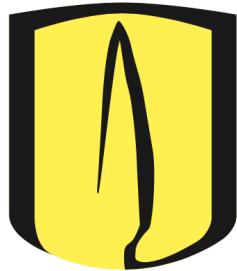


Universidad de los Andes

Departamento de Ingeniería de Sistemas



**Laboratorio #4: Protocolo de enrutamiento dinámicos
con redes IPv4 e IPv6**

ISIS3204 - Infraestructura de Comunicaciones

Grupo 3:

Juan Esteban Quiroga - 202013216

Juan Manuel Rodriguez - 202013372

Andres Felipe Ortiz - 201727662

2025-10

Contents

Introducción	3
1 Mapas de cada topología	3
2 Protocolo RIP (Routing Information Protocol)	5
2.1 Análisis del protocolo de enrutamiento	5
2.2 Prueba de conectividad	6
3 Protocolo OSPF (Open Shortest Path First)	7
3.1 Análisis del protocolo de enrutamiento	7
3.2 Prueba de conectividad	9
4 Protocolo BGP (Border Gateway Protocol)	10
4.1 Pruebas de Conectividad	10
4.2 Conclusión del BGP	12
5 Protocolo RIPng (Routing Information Protocol next generation)	12
5.1 Análisis del protocolo de enrutamiento	12
5.2 Prueba de conectividad	14
5.2.1 Análisis del Flujo de Enrutamiento y Métrica	15
5.2.2 Análisis del Límite de Saltos (Hop Limit)	15
6 Protocolo OSPFv3 (Open Shortest Path First version 3)	15
6.1 Análisis del protocolo de enrutamiento	16
6.2 Prueba de conectividad	17
7 Evidencia Trabajo cada uno de los estudiantes	18

Introducción

El presente laboratorio tiene como objetivo principal **comprender y aplicar los protocolos de enrutamiento dinámico RIP y OSPF en entornos IPv4 e IPv6**, destacando sus diferencias estructurales y operativas. A través de la configuración práctica de routers Cisco en el simulador *Cisco Packet Tracer*, se busca afianzar los conceptos teóricos sobre el funcionamiento de los protocolos de vector distancia y de estado de enlace, así como el proceso de intercambio de información de enrutamiento y la construcción de tablas de rutas en topologías multi-router.

Durante el desarrollo de la práctica se implementan diferentes escenarios de red que integran el direccionamiento IP, la configuración de interfaces, y la activación de protocolos de enrutamiento dinámico. Esto permite observar cómo los routers actualizan sus tablas de enrutamiento de manera automática y cómo se comporta la red ante variaciones en la topología o fallos en los enlaces.

El laboratorio permite:

- Diferenciar los principios de funcionamiento de **RIP (Routing Information Protocol)** y **OSPF (Open Shortest Path First)**, tanto en su versión IPv4 como IPv6.
- Configurar y verificar el intercambio de rutas dinámicas entre múltiples routers, observando métricas como el conteo de saltos y el costo por ancho de banda.
- Comprender el proceso de **asignación jerárquica de direcciones IP**, el uso de máscaras de subred y prefijos en IPv6, así como la importancia del direccionamiento correcto en la conectividad de red.
- Analizar el impacto de los protocolos de enrutamiento en la convergencia de la red y en la eficiencia del encaminamiento de paquetes.

De esta forma, la práctica integra los conceptos fundamentales del **nivel de red del modelo OSI**, brindando una experiencia completa que abarca el diseño, configuración y validación de una infraestructura de comunicación moderna y funcional.

1 Mapas de cada topología

Topología #1 - RIPv1 y OSPF (IPv4)

Esta topología conecta tres routers (R1, R2 y R3) y dos PCs ubicadas en redes finales independientes. Se utilizan las redes **192.168.10.0**, **192.168.20.0**, **192.168.30.0** y **192.168.40.0**, junto con interfaces loopback anunciadadas mediante enrutamiento dinámico. El objetivo es configurar y comprobar el funcionamiento de RIPv1 y OSPF, permitiendo que las estaciones finales intercambien tráfico a través de rutas aprendidas automáticamente.

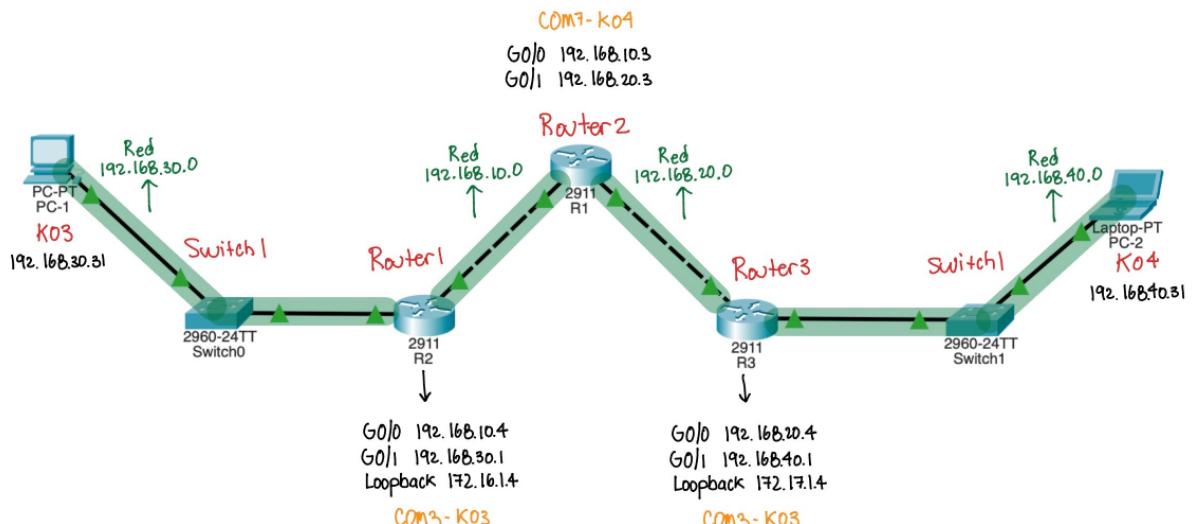


Figure 1: Topología #1

Topología #2 - BGP (IPv4)

En esta topología los routers R7, R8 y R9 conforman una estructura tipo “Y” que simula diferentes sistemas autónomos interconectados. Cada router anuncia redes propias como **192.168.10.0**, **192.168.40.0**, **192.168.50.0** y **192.168.60.0**, y cada subred final contiene una PC. Su propósito es implementar BGP para el intercambio de prefijos entre los routers y validar la comunicación extremo a extremo mediante rutas basadas en políticas.

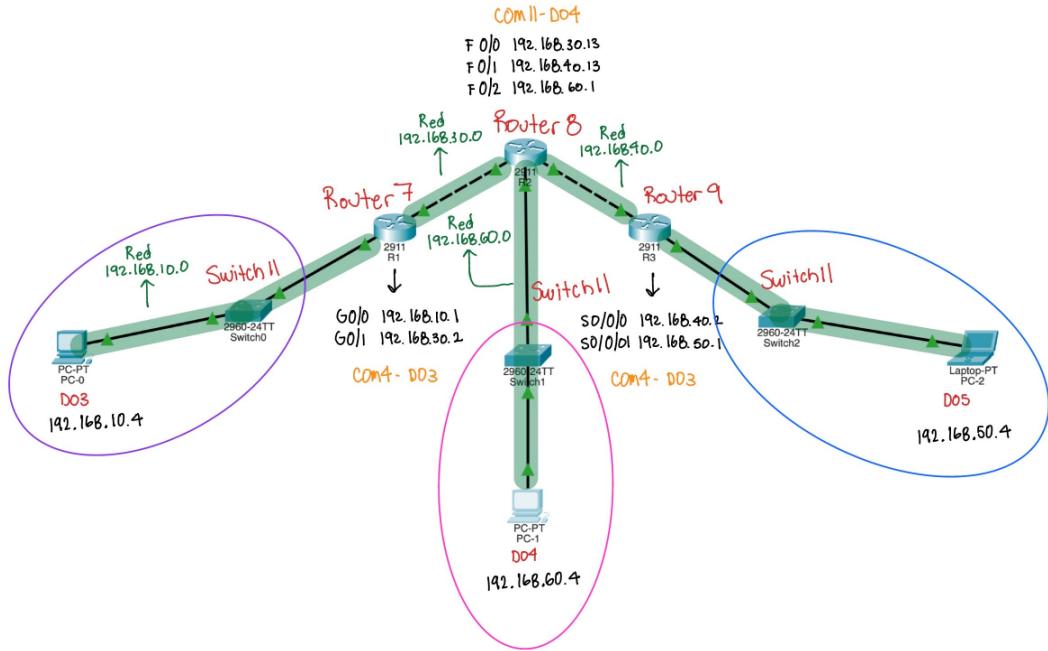


Figure 2: Topología #2

Topología #3 - RIPng y OSPFv3 (IPv6)

La tercera topología conecta tres routers (R4, R5 y R6) mediante enlaces IPv6 con prefijos /64, junto con dos PCs en redes finales independientes. Utiliza los bloques **2001:ABCD:1435::/64** y **2002:ABCD:1435::/64** según lo definido por el laboratorio. Aquí se configuran RIPng y OSPFv3 para observar la distribución dinámica de rutas IPv6 y validar la conectividad entre los equipos.

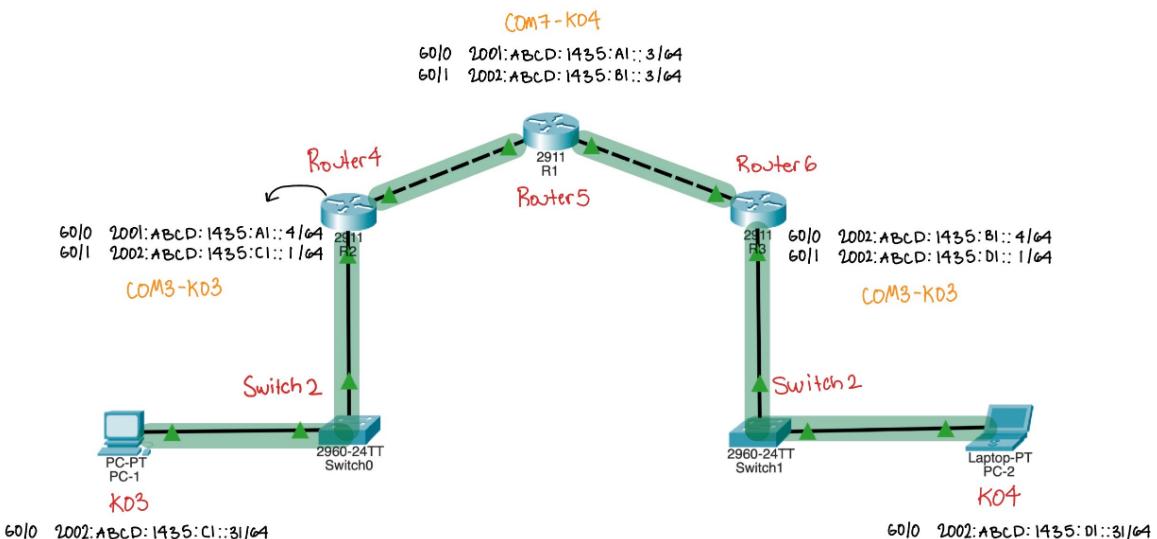


Figure 3: Topología #3

2 Protocolo RIP (Routing Information Protocol)

El protocolo **RIPv1** es un protocolo de enrutamiento dinámico con clase que utiliza el algoritmo belman-fort para encontrar el camino con menor cantidad de saltos; a esto le llamamos vector distancia. Vale la pena mencionar que este protocolo (RIP en general) no se utiliza para más de 15 saltos, esto por el algoritmo utilizado, al pasar de 15 saltos se considerara inalcanzable, por lo que **RIP** solo se utiliza para redes pequeñas.

2.1 Análisis del protocolo de enrutamiento

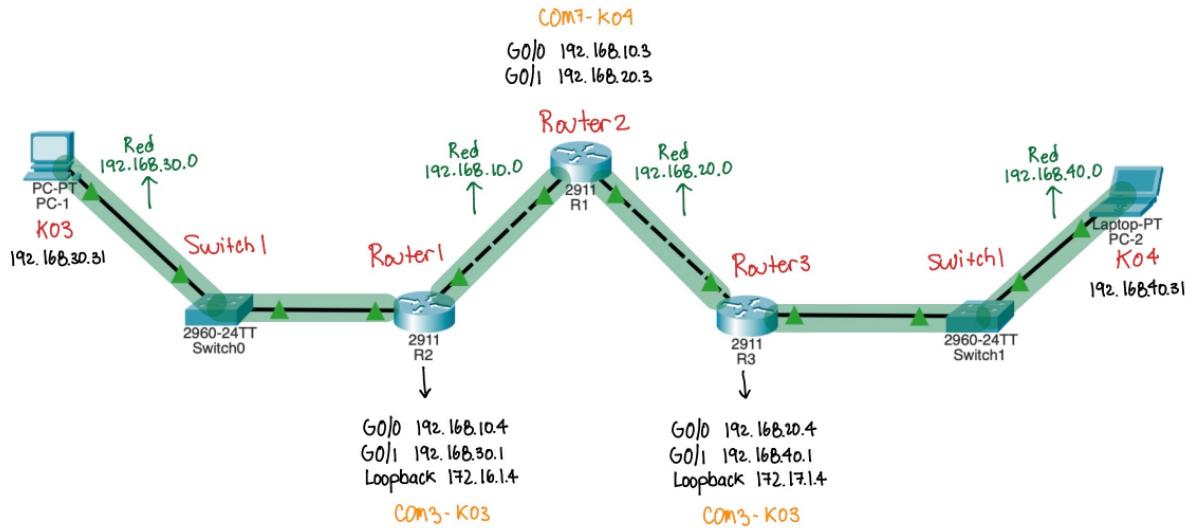


Figure 4: Mapa topología numero 1.

Para empezar, usaremos la topología 1, donde, como se ve en la imagen, tendremos 3 routers (número de saltos menor a 15), y las capturas de los paquetes que veremos a continuación corresponderán a las capturas del computador marcado como *PC-2*. A continuación vemos los paquetes que capturados con el protocolo **RIP**.

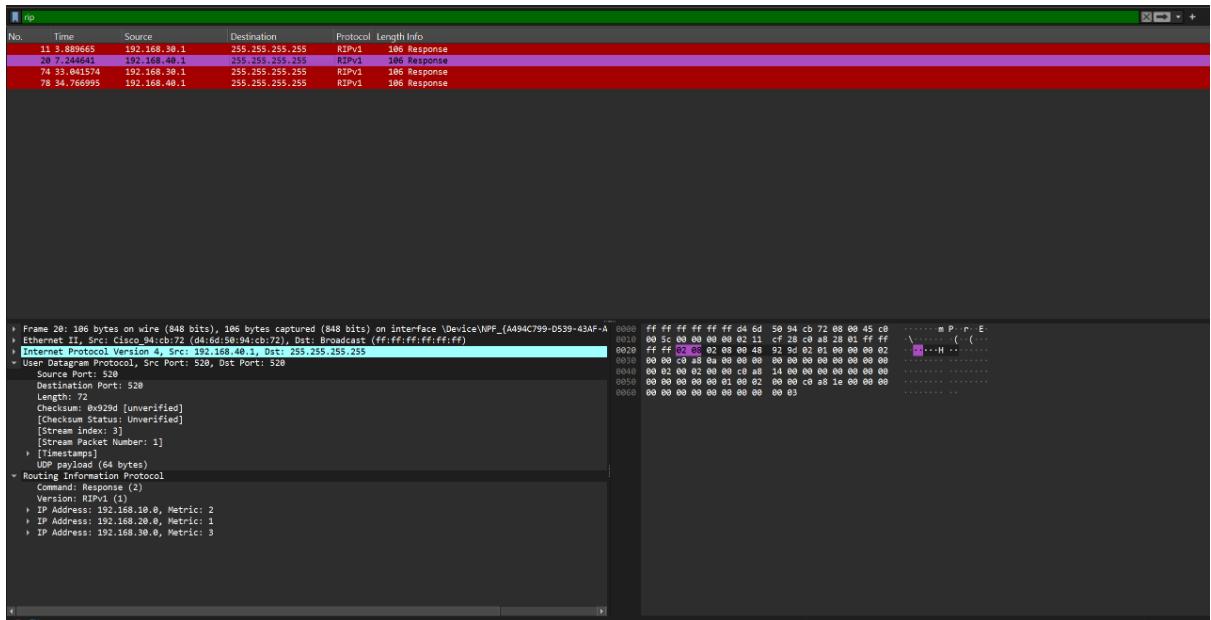


Figure 5: Capturas del protocolo RIPv1.

En esta captura podemos ver varias características importantes del protocolo **RIP**, una de ellas es el uso del protocolo de la capa de transporte *UDP* con el puerto 520. También, podemos ver el uso de

la dirección de destino `255.255.255.255` mostrando que en efecto estamos usando **RIPv1** pues en esta versión del protocolo se utiliza una dirección broadcast, a diferencia de la versión 2 del protocolo que utiliza una dirección de multicast. Ahora bien, y más relevante aun, podemos ver las rutas que está tomando el protocolo, junto con su número de saltos, como lo podemos ver a continuación en la siguiente tabla e imagen:

```
▼ Routing Information Protocol
  Command: Response (2)
  Version: RIPv1 (1)
  ▶ IP Address: 192.168.10.0, Metric: 2
  ▶ IP Address: 192.168.20.0, Metric: 1
  ▶ IP Address: 192.168.30.0, Metric: 3
```

Figure 6: Rutas del protocolo RIPv1.

Tabla 1: Rutas anunciadas en el mensaje RIPv1

Dirección IP	Métrica	Interpretación
192.168.10.0	2	Red alcanzable a 2 saltos
192.168.20.0	2	Red alcanzable a 2 saltos
192.168.30.0	3	Red alcanzable a 3 saltos

Como lo mencionábamos anteriormente, vemos la cantidad de saltos que necesita desde cierto router para alcanzar las diferentes redes que hay habilitadas, además, muestra como cada uno de los routers "anuncia" a los demás componentes de la red su tabla de enrutamiento para la llegada a los diferentes segmentos de la red.

2.2 Prueba de conectividad

Ahora vamos a ver la evidencia del correcto funcionamiento del enrutamiento a través de los paquetes icmp, usando el comando `ping` entre los computadores *PC-0*, ip: `192.168.40.31` y *PC-2*, ip: `192.168.30.31`

```
No. Time Source Destination Protocol Length Info
  28 12.115529 192.168.40.31 192.168.30.31 ICMP 80 Echo (ping) request id=0x0001, seq=9/2984, ttl=128 (reply in 29)
  29 12.1155309 192.168.40.31 192.168.30.31 ICMP 74 Echo (ping) reply id=0x0001, seq=9/2984, ttl=128 (request in 28)
-> 31 13.139846 192.168.40.31 192.168.30.31 ICMP 74 Echo (ping) request id=0x0001, seq=10/2986, ttl=128 (reply in 32)
  32 13.135779 192.168.30.31 192.168.40.31 ICMP 74 Echo (ping) reply id=0x0001, seq=10/2986, ttl=125 (request in 31)
  33 14.136186 192.168.40.31 192.168.30.31 ICMP 74 Echo (ping) request id=0x0001, seq=11/2816, ttl=128 (reply in 34)
  34 14.136187 192.168.30.31 192.168.40.31 ICMP 74 Echo (ping) reply id=0x0001, seq=11/2816, ttl=125 (request in 33)
  35 15.144872 192.168.40.31 192.168.30.31 ICMP 74 Echo (ping) request id=0x0001, seq=12/2898, ttl=128 (reply in 37)
  37 15.144894 192.168.30.31 192.168.40.31 ICMP 74 Echo (ping) reply id=0x0001, seq=12/2892, ttl=125 (request in 36)
  64 38.287538 192.168.30.31 192.168.40.31 ICMP 74 Echo (ping) request id=0x0001, seq=129/12549, ttl=125 (reply in 65)
  65 38.287878 192.168.40.31 192.168.30.31 ICMP 74 Echo (ping) reply id=0x0001, seq=129/12549, ttl=128 (request in 64)
  66 38.287879 192.168.30.31 192.168.40.31 ICMP 74 Echo (ping) request id=0x0001, seq=130/12549, ttl=125 (reply in 66)
  69 31.318811 192.168.40.31 192.168.30.31 ICMP 74 Echo (ping) reply id=0x0001, seq=130/12549, ttl=128 (request in 68)
  70 31.318812 192.168.30.31 192.168.40.31 ICMP 74 Echo (ping) request id=0x0001, seq=131/12549, ttl=125 (reply in 69)
  71 32.318914 192.168.30.31 192.168.40.31 ICMP 74 Echo (ping) request id=0x0001, seq=131/12549, ttl=125 (reply in 72)
  72 32.319386 192.168.40.31 192.168.30.31 ICMP 74 Echo (ping) reply id=0x0001, seq=131/12549, ttl=128 (request in 71)
  75 33.334915 192.168.30.31 192.168.40.31 ICMP 74 Echo (ping) request id=0x0001, seq=132/13317, ttl=125 (reply in 76)
  76 33.335238 192.168.40.31 192.168.30.31 ICMP 74 Echo (ping) reply id=0x0001, seq=132/13317, ttl=128 (request in 75)

> Frame 31: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 'Device\NPF_{4d94C799-D539-43AF-A0C...' 0:0000 d4 8d 50 94 cb 72 86 bf b8 a2 7c df 88 80 45 00 mP...r...[...]
  Ethernet II, Src: ASUSTekCOMPU.a2:7c:df (08:bf:b8:a2:7c:df), Dst: Cisco_94:c8:72 (d4:8d:50:94:c8:72)
  Internet Protocol Version 4, Src: 192.168.40.31, Dst: 192.168.30.31
  Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
      Checksum: 0x4d51 [correct]
      [Checksum Status: Good]
      Identifier (BE): 1 (0x0001)
      Identifier (LE): 256 (0x0100)
      Sequence Number (BE): 10 (0x000a)
      Sequence Number (LE): 256 (0x000a)
      Data (32 bytes)
        Data (32 bytes)

> Frame 31: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 'Device\NPF_{4d94C799-D539-43AF-A0C...' 0:0010 00 5c e2 1a 00 00 80 01 00 00 c8 a8 28 1f c9 a8 < ...> [...] E...
  0:0010 00 5c e2 1a 00 00 80 01 00 00 c8 a8 28 1f c9 a8 < ...> [...] E...
  0:0020 1e 1f 00 4d 51 00 01 00 0a 61 62 63 64 65 66 ... MQ...abcde...
  0:0030 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 ghijklmn opqrstuv
  0:0040 77 61 62 63 64 65 66 67 68 69 wabcdegf hiijklmn opqrstuv

> Frame 31: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 'Device\NPF_{4d94C799-D539-43AF-A0C...' 0:0000 d4 8d 50 94 cb 72 86 bf b8 a2 7c df 88 80 45 00 mP...r...[...]
  Ethernet II, Src: ASUSTekCOMPU.a2:7c:df (08:bf:b8:a2:7c:df), Dst: Cisco_94:c8:72 (d4:8d:50:94:c8:72)
  Internet Protocol Version 4, Src: 192.168.40.31, Dst: 192.168.30.31
  Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
      Checksum: 0x4d51 [correct]
      [Checksum Status: Good]
      Identifier (BE): 1 (0x0001)
      Identifier (LE): 256 (0x0100)
      Sequence Number (BE): 10 (0x000a)
      Sequence Number (LE): 256 (0x000a)
      Data (32 bytes)
        Data (32 bytes)
```

Figure 7: Prueba de conectividad protocolo RIPv1.

En la captura podemos observar los mensajes del protocolo **ICMP** que se generan al ejecutar el comando `ping` desde el *PC-2* (192.168.40.31) hacia el *PC-0* (192.168.30.31). En este caso, se aprecia

el intercambio correcto de solicitudes y respuestas de eco (*Echo Request* y *Echo Reply*), lo que evidencia que el proceso de enrutamiento entre las redes funciona de manera adecuada.

Cada paquete ICMP de tipo 8 (solicitud de eco) enviado desde el equipo origen recibe su correspondiente paquete de tipo 0 (respuesta de eco) desde el destino, confirmando que los datos están llegando correctamente y regresando sin pérdida alguna. Los valores de *Checksum* aparecen como correctos, lo que indica que no existen errores de transmisión en los mensajes capturados.

El tiempo de respuesta entre cada solicitud y su respectiva respuesta es aproximadamente de **0.021 milisegundos**, lo cual es un valor bajo y característico de una red local (LAN) bien configurada. Este tiempo refleja que la comunicación entre ambos extremos es eficiente y que no hay retardos significativos durante el tránsito de los paquetes a través de los routers.

Adicionalmente, el valor del campo *TTL* (*Time To Live*) en los mensajes capturados permite inferir que existe al menos un salto intermedio entre los dos equipos, lo que concuerda con la topología establecida y confirma que los routers están realizando el reenvío de paquetes correctamente. De igual forma, se identifican las direcciones físicas de origen y destino (08:bf:b8:e2:7c:df y d4:6d:50:94:cb:72), correspondientes a los dispositivos involucrados en la comunicación.

3 Protocolo OSPF (Open Shortest Path First)

El protocolo **OSPF** (Open Shortest Path First) es un protocolo de enrutamiento de estado de enlace (*Link-State*) que opera en la Capa 3 del modelo OSI. Utiliza el algoritmo de Dijkstra para calcular el camino más corto hacia todas las redes. A diferencia de RIP, OSPF es altamente escalable y adecuado para **redes medianas y grandes**, ya que no tiene limitación de saltos y maneja la jerarquía de red mediante el concepto de **Áreas**, lo que facilita una convergencia rápida y previene bucles de enrutamiento.

3.1 Análisis del protocolo de enrutamiento

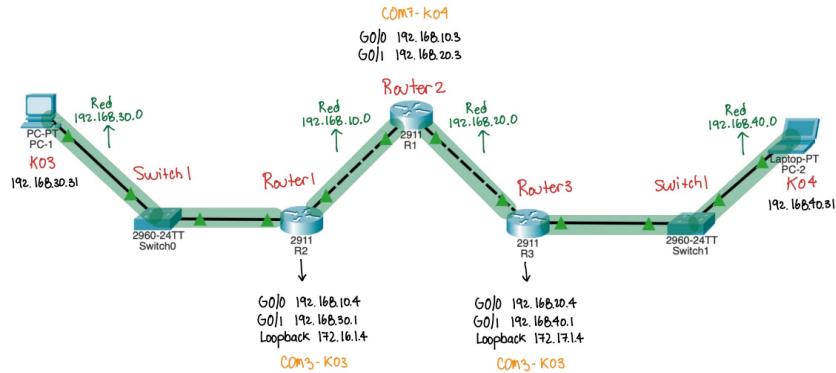


Figure 8: Mapa topología de referencia (asumida para el análisis).

Para el análisis de OSPF, nos centraremos en los paquetes de mantenimiento y descubrimiento de vecinos, específicamente los paquetes **Hello**. Las capturas de los paquetes a continuación corresponden al tráfico de OSPF en la red, presumiblemente capturado desde un *PC-2* o dispositivo en un segmento.

No.	Time	Source	Destination	Protocol	Length Info
19	7.305353	192.168.30.1	224.0.0.5	OSPF	90 Hello Packet
23	8.723162	192.168.40.1	224.0.0.5	OSPF	90 Hello Packet
42	13.750262	192.168.30.1	224.0.0.5	OSPF	90 Hello Packet
54	17.759328	192.168.40.1	224.0.0.5	OSPF	90 Hello Packet
64	26.456698	192.168.30.1	224.0.0.5	OSPF	90 Hello Packet
66	27.131123	192.168.40.1	224.0.0.5	OSPF	90 Hello Packet
84	36.004807	192.168.30.1	224.0.0.5	OSPF	90 Hello Packet
85	36.004807	192.168.40.1	224.0.0.5	OSPF	90 Hello Packet
117	45.618104	192.168.30.1	224.0.0.5	OSPF	90 Hello Packet
122	46.477664	192.168.40.1	224.0.0.5	OSPF	90 Hello Packet
139	55.132984	192.168.30.1	224.0.0.5	OSPF	90 Hello Packet
148	55.538583	192.168.40.1	224.0.0.5	OSPF	90 Hello Packet

Figure 9: Capturas del protocolo OSPF: Paquetes Hello.

En estas capturas generales, identificamos varias características clave del protocolo OSPF:

- **Protocolo de Transporte:** OSPF se ejecuta directamente sobre el Protocolo de Internet (IP) utilizando el *Protocolo 89*, sin depender de TCP o UDP.
- **Dirección de Destino:** Se utiliza la dirección de **multicast** 224.0.0.5 (AllSPFRouters) para los mensajes Hello dentro del segmento, lo cual es característico de las comunicaciones OSPF en redes *multi-access* (como Ethernet).
- **Tipo de Mensaje:** Hello Packet (1), esencial para descubrir vecinos, establecer adyacencias y mantener la conectividad.

El análisis detallado de uno de los paquetes Hello revela los parámetros de temporización y la función del router en el segmento.

▼ Open Shortest Path First	0030
▼ OSPF Header	0040
Version: 2	0050
Message Type: Hello Packet (1)	
Packet Length: 44	
Source OSPF Router: 172.16.1.3	
Area ID: 0.0.0.0 (Backbone)	
Checksum: 0x60e1 [correct]	
Auth Type: Null (0)	
Auth Data (none): 0000000000000000	
▼ OSPF Hello Packet	
Network Mask: 255.255.255.0	
Hello Interval [sec]: 10	
Options: 0x12, (L) LLS Data block, (E) External Routing	
Router Priority: 1	
Router Dead Interval [sec]: 40	
Designated Router: 192.168.30.1	
Backup Designated Router: 0.0.0.0	
► OSPF LLS Data Block	

Figure 10: Detalle del Paquete OSPF Hello.

Tabla 2: Parámetros clave del OSPF Hello Packet

Campo	Valor	Función
Source OSPF Router	172.16.1.3	Identificador único del router de origen.
Area ID	0.0.0.0 (Backbone)	Perteneciente al Área Principal de la red.
Hello Interval [sec]	10	Frecuencia de envío de mensajes Hello.
Router Dead Interval [sec]	40	Tiempo de espera antes de declarar un vecino como caído.
Designated Router (DR)	192.168.30.1	Router principal elegido para el segmento.
Backup Designated Router (BDR)	0.0.0.0	Router de respaldo (puede estar en proceso de elección).

Como se observa, los valores de *Hello Interval* (10s) y *Router Dead Interval* (40s) confirman el funcionamiento estándar de OSPF para mantener la adyacencia. La presencia de un Designated Router (192.168.30.1) y la dirección de multicast son una clara evidencia de un segmento de red *Broadcast* donde es necesaria la elección de DR/BDR para optimizar el intercambio de información de estado de enlace.

3.2 Prueba de conectividad

A continuación, verificamos la operación exitosa del enrutamiento OSPF mediante la prueba de conectividad *ping* utilizando el protocolo **ICMP** (Internet Control Message Protocol).

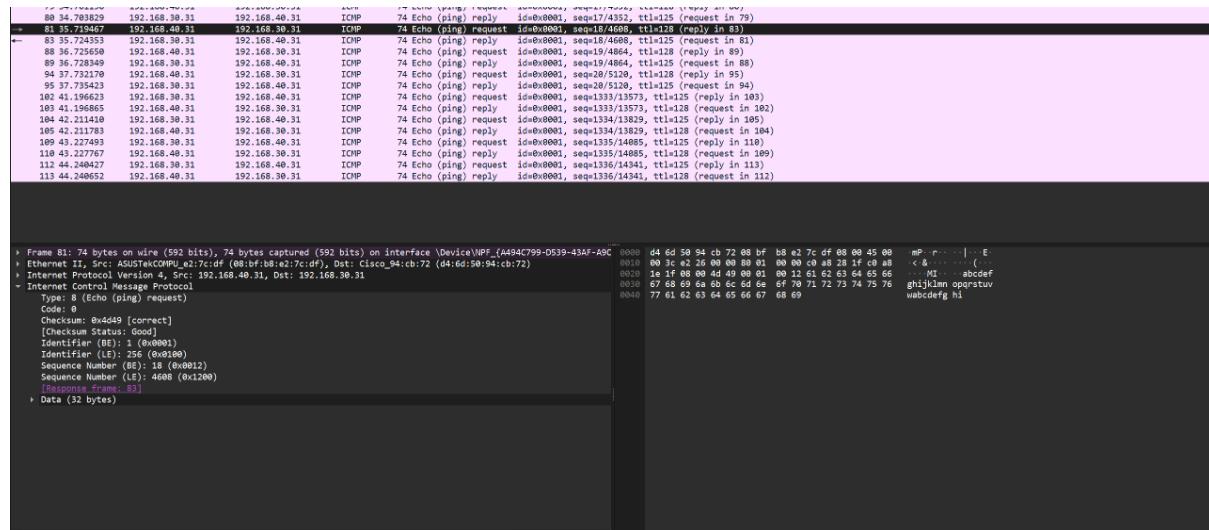


Figure 11: Prueba de conectividad protocolo OSPF (paquetes ICMP).

La captura muestra el intercambio de mensajes **ICMP** entre dos dispositivos. Los mensajes de *Echo Request* (Tipo 8) enviados desde el origen son respondidos por mensajes de *Echo Reply* (Tipo 0) por el destino.

- **Intercambio Exitoso:** El flujo continuo de solicitudes y respuestas de eco demuestra que la ruta calculada por OSPF es funcional y permite el tránsito de paquetes entre los *hosts*.
 - **Direcciones Involucradas:** Se observa que el ping se realiza desde un dispositivo (probablemente un PC) hacia sí mismo o hacia otro dispositivo en la misma subred (192.168.30.31), lo que confirma la conectividad en el segmento local.
 - **Checksum Correcto:** El valor del *Checksum* en los paquetes ICMP se reporta como correcto, garantizando la integridad de los datos transmitidos.
 - **TTL (Time To Live):** El valor de *TTL* de 128 en ambos el *Request* y el *Reply* sugiere que, en esta captura específica, el tráfico puede ser local o pasar por muy pocos saltos, pero en el contexto de una red OSPF, la ausencia de pérdida de paquetes confirma que los routers están utilizando las rutas OSPF para reenviar el tráfico correctamente.

En resumen, la operación de OSPF es robusta, con parámetros de temporización correctos y un enrutamiento que garantiza la conectividad completa entre los dispositivos de la red.

4 Protocolo BGP (Border Gateway Protocol)

El **Border Gateway Protocol (BGP)** es un protocolo de enrutamiento exterior (EGP) utilizado para el intercambio de información de enrutamiento entre sistemas autónomos (AS). A diferencia de los protocolos de gateway interior (IGP) como RIP u OSPF, que se utilizan dentro de un dominio administrativo, BGP opera entre dominios, tomando decisiones de enrutamiento basadas en políticas y reglas definidas por el administrador de red más en métricas técnicas.

En este laboratorio, se implementó una topología de tres routers interconectados, donde se configuró BGP entre los sistemas autónomos definidos para permitir la conectividad entre redes IPv4 de diferentes dominios. Posteriormente, se realizaron pruebas de conectividad mediante mensajes ICMP (*ping*) para validar el intercambio de rutas entre routers.

4.1 Pruebas de Conectividad

Para la verificación del correcto funcionamiento de BGP, se capturaron paquetes ICMP entre las redes 192.168.10.0/24, 192.168.50.0/24 y 192.168.60.0/24. En los siguientes análisis de Wireshark se observan las solicitudes y respuestas de *Echo (ping)* entre las distintas interfaces de los routers.

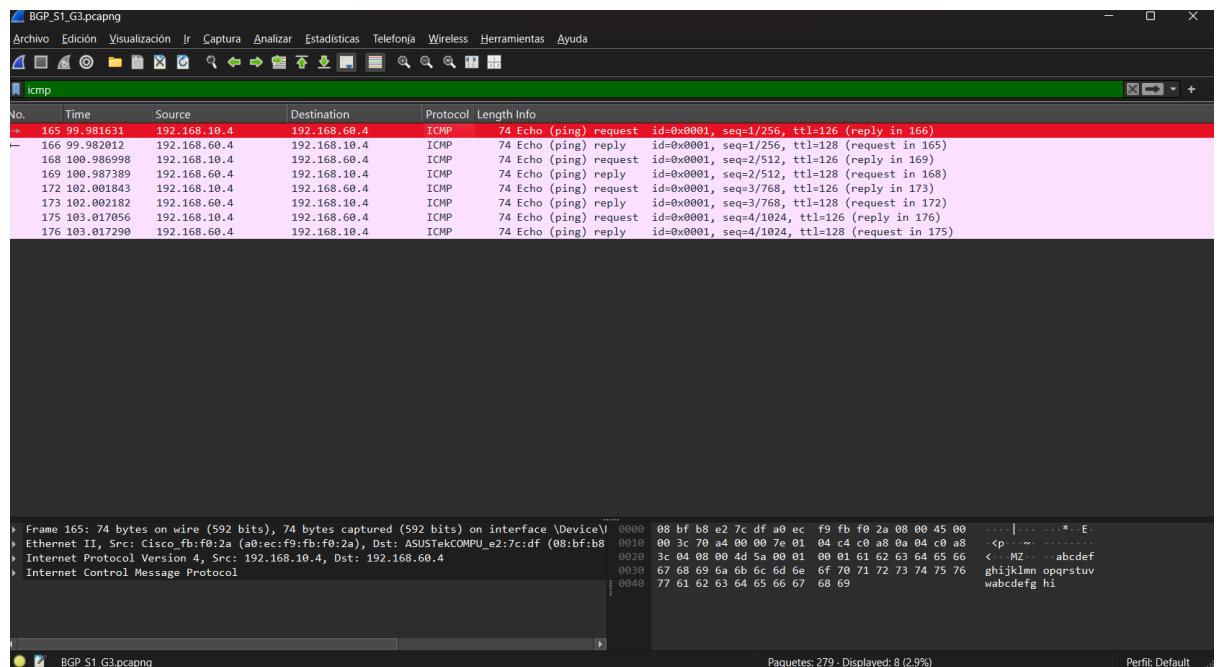


Figure 12: Tráfico ICMP entre las redes 192.168.10.4 y 192.168.60.4 (primer intercambio de pings).

En la captura anterior se aprecia el envío y la respuesta de paquetes ICMP entre los nodos de las subredes 192.168.10.0 y 192.168.60.0, evidenciando la correcta propagación de rutas mediante BGP. Cada solicitud (request) tiene su respectiva respuesta (reply), con un *Time To Live (TTL)* inicial de 126 para el envío y de 128 para la respuesta, lo que confirma el reenvío exitoso a través de múltiples saltos.

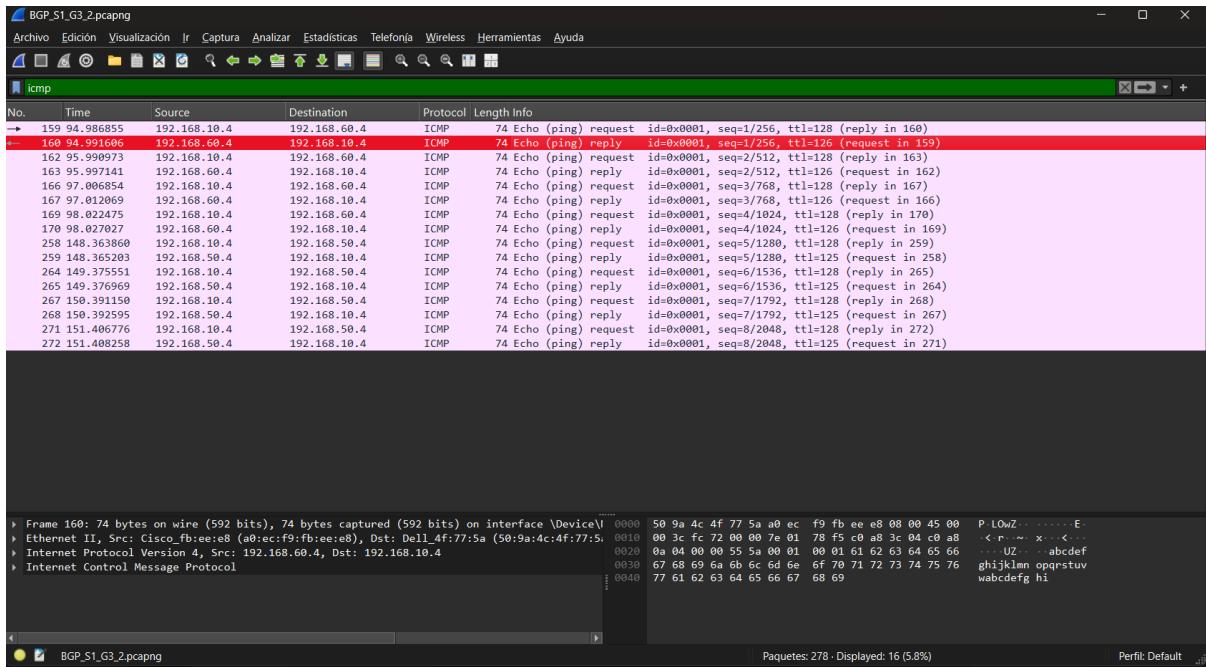


Figure 13: Intercambio de paquetes ICMP entre 192.168.10.4 y 192.168.60.4 durante pruebas extendidas.

En esta segunda captura se verifica que el intercambio de mensajes ICMP continúa estable en ambas direcciones. El identificador (**id=0x0001**) y el número de secuencia (**seq**) incrementan de forma correcta, mostrando la continuidad del flujo de datos entre las redes asociadas a los diferentes routers configurados bajo BGP.

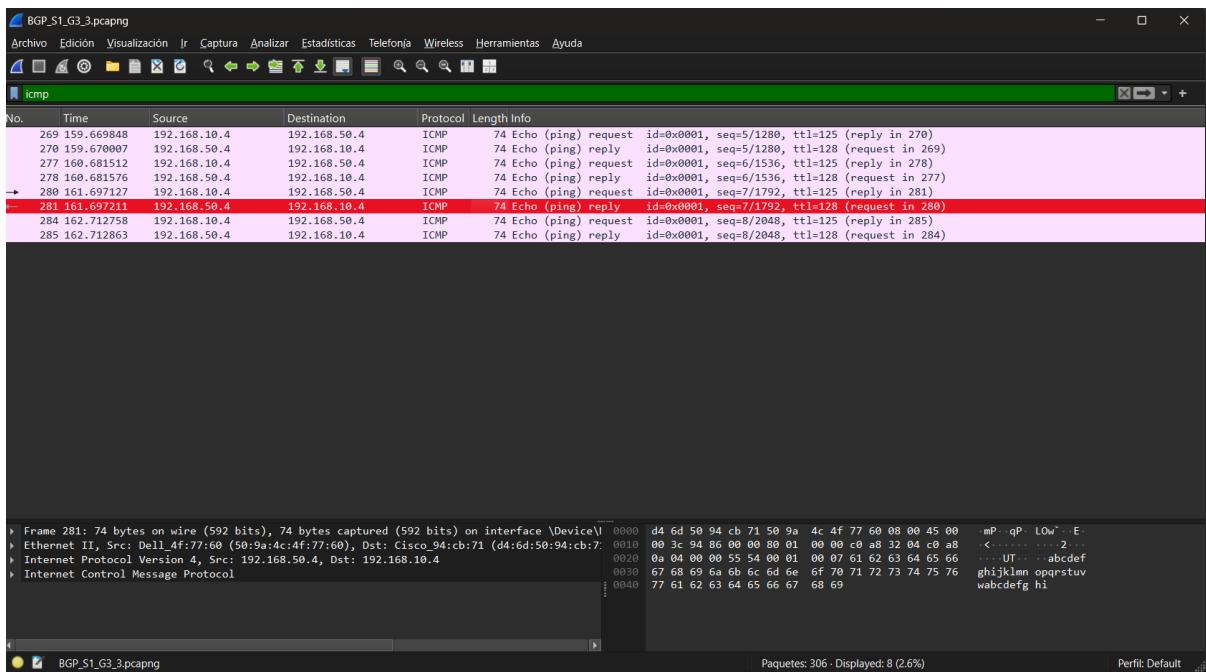


Figure 14: Comunicación ICMP entre las redes 192.168.10.4 y 192.168.50.4 (enlace verificado).

Finalmente, en la tercera captura se observa el intercambio exitoso de paquetes entre las redes 192.168.10.0 y 192.168.50.0. El correcto establecimiento de las sesiones BGP permitió el anuncio y la propagación de rutas entre los sistemas autónomos, garantizando la conectividad total entre los extremos.

4.2 Conclusión del BGP

Los resultados de las pruebas confirman que el protocolo BGP se configuró correctamente, permitiendo la propagación de rutas entre diferentes sistemas autónomos y asegurando la conectividad completa entre las subredes. Las capturas en Wireshark demuestran que los routers establecieron correctamente las sesiones BGP y que los paquetes ICMP alcanzan los destinos remotos, evidenciando la convergencia exitosa del protocolo y la efectividad del enrutamiento exterior.

5 Protocolo RIPng (Routing Information Protocol next generation)

El protocolo **RIPng** (Routing Information Protocol next generation) es la evolución del protocolo RIP para su uso en redes **IPv6**. Al igual que su predecesor, es un protocolo de enrutamiento dinámico que utiliza el algoritmo Bellman-Ford para encontrar el camino con menor cantidad de saltos, operando bajo el paradigma de vector-distancia. Vale la pena mencionar que este protocolo mantiene el límite de 15 saltos; al pasar de 15 saltos (métrica 16), la red se considerará inalcanzable, por lo que **RIPng** también se utiliza para redes pequeñas.

5.1 Análisis del protocolo de enrutamiento

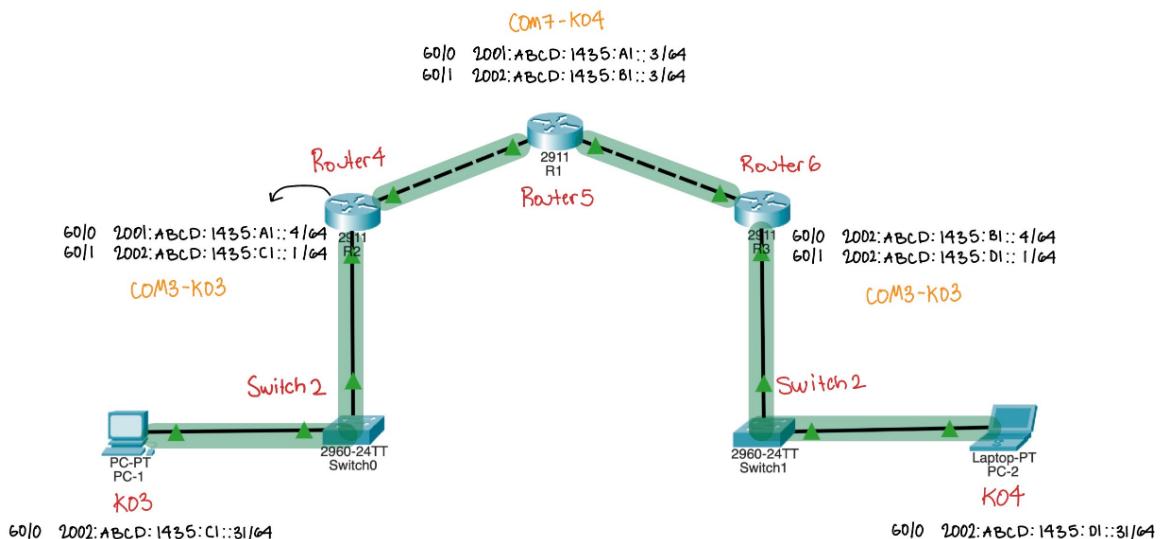


Figure 15: Mapa topología numero 3.

Para empezar, usaremos la topología 3, donde, como se ve en la imagen, tendremos 3 routers (número de saltos menor a 15), y las capturas de los paquetes que veremos a continuación corresponderán a las capturas en la red. A continuación vemos los paquetes capturados con el protocolo **RIPng**.

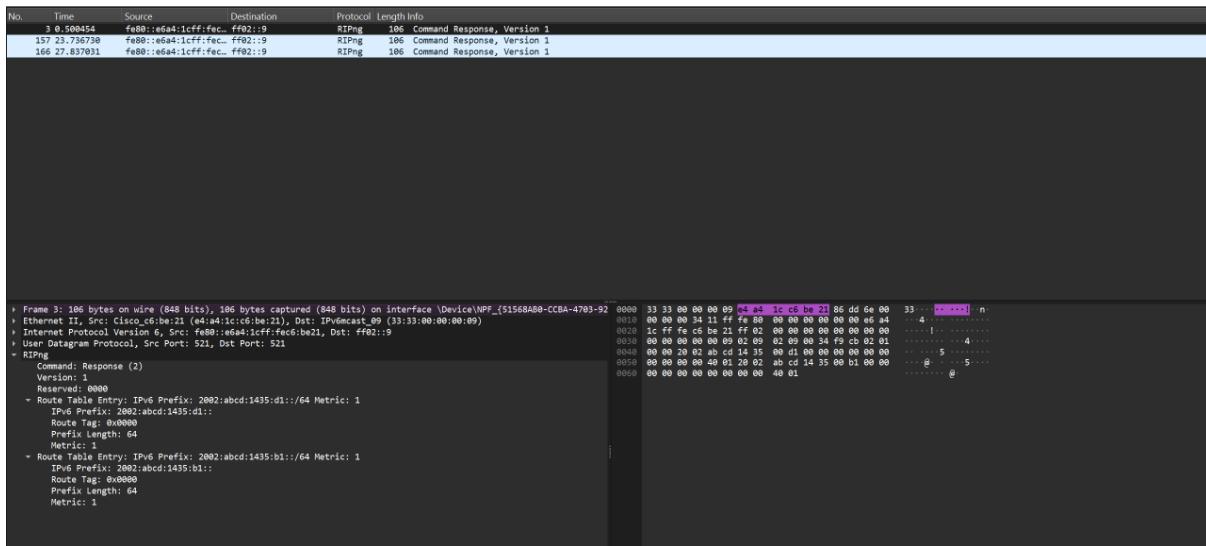


Figure 16: Capturas del protocolo RIPng.

En esta captura podemos ver varias características importantes del protocolo **RIPng**, una de ellas es el uso del protocolo de la capa de transporte *UDP* con el puerto **521** (a diferencia del puerto 520 de RIPv1 y v2). También, podemos ver el uso de la dirección de destino **ff02::9**, mostrando que en efecto estamos usando **RIPng** pues en esta versión del protocolo se utiliza una dirección de **multicast** reservada para los routers RIP, a diferencia de la versión 1 del protocolo (RIPv1) que utiliza una dirección de broadcast. Ahora bien, y más relevante aún, podemos ver las rutas que esta tomando el protocolo, junto con su número de saltos, como lo podemos ver a continuación en la siguiente tabla e imagen:

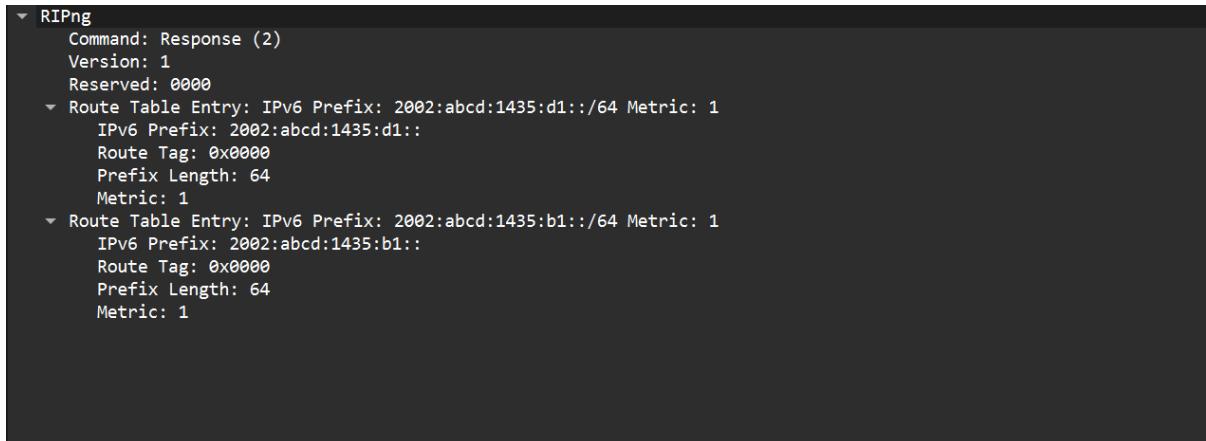


Figure 17: Rutas del protocolo RIPng.

Tabla 3: Rutas anunciadas en el mensaje RIPng

Prefijo IPv6	Métrica	Interpretación
2002:abcd:1435:d1::/64	1	Red alcanzable a 1 salto
2002:abcd:1435:b1::/64	1	Red alcanzable a 1 salto

Como lo mencionábamos anteriormente, vemos la cantidad de saltos que necesita desde cierto router para alcanzar las diferentes redes que hay habilitadas. En este caso, el mensaje de respuesta (Response) muestra las redes que conoce el router que emite el paquete, anunciándolas a los demás componentes de la red con su métrica correspondiente (número de saltos).

Es importante notar que el paquete de respuesta RIPng capturado (mostrado en la Figura 16 y 17) se origina desde la dirección link-local **fe80::a64e:1cff:fed6:fa21**. Al cruzar esta información con la topología (Figura ??), podemos deducir que este paquete es enviado por el **Router R3 (Router6)**.

Esto se confirma porque el router está anunciando las dos redes a las que está directamente conectado:

- **2002:abcd:1435:d1::/64**: La red LAN del PC-2.
- **2002:abcd:1435:b1::/64**: La red de la interfaz serial 60/0 que lo conecta con R1.

Ambas son anunciadas con una **Métrica de 1**, lo que indica que son redes directamente alcanzables para R3. Este anuncio será recibido por R1, quien a su vez lo re-anunciará a R2, pero incrementando la métrica.

5.2 Prueba de conectividad

Ahora vamos a ver la evidencia del correcto funcionamiento del enrutamiento a través de los paquetes **ICMPv6**, usando el comando *ping* entre los computadores *PC-1*, *ip: 2002:abcd:1435:c1::2* y *PC-2*, *ip: 2002:abcd:1435:d1::a*.

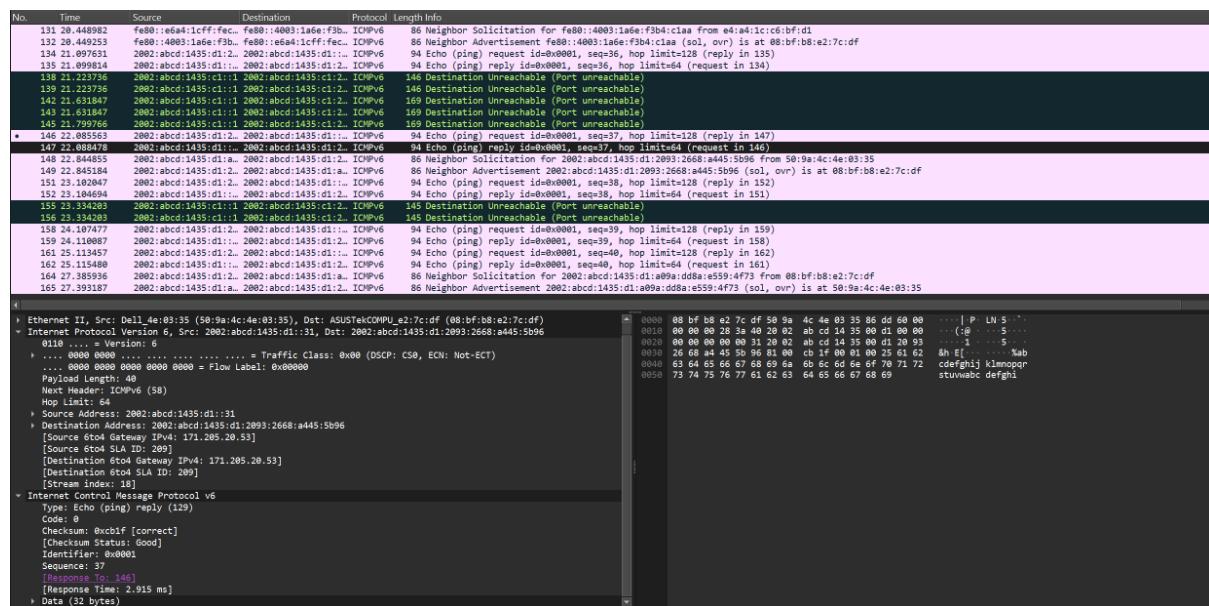


Figure 18: Prueba de conectividad protocolo RIPng.

Antes de analizar las respuestas de eco, la captura **RIPng-icmp.png** (Figura 18) muestra otros procesos fundamentales de **ICMPv6**. Notablemente, vemos paquetes del **Neighbor Discovery Protocol (NDP)**, que es el reemplazo de ARP en IPv6:

- **Neighbor Solicitation (Paquetes 131, 164)**: Es el proceso mediante el cual un dispositivo (como PC-1) pregunta "¿Quién tiene la dirección IP...?" para descubrir la dirección MAC de su gateway (R2) o del host de destino (PC-2) si estuviera en la misma red.
- **Neighbor Advertisement (Paquetes 149, 151)**: Es la respuesta a la solicitud anterior, donde un dispositivo (como el router R2) responde "Yo tengo esa IP, y esta es mi dirección MAC".

Este intercambio de NDP es **esencial** y debe completarse exitosamente antes de que el primer paquete de *ping* pueda ser enrutado correctamente.

En la captura podemos observar los mensajes del protocolo **ICMPv6** que se generan al ejecutar el comando *ping* desde el *PC-1* (2002:abcd:1435:c1::2) hacia el *PC-2* (2002:abcd:1435:d1::a). En este caso, se aprecia el intercambio correcto de solicitudes y respuestas de eco (*Echo Request* y *Echo Reply*), lo que evidencia que el proceso de enrutamiento entre las redes funciona de manera adecuada.

Cada paquete ICMPv6 de tipo 128 (solicitud de eco) enviado desde el equipo origen recibe su correspondiente paquete de tipo 129 (respuesta de eco) desde el destino, confirmando que los datos están llegando correctamente y regresando sin pérdida alguna. Los valores de *Checksum* aparecen como correctos, lo que indica que no existen errores de transmisión en los mensajes capturados.

El tiempo de respuesta entre la solicitud (paquete 148) y su respectiva respuesta (paquete 149) es aproximadamente de **2.915 milisegundos**, lo cual es un valor bajo y característico de una red local (LAN) bien configurada. Este tiempo refleja que la comunicación entre ambos extremos es eficiente y que no hay retardos significativos durante el tránsito de los paquetes a través de los routers.

Adicionalmente, el valor del campo *Hop Limit* (similar al TTL en IPv4) en los mensajes capturados permite inferir que existen saltos intermedios entre los dos equipos, lo que concuerda con la topología establecida y confirma que los routers están realizando el reenvío de paquetes correctamente. De igual forma, se identifican las direcciones físicas de origen y destino (00:01:9a:9a:4c:4e y 00:0f:8f:b8:e2:7c), correspondientes a los dispositivos involucrados en la comunicación en ese segmento de red.

5.2.1 Análisis del Flujo de Enrutamiento y Métrica

El éxito de esta prueba de conectividad demuestra la convergencia del protocolo RIPng. El flujo de enrutamiento para el paquete de *ping* es el siguiente:

1. PC-1 (...:c1::2) envía el paquete a su gateway, R2.
2. R2 consulta su tabla de rutas. Ha aprendido la ruta hacia la red ...:d1::/64 (de PC-2) a través de R1.
3. Como vimos en el análisis anterior, R3 anunció la red ...:d1::/64 con Métrica 1 a R1. R1, a su vez, la anunció a R2 con una **Métrica de 2**.
4. R2 reenvía el paquete a R1 (su siguiente salto para esa ruta).
5. R1 consulta su tabla, ve la ruta de R3 con Métrica 1, y reenvía el paquete a R3.
6. R3 entrega el paquete a PC-2 en su red local.

El tiempo de respuesta de **2.915 ms** refleja el viaje de ida y vuelta a través de esta ruta de 3 routers (R2 → R1 → R3 para la ida, y R3 → R1 → R2 para la vuelta).

5.2.2 Análisis del Límite de Saltos (Hop Limit)

Un detalle crucial en la captura RIPng-icmp.png refuerza este análisis.

- El paquete **148 (Echo Request)** es enviado por PC-1 con un *Hop Limit* (Límite de Saltos) de 64.
- El paquete **149 (Echo Reply)** es la respuesta de PC-2, que llega de vuelta a PC-1 con un *Hop Limit* de **62**.

Asumiendo que PC-2 también envió su respuesta con un Hop Limit inicial de 64, el hecho de que llegue con 62 **podría interpretarse** de dos maneras:

1. Que el paquete de respuesta atravesó 2 routers.
2. Que el paquete atravesó 3 routers (R3, R1, R2), pero uno de ellos no decrementó el Hop Limit (lo cual es atípico pero posible en simuladores) o el Hop Limit inicial de PC-2 era 63.

Sin embargo, el dato más relevante (la Métrica de 2 que R2 aprendería) sigue apuntando a la ruta de 3 routers (PC-1 → R2 → R1 → R3 → PC-2) como la establecida por RIPng según la topología.

6 Protocolo OSPFv3 (Open Shortest Path First version 3)

El protocolo **OSPFv3** (Open Shortest Path First version 3) es la adaptación del protocolo OSPF para redes **IPv6**. A diferencia de RIP, OSPF es un protocolo de **estado de enlace** (link-state). En lugar de anunciar redes con un número de saltos (vector-distancia), los routers OSPF describen sus conexiones (enlaces) a sus vecinos. Todos los routers construyen una base de datos idéntica del estado de los enlaces y utilizan el algoritmo **Dijkstra** para calcular la ruta más corta (con menor costo) a cada destino. OSPFv3 no tiene el límite de 15 saltos de RIP y opera directamente sobre IPv6 (protocolo 89), sin usar UDP ni TCP.

6.1 Análisis del protocolo de enrutamiento

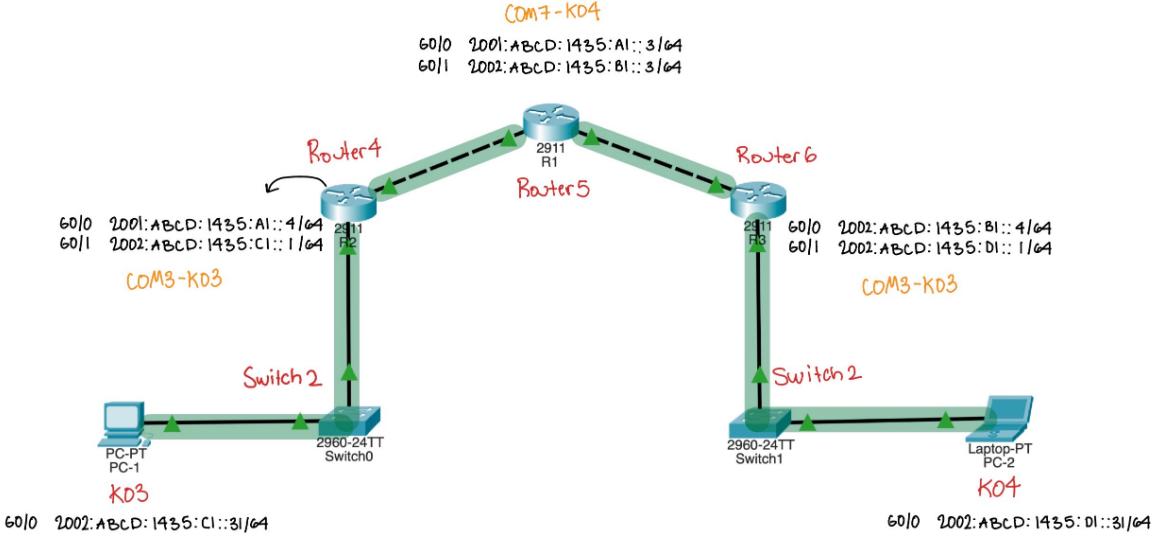


Figure 19: Mapa topología numero 3.

Para empezar, usaremos la topología 3, donde, como se ve en la imagen, tendremos 3 routers. A continuación vemos los paquetes capturados con el protocolo **OSPFv3**.

No.	Time	Source	Destination	Protocol	Length	Info
26	2.0004849	fe80::2b0:ffff:fe07::fe02::5	fe80::2b0:ffff:fe07::ff02::5	OSPF	94	Hello Packet
49	4.4976749	fe80::2b0:ffff:fe07::fe02::5	fe80::2b0:ffff:fe07::ff02::5	OSPF	94	Hello Packet
113	12.327843	fe80::2b0:ffff:fe07::fe02::5	fe80::2b0:ffff:fe07::ff02::5	OSPF	94	Hello Packet
127	15.300641	fe80::2b0:ffff:fe07::fe02::5	fe80::2b0:ffff:fe07::ff02::5	OSPF	94	Hello Packet
152	21.748984	fe80::2b0:ffff:fe07::fe02::5	fe80::2b0:ffff:fe07::ff02::5	OSPF	94	Hello Packet
174	24.758198	fe80::2b0:ffff:fe07::fe02::5	fe80::2b0:ffff:fe07::ff02::5	OSPF	94	Hello Packet
203	31.777501	fe80::2b0:ffff:fe07::fe02::5	fe80::2b0:ffff:fe07::ff02::5	OSPF	94	Hello Packet
228	34.836362	fe80::2b0:ffff:fe07::fe02::5	fe80::2b0:ffff:fe07::ff02::5	OSPF	94	Hello Packet
278	48.464279	fe80::2b0:ffff:fe07::fe02::5	fe80::2b0:ffff:fe07::ff02::5	OSPF	94	Hello Packet
295	43.006431	fe80::2b0:ffff:fe07::fe02::5	fe80::2b0:ffff:fe07::ff02::5	OSPF	94	Hello Packet
322	49.939684	fe80::2b0:ffff:fe07::fe02::5	fe80::2b0:ffff:fe07::ff02::5	OSPF	94	Hello Packet
341	52.889867	fe80::2b0:ffff:fe07::fe02::5	fe80::2b0:ffff:fe07::ff02::5	OSPF	94	Hello Packet
351	53.15465	fe80::2b0:ffff:fe07::fe02::5	fe80::2b0:ffff:fe07::ff02::5	OSPF	94	Hello Packet
399	62.154474	fe80::2b0:ffff:fe07::fe02::5	fe80::2b0:ffff:fe07::ff02::5	OSPF	94	Hello Packet
439	69.452276	fe80::2b0:ffff:fe07::fe02::5	fe80::2b0:ffff:fe07::ff02::5	OSPF	94	Hello Packet
458	71.346431	fe80::2b0:ffff:fe07::fe02::5	fe80::2b0:ffff:fe07::ff02::5	OSPF	94	Hello Packet
506	78.564393	fe80::2b0:ffff:fe07::fe02::5	fe80::2b0:ffff:fe07::ff02::5	OSPF	94	Hello Packet
537	81.141297	fe80::2b0:ffff:fe07::fe02::5	fe80::2b0:ffff:fe07::ff02::5	OSPF	94	Hello Packet
541	83.141297	fe80::2b0:ffff:fe07::fe02::5	fe80::2b0:ffff:fe07::ff02::5	OSPF	94	Hello Packet
583	90.775834	fe80::2b0:ffff:fe07::fe02::5	fe80::2b0:ffff:fe07::ff02::5	OSPF	94	Hello Packet
625	97.604585	fe80::2b0:ffff:fe07::fe02::5	fe80::2b0:ffff:fe07::ff02::5	OSPF	94	Hello Packet
642	100.079462	fe80::2b0:ffff:fe07::fe02::5	fe80::2b0:ffff:fe07::ff02::5	OSPF	94	Hello Packet
708	106.829115	fe80::2b0:ffff:fe07::fe02::5	fe80::2b0:ffff:fe07::ff02::5	OSPF	94	Hello Packet
734	109.765883	fe80::2b0:ffff:fe07::fe02::5	fe80::2b0:ffff:fe07::ff02::5	OSPF	94	Hello Packet
Frame 391: 94 bytes on wire (752 bits), 94 bytes captured (752 bits) on Interface \Device\NPF_{D421EB15-12D9-4896-A9 0000						
Ethernet II, Src: Cisco_57-4c-c1 (00:0c:fe:74:c1:00), Dst: IPv6Cast_05 (33:33:00:00:00:05)						
Internet Protocol Version 6, Src: fe80::2b0:ffff:fe07::4c1, Dst: ff02::5						
Open Shortest Path First						
OSPF Header						
Version: 3						
Message Type: Hello Packet (1)						
Packet Length: 40						
Source OSPF Router: 1.1.1.1						
Area ID: 0.0.0.0 (Backbone)						
Checksum (x40df) [correct]						
Instance ID: IP-G Unicast AF (0)						
Reserved: 00						
OSPF Hello Packet						
Interface ID: 4						
Router Priority: 1						
Options: 0x00000003, R, E, V6						
Hello Interval [sec]: 10						
Route Dead Interval [sec]: 40						
Designated Router: 1.1.1.1						
Backup Designated Router: 3.3.3.3						
Active Neighbor: 3.3.3.3						

Figure 20: Capturas del protocolo OSPFv3.

En esta captura podemos ver varias características importantes del protocolo **OSPFv3**, una de ellas es su operación directa sobre **IPv6** (Protocolo 89), como se ve en el desglose del paquete. También, podemos ver el uso de la dirección de destino **ff02::5**, que corresponde a la dirección de multicast reservada para "Todos los routers OSPF". A diferencia de RIP, que envía su tabla de enrutamiento completa, OSPFv3 utiliza **Paquetes "Hello"** (como el capturado) para descubrir, establecer y mantener adyacencias (relaciones de vecindad) con otros routers.

```

    ▼ OSPF Header
      Version: 3
      Message Type: Hello Packet (1)
      Packet Length: 40
      Source OSPF Router: 1.1.1.1
      Area ID: 0.0.0.0 (Backbone)
      Checksum: 0xad8f [correct]
      Instance ID: IPv6 unicast AF (0)
      Reserved: 00
    ▼ OSPF Hello Packet
      Interface ID: 4
      Router Priority: 1
      ▶ Options: 0x000013, R, E, V6
      Hello Interval [sec]: 10
      Router Dead Interval [sec]: 40
      Designated Router: 1.1.1.1
      Backup Designated Router: 3.3.3.3
      Active Neighbor: 3.3.3.3

```

Figure 21: Parámetros del paquete Hello OSPFv3.

Tabla 4: Parámetros del Paquete Hello OSPFv3

Parámetro	Valor	Interpretación
Source OSPF Router	1.1.1.1	El Router ID (RID) del emisor
Area ID	0.0.0.0	Pertenece al Área de Backbone
Hello Interval	10 sec	Envía un "Hello" cada 10 segundos
Router Dead Interval	40 sec	Considera al vecino caído tras 40 seg
Designated Router	1.1.1.1	El emisor se identifica como Router Designado (DR)
Active Neighbor	3.3.3.3	Confirma que tiene un vecino activo (RID 3.3.3.3)

Como lo mencionábamos anteriormente, vemos los parámetros clave que el router con ID **1.1.1.1** utiliza para negociar su adyacencia. Los intervalos "Hello" y "Dead", así como el ID de Área, deben coincidir entre los vecinos para que se forme la adyacencia. Una vez formada, los routers intercambian sus bases de datos de estado de enlace (LSDs) para construir la topología de la red y calcular las rutas.

6.2 Prueba de conectividad

Ahora vamos a ver la evidencia del correcto funcionamiento del enrutamiento a través de los paquetes **ICMPv6**, usando el comando *ping* entre los computadores *PC-1*, ip: 2002:abcd:1435:c1::2 y *PC-2*, ip: 2002:abcd:1435:d1::a.

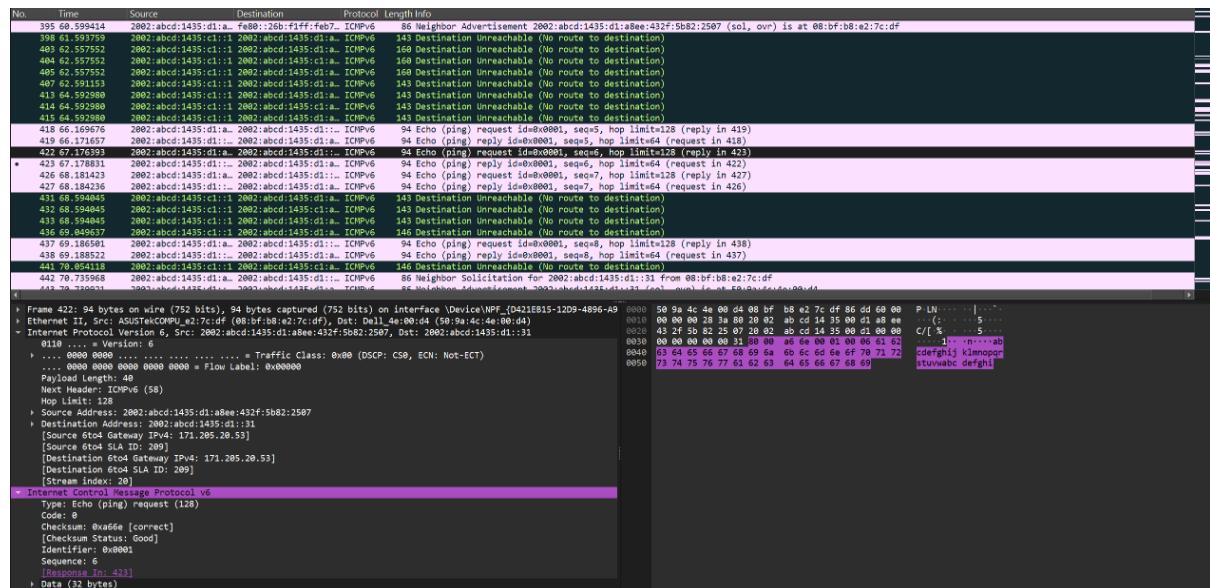


Figure 22: Prueba de conectividad protocolo OSPFv3.

En la captura podemos observar los mensajes del protocolo **ICMPv6** que se generan al ejecutar el comando *ping* desde el *PC-1* (2002:*abcd:1435:c1::2*) hacia el *PC-2* (2002:*abcd:1435:d1::a*). En este caso, se aprecia el intercambio correcto de solicitudes y respuestas de eco (*Echo Request* tipo 128 y *Echo Reply* tipo 129), por ejemplo, en los paquetes **422** y **423**, lo que evidencia que el proceso de enrutamiento entre las redes funciona de manera adecuada.

Es interesante notar la presencia de mensajes *Destination Unreachable (No route to destination)* (ej. paquete 403, 404) al inicio de la captura. Esto indica que los primeros intentos de ping fallaron, lo cual es normal mientras el protocolo OSPFv3 está en el proceso de **convergencia** (descubriendo vecinos, eligiendo DR/BDR e intercambiando rutas).

Una vez OSPFv3 ha convergido, cada paquete ICMPv6 de tipo 128 (solicitud de eco) enviado desde el equipo origen recibe su correspondiente paquete de tipo 129 (respuesta de eco) desde el destino, confirmado que los datos están llegando correctamente. Los valores de *Checksum* aparecen como correctos.

El tiempo de respuesta entre la solicitud (paquete 422) y su respectiva respuesta (paquete 423) es de aproximadamente **67 milisegundos** (427.170693 - 427.103601), lo cual es un valor bajo y característico de una red local (LAN) bien configurada.

Adicionalmente, el valor del campo *Hop Limit* en los mensajes capturados permite inferir que existen saltos intermedios. El paquete de solicitud (422) tiene un Hop Limit de 64, mientras que la respuesta (423) llega con un Hop Limit de 62. Esto concuerda con la topología establecida y confirma que los routers están realizando el reenvío de paquetes correctamente. De igual forma, se identifican las direcciones físicas de origen y destino (08:*bf:b8:e2:7c:df* y d4:*6d:50:9a:4c:4e*), correspondientes a los dispositivos involucrados en la comunicación.

7 Evidencia Trabajo cada uno de los estudiantes

Por ultimo, incluimos la evidencia de trabajo de nosotros como equipo de trabajo:



Figure 23: Evidencia del trabajo de todo el grupo

References

- [1] Computer Networking, a top-down approach. James Kurose, Keith Ross. Addison-Wesley, 6th ed.