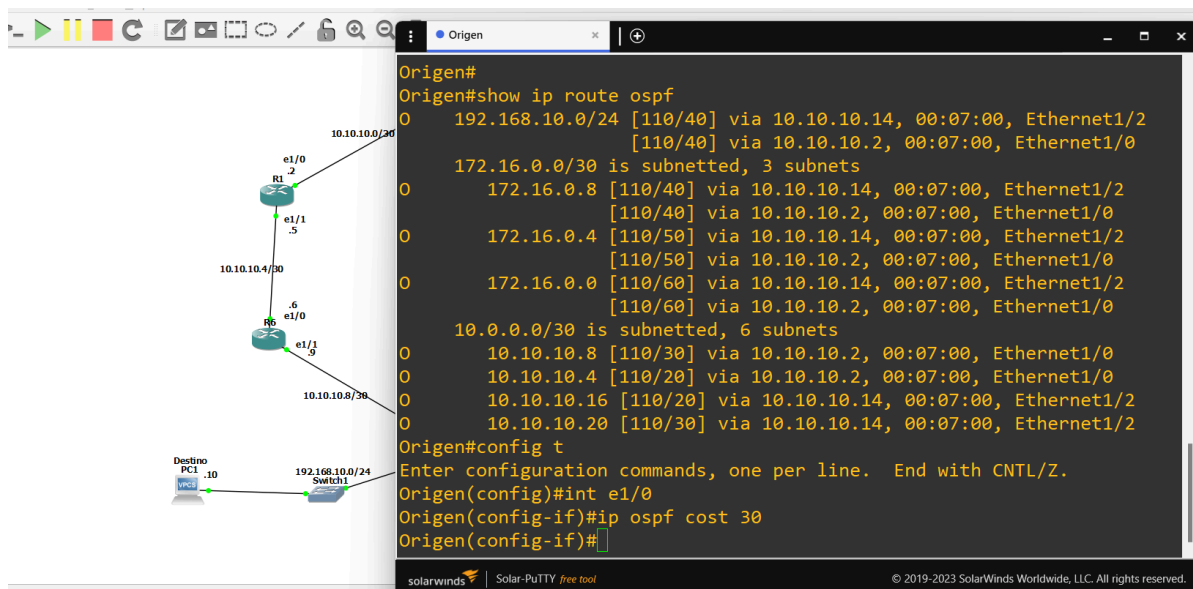


Para cambiar los costos de las interfaces ingresamos los siguientes comandos:

```

Origen# config t
Origen(config)# inter e1/0
Origen(config-if)# ip ospf cost 10

```



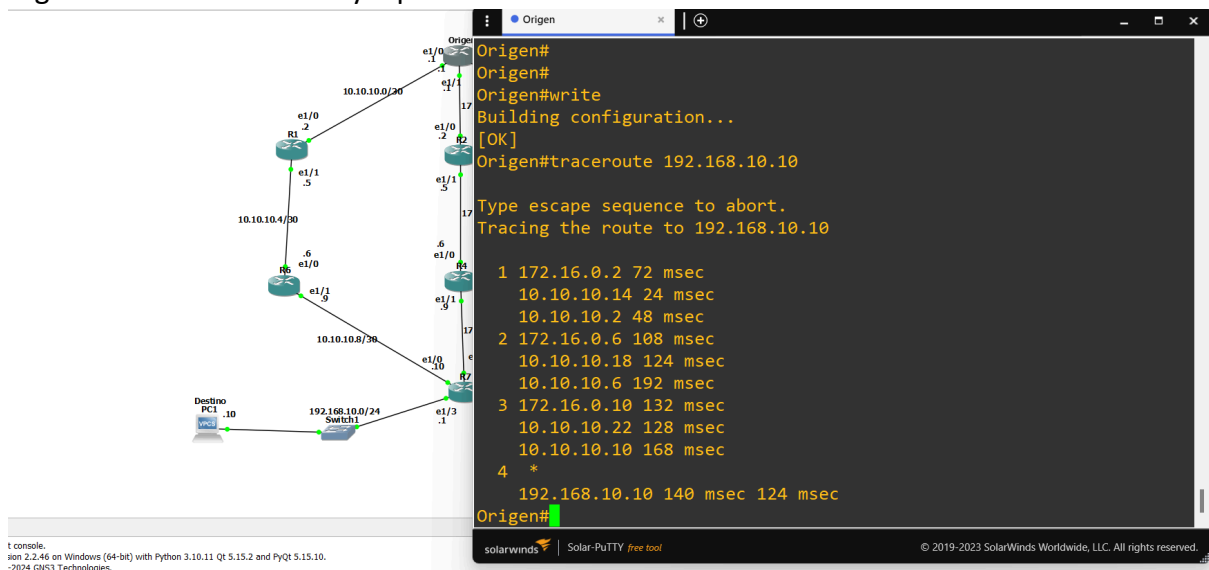
The network diagram shows a topology with three paths from 'Origen' (Router R1) to 'Destino' (PC1). Path 1 goes through R1 (e1/0) to R2 (e1/1) to R3 (e1/0) to PC1. Path 2 goes through R1 (e1/0) to R4 (e1/1) to R5 (e1/0) to PC1. Path 3 goes through R1 (e1/0) to R6 (e1/1) to R7 (e1/0) to PC1. All links have a cost of 10. The terminal output shows the OSPF route table for 192.168.10.0/24, listing three equal-cost paths.

```

Origen#
Origen#show ip route ospf
0 192.168.10.0/24 [110/40] via 10.10.10.14, 00:07:00, Ethernet1/2
0 192.168.10.0/24 [110/40] via 10.10.10.2, 00:07:00, Ethernet1/0
0 172.16.0.0/30 is subnetted, 3 subnets
0 172.16.0.8 [110/40] via 10.10.10.14, 00:07:00, Ethernet1/2
0 172.16.0.4 [110/50] via 10.10.10.14, 00:07:00, Ethernet1/2
0 172.16.0.0 [110/50] via 10.10.10.2, 00:07:00, Ethernet1/0
0 172.16.0.0 [110/60] via 10.10.10.14, 00:07:00, Ethernet1/2
0 172.16.0.0 [110/60] via 10.10.10.2, 00:07:00, Ethernet1/0
0 10.0.0.0/30 is subnetted, 6 subnets
0 10.10.10.8 [110/30] via 10.10.10.2, 00:07:00, Ethernet1/0
0 10.10.10.4 [110/20] via 10.10.10.2, 00:07:00, Ethernet1/0
0 10.10.10.16 [110/20] via 10.10.10.14, 00:07:00, Ethernet1/2
0 10.10.10.20 [110/30] via 10.10.10.14, 00:07:00, Ethernet1/2
Origen#config t
Enter configuration commands, one per line. End with CNTL/Z.
Origen(config)#int e1/0
Origen(config-if)#ip ospf cost 30
Origen(config-if)#

```

En esta captura podemos ver que al hacer un ping a la pc desde el router origen podemos llegar desde los 3 caminos ya que todas las rutas tienen el mismo costo



The network diagram shows the same topology as the previous image. The terminal output shows the configuration of the cost for the e1/0 interface to 30, followed by a traceroute to 192.168.10.10. The traceroute shows three distinct paths, each with four hops, confirming that traffic can reach the destination via multiple equal-cost paths.

```

Origen#
Origen#
Origen#write
Building configuration...
[OK]
Origen#traceroute 192.168.10.10
Type escape sequence to abort.
Tracing the route to 192.168.10.10
 0 172.16.0.2 72 msec
 1 10.10.10.14 24 msec
 2 10.10.10.2 48 msec
 3 172.16.0.6 108 msec
 4 10.10.10.18 124 msec
 5 10.10.10.6 192 msec
 6 172.16.0.10 132 msec
 7 10.10.10.22 128 msec
 8 10.10.10.10 168 msec
 9 *
10 192.168.10.10 140 msec 124 msec
Origen#

```

Three Wireshark packet capture windows are shown, illustrating network traffic. The top window shows OSPF Hello packets and ICMP Echo requests/replies. The middle window shows ICMP Echo requests/replies. The bottom window shows ICMP Echo requests/replies. A terminal window on the right shows the output of a ping command.

Terminal Output:

```

10.10.10.10 168 msec
4 *
192.168.10.10 140 msec 124 msec
Origen#ping 192.168.10.10

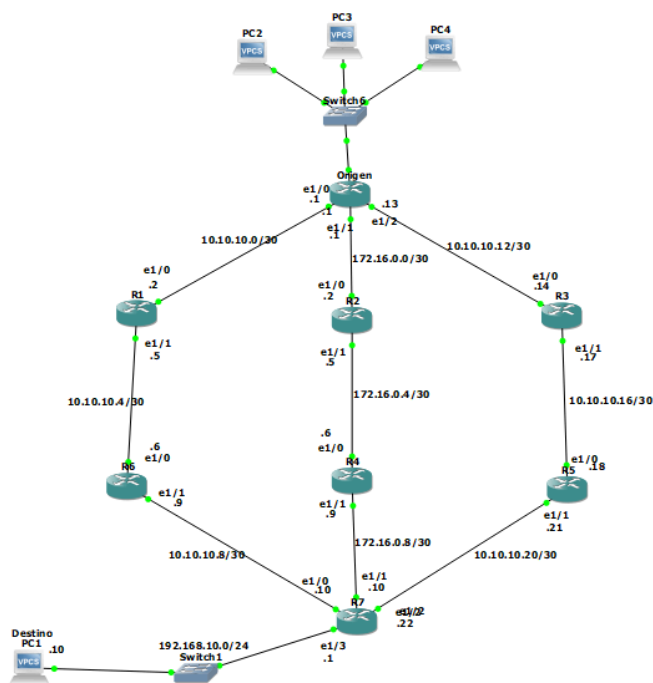
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.10.10, timeout is 2 s
econds:
!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 140/
427/1232 ms
Origen#

```

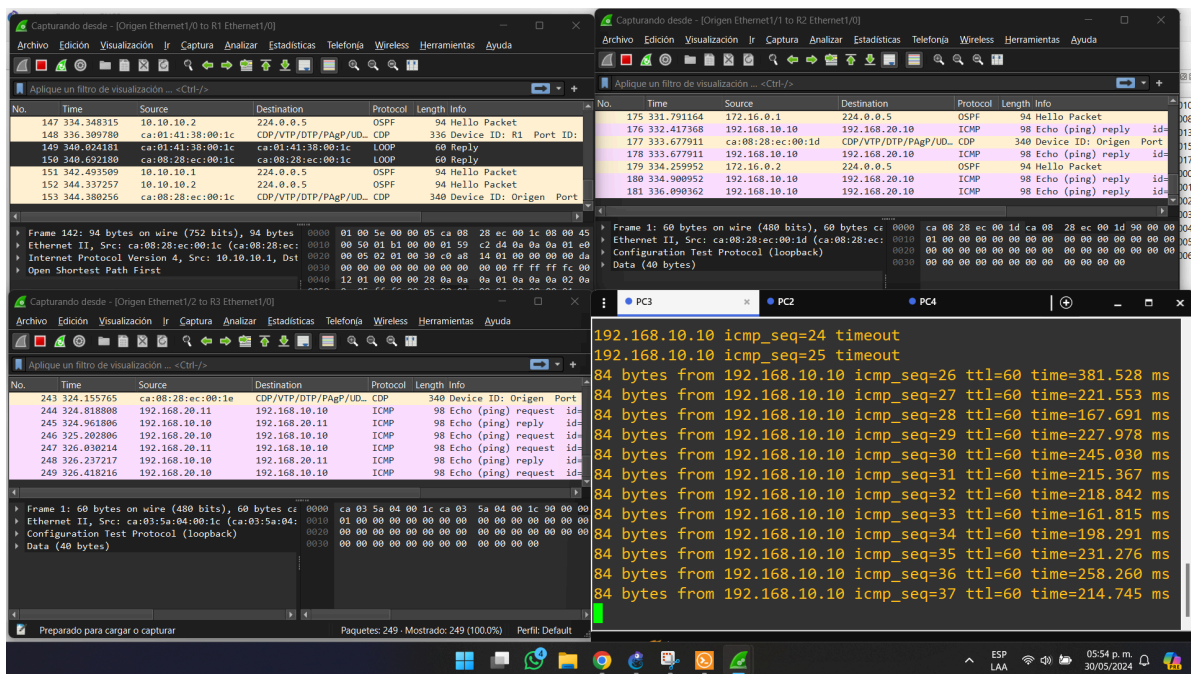
De esta manera podemos visualizar como los paquetes de nuestro ping se distribuyen entre los primeros routers

Ejercicio de balanceo de cargas:

Agregamos 3 PC en la parte del router origen para hacer un ping extendido hacia la PC destino y analizar el tráfico de los routers.



vemos cómo al realizar el ping extendido los paquetes se distribuyen entre las 3 redes



Conclusión

La práctica de balanceo de cargas en routers utilizando el simulador GNS3 nos permitió comprender en profundidad cómo funciona esta técnica esencial en redes de computadoras. Mediante la configuración de múltiples routers y la distribución del tráfico entre ellos, logramos observar de primera mano los beneficios del balanceo de cargas, como la mejora en la distribución del tráfico y la optimización del rendimiento de la red.

El uso de Wireshark para analizar los paquetes de datos fue fundamental para entender mejor los detalles técnicos y el comportamiento del tráfico de red bajo un escenario de balanceo de cargas. Al capturar y examinar las tramas, pudimos ver cómo las solicitudes se distribuían entre los diferentes routers, lo que proporcionó una visión clara de la eficiencia y efectividad del balanceo de cargas en tiempo real.

Uno de los retos significativos que enfrentamos durante la práctica fue la disponibilidad de recursos. La simulación en GNS3 y el análisis exhaustivo de tramas con Wireshark son actividades intensivas en términos de procesamiento y memoria. A medida que utilizamos todos los recursos de los routers y analizamos un volumen considerable de paquetes, notamos que el rendimiento de la computadora se veía afectado, resultando en un calentamiento notable del sistema. Este desafío subraya la importancia de contar con hardware adecuado y suficiente capacidad de recursos para llevar a cabo simulaciones complejas y análisis detallados sin comprometer la funcionalidad.