UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf

Escola Tècnica Superior d'Enginyeria Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Estudio sobre OSPF

Trabajo SCO

MUCC

Autor: Pablo Picó Silvestre

Curso 2025-2026

Índice general

Índice general I
Índice de figuras II

1 Introducción 1
2 Protocolo OSPF 2
2.1 Términos . . . 2
2.2 Funcionamiento . . . 2
2.3 Áreas OSPF . . . 3
2.4 Tipos de paquetes . . . 3
2.5 Anuncios del estado de la red (LSA) . . . 4
3 Configuraciones del protocolo OSPF en Mikrotik 5
3.0.1 Instancia OSPF . . . 5
3.0.2 Definición de área . . . 5
3.0.3 Activación de Interfaces (Interface Templates) . . . 5
3.0.4 Securización . . . 6
4 Simulación de red con el protocolo OSPF 7
4.1 Topología con redundancias . . . 7
4.1.1 Configuración inicial del router . . . 7
4.1.2 Configuración del protocolo OSPF . . . 8
4.1.3 Comprobaciones . . . 11
4.1.4 Aumento de costos o caída en un router . . . 12
4.1.5 Securización . . . 12
4.1.6 Wireshark . . . 13
4.2 Topología con distintas áreas . . . 16
4.2.1 Configuraciones . . . 16
4.2.2 Comprobaciones . . . 17
5 Conclusión 19

Índice de figuras

4.1 Topología de la red virtualizada 7
4.2 Direcciones IP asignadas a las interfaces de R4 8
4.3 Creación de una instancia OSPF 9
4.4 Creación de una área OSPF 9
4.5 Configuración de vecinos de R1 10
4.6 Configuración subnet de R4 10
4.7 Vecinos de R1 11
4.8 Mensajes LSA recibidos por R1 11
4.9 Traceroute durante un cambio de costos y una caída de router. 12
4.10 Parámetros de autenticación OSPF 13
4.11 Tabla LSA de R1 13
4.12 Paquete *hello* capturado mediante Wireshark 14
4.13 Captura de Wireshark donde se muestran distintos paquetes OSPF 15
4.14 Topología con distintas áreas 16
4.15 Comunicación entre áreas 17
4.16 Tabla LSA de R2 18

CAPÍTULO 1
Introducción

En el ámbito de las redes de computadoras, la escalabilidad y la eficiencia en la transmisión de datos son requisitos muy importantes. Los protocolos de enrutamiento dinámico juegan un papel crucial en este escenario, permitiendo a los dispositivos de red adaptarse automáticamente a cambios en la topología. Entre estos, el protocolo OSPF (Open Shortest Path First) destaca como un estándar abierto que utiliza el algoritmo de Dijkstra para calcular las rutas más eficientes.

Este trabajo tiene como finalidad realizar un análisis del funcionamiento del protocolo OSPF, abordando tanto sus fundamentos teóricos como su aplicación práctica. Para ello, se ha llevado a cabo una prueba de concepto utilizando el entorno de virtualización PNetLab sobre Proxmox, implementando la configuración en equipos MikroTik con RouterOS v7.

Mediante este proyecto se tratan de conseguir los siguientes objetivos principales:

- Comprender y explicar el funcionamiento del protocolo OSPF, analizando conceptos como la base de datos de estado de enlace (LSDB), el intercambio de mensajes LSA y la jerarquía de áreas.
- Configurar el protocolo OSPF en routers MikroTik, detallando el uso de instancias, áreas y plantillas de interfaz (Interface Templates).
- Simular distintos escenarios de red, incluyendo topologías con redundancia para verificar la actualización de rutas ante fallos y topologías multi-área para estudiar la interconexión y segmentación lógica de una corporación.
- Implementar medidas de autenticación para proteger la infraestructura de red frente a dispositivos no autorizados.
- Analizar el tráfico generado por el protocolo mediante herramientas de captura como Wireshark, con el objetivo de comprender el intercambio de mensajes OSPF a bajo nivel.

CAPÍTULO 2
Protocolo OSPF

El protocolo OSPF (Open Shortest Path First) es un protocolo abierto de enrutamiento dinámico que utiliza el algoritmo Dijkstra para calcular la ruta más eficiente hacia el destino y se actualiza automáticamente cuando hay cambios en la red.

A diferencia de otros protocolo como RIP el cual está limitado a 15 nodos y envía actualizaciones de ruta cada 30 segundos. OSPF solo envía las actualizaciones cuando hay un cambio en la topología lo que reduce la carga en la red.

OSPF crea un mapa completo de la red mediante una base de datos con el estado de las conexiones (LSDB: Link State Database) con la que puede construir facilmente cualquier ruta. El problema es que el aumento de tiempo y recursos crece con en número de rutas que se deben manejar. Por lo tanto, OSPF se utiliza como un protocolo de enrutamiento interior (IGP) y no como enrutamiento en Internet.

2.1 Términos

Cada router dentro del protocolo tiene una **Router ID** que lo identifica y un estado que hace saber al resto si se encuentra funcionando (UP) o desconectado (DOWN).

Para el funcionamiento del protocolo se designa un router, **DR (Designated Router)** para el intercambio de información, reduciendo la cantidad de mensajes y un enrutador de respaldo (**BDR**) por si el DR falla. Los routers intercambian mensajes **LSA (Link-State Advertisements)** para actualizar la información sobre el estado de la red. La **LSDB** (Link-State Database) se completa con estos mensajes y contiene información actualizada sobre el estado de la red, las rutas disponibles y su costo. El costo (**OSPF cost**) se calcula basado en el ancho de banda, cuanto más ancho de banda menos costo.

2.2 Funcionamiento

Cada enrutador con el protocolo OSPF intercambia sus rutas disponibles y su costo con el resto para determinar los vecinos y calcular las rutas. Estas rutas pueden cambiar en caso de que alguno de los enrutadores este sobrecargado o se desconecte.

Al inicio del protocolo todos los enrutadores envían paquetes *hello* para establecer la conexión con el resto. Cada uno de estos paquetes contiene:

- Router ID: identificador del router.

- OSPF-priority: valor de 0 a 255 que determina la probabilidad de que el enrutador se convierta en DR. El router con mayor valor es el designado.
- Máscara de subred.
- Identificador de zona OSPF: esto permite dividir los routers en zonas para no sobrecargar el protocolo.
- Tipo de autenticación: para asegurar que el intercambio de información se realiza únicamente entre dispositivos confiables.
- Intervalo de envío de paquetes *hello*: determina la frecuencia de envío de paquetes para verificar el estado de los vecinos. (Por defecto 10 segundos)
- Dead timer: tiempo de espera del paquete*hello*. Si no hay respuesta en este tiempo, el vecino se marca como no disponible.

Al enviar y recibir paquetes *hello*, los routers actualizan sus bases de datos **LSBD** y comienzan a intercambiar información sobre la topología y se designan el DR y BDR con el valor del OSPF-priority. El resto de routers se marcan con el estado DROTHER para evitar bucles o envío de datos innecesarios.

2.3 Áreas OSPF

Este protocolo no escala bien, con más enrutadores la frecuencia de la actualización de la LSBD se reduce y su tamaño aumenta. En el pasado, debido a las limitaciones tecnologicas era necesario realizar una divisón por áreas para mantener la eficacia. En la actualidad ya no existen estas limitaciones y una misma zona puede incluir miles de routers, por lo que el zonificado se utiliza más para facilitar la gestión que para resolver problemas de escalabilidad.

Al realizar la división por áreas, cada ruter puede pertenecer únicamente a una de estas exceptuando los roters de frontera (**ABR area border routers**) responsables del intercambio de datos entre zonas.

Hay distintos tipos de áreas:

- **Standard area**: esta zona intercambia información de enrutamiento completa tanto dentro de la zona como a través del backbone.
- **Stub area**: esta zona limita la cantidad de información respecto a las rutas externas para reducir la carga de los routers.
- **NSSA**: not-so-stubby area, esta zona no recibe información sobre rutas externas pero, a diferencia de la anterior, si que puede enviarla (mediante paquetes LSA de tipo 7).

2.4 Tipos de paquetes

El protocolo OSPF envía distintos tipos de paquetes:

- *hello* packet: sirve para descubrir vecinos y mantenerlos vivos ("keepalive")
- Database Description: Sirven para la sincronización de la base de datos (LSBD), el router envía solo los encabezados de los LSAs que tiene.

- Link State Request (LSR): Viendo el paquete anterior, el router pide la información que le falta mediante este tipo de paquetes.
- Link State Update (LSU): El paquete que contiene los LSA en respuesta a un LSR.
- Link State Aknowledgement (LSAck): Confirma recepción de un paquete*hello* para decir que sigue activo o de un paquete LSU.

2.5 Anuncios del estado de la red (LSA)

Los mensajes LSA (Link State Announcement) sirven para intercambiar información sobre rutas y el estado de la red. Hay de distintos tipos:

- **Router LSA**(Tipo 1): indica el identificador del router y el de sus vecinos dentro del área.
- **Network LSA** (Tipo 2): lo genera únicamente el designated router para identificarse como tal.
- **Summary LSA** (Tipo 3): generado por los routers de frontera para anunciar sus rutas de red a otras zonas.
- **ASBR Summary LSA** (Tipo 4): generado por los routers de frontera si hay una ruta externa en una de las zonas. Informa la ruta al router que gestiona esta ruta externa.
- **AS-External LSA** (Tipo 5): generado por los routers de frontera que esten conectados a redes externas para expandir la ruta que ha aprendido. Este mensaje se envía a toda la red excepto a las áreas configuradas como Stub o NSSA.
- **Multicast OSPF LSA** (Tipo 6): permite sincronizar rutas de multidifusión IPv6, solo se utiliza en OSPFv3.
- **NSSA external LSA** (Tipo 7): permite a las áreas de tipo NSSA enviar rutas externas. Una vez sale del área NSSA el paquete se convierte en un LSA de tipo 5.

CAPÍTULO 3
Configuraciones del protocolo OSPF en Mikrotik

En este capítulo se mostrará como se ha implementado el protocolo OSPF versión 2 utilizando equipos MikroTik con RouterOS v7.

3.0.1. Instancia OSPF

Para iniciar el la configuración del protocolo es necesario crear una instancia en la que se especifique la ID del router y la versión de protocolo a utilizar, siendo la versión dos para la IPv4 y la tres para la IPv6.

Es recomendable utilizar la IP de loopback como identificador del router debido a que, a diferencia de las interfaces físicas, esta nunca se cae almenos que el router se apague.

3.0.2. Definición de área

Todos los routers de una misma área se comunican mediante el protocolo OSPF y comparten la misma LSDB. Como se explica anteriormente, se pueden usar varias áreas para dividir una organización pero no es un requisito. Debido a la mejora en el rendimiento, en la actualidad, la división de áreas sirve más para organizar que para mejorar la escalabilidad.

Como mínimo se debe configurar un área principar para la cual debemos determinar su identificador, a que instancia OSPF pertenece y su tipo (default, nssa o stub).

3.0.3. Activación de Interfaces (Interface Templates)

Mediante la creación de plantillas, se configura como los routers se comunican entre si dentro del área OSPF. A continuación se explican los distintos parámetros configurables:

- **Interfaces**: se decide que interfaces son afectadas por esta regla.
- **Networks**: si se deja por defecto, se activa en todas las direcciones ip asignadas a la interfaz seleccionada. Si se especifica una red, solo se activará si la interfaz tiene una ip dentro de ese rango.
- **Network Type**: define como OSPF encapsula los paquetes y a quien los envía. Nos centraremos en el modo broadcast y el PTP. El primero envía los paquetes al DR

para que este los reenvíe, el segundo, se utiliza cuando hay un enlace directo entre dos routers.

- **Area**: asigna el área a la que pertenece la regla.
- **Passive**: en modo pasivo la interfaz no envía ni escucha paquetes*hello* pero permite que el resto de nodos sepan la ruta hacía la subred asociada a esa interfaz.
- **Cost**: representa la penalización por usar este enlace.
- **Priority**: determina que router será seleccionado como DR.
- **Autenticación**: permite asignar un algoritmo de encriptación y una clave para impedir que routers no autorizados se unan a la topología.
- **Parámetros de control**: nos sirven para configurar con que frecuencia se envían los paquetes*hello*, a partir de cuanto tiempo se considera un router como caído o tiempo que el router espera a volver a enviar un paquete LSA si no ha recibido una confirmación de recibo.

3.0.4. Securización

Con en los parámetros de autenticación podemos elegir la contraseña y el tipo de cifrado a utilizar, entre los que tenemos: md5, varias versiones de sha y simple. Este último, manda la contraseña en texto llano por la red, lo que hace que esta medida de seguridad pierda su eficacia.

CAPÍTULO 4

Simulación de red con el protocolo OSPF

En este capítulo se realizará la simulación de dos redes para ver distintos aspectos del protocolo OSPF. Primeramente se simulará una red con redundancias para comprobar como el protocolo es capaz calcular la mejor ruta para el direccionamiento de mensajes, adaptarse si algún enrutador cae y evitar bucles. Posteriormente, se observará como se pueden crear distintas áreas en las que se podría organizar una corporación y poder mantener la comunicacón entre distintas zonas.

Esta simulación se ha realizado en el entorno de virtualización PnetLab corriendo sobre el hipervisor proxmox. La configuración del protocolo se ha realizado en routers Mikrotik 7.21.1 a los que se ha accedido mediante Winbox.

4.1 Topología con redundancias

Para la primera simulación de red, se ha elegido una topología con 4 routers donde R1 y R4 están conectados por R2 y R3 formando un ciclo como podemos ver en la imagen 4.1. Con este diseño de red se pretende comprobar la resiliencia del protocolo frente a cambios en la topología y como el protocolo es capaz de elegir una ruta cuando hay más de una posibilidad.

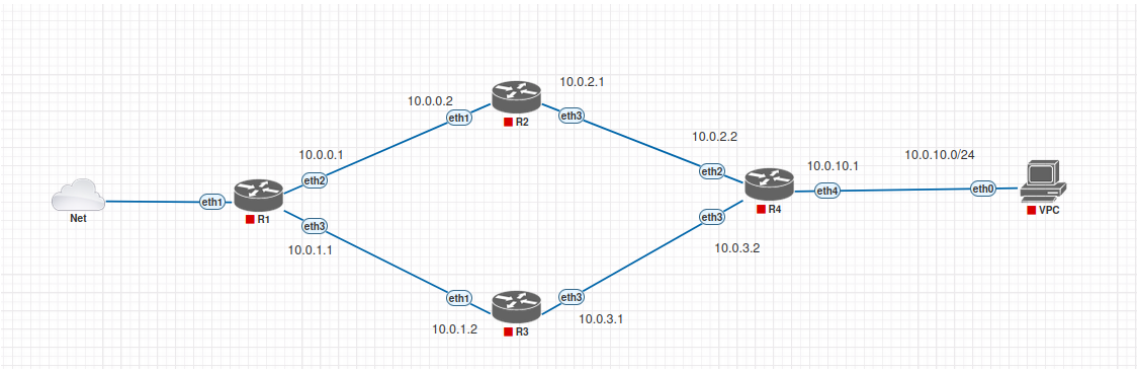

Figura 4.1: Topología de la red virtualizada

4.1.1. Configuración inicial del router

Para poder acceder a cada uno de los routers mediante winbox, los hemos conectado una red de tipo Management(Cloud0) que les da una IP dentro de nuestra red local.

Para activar el dhcp dentro del router se ha usado el comando: *ip dhcp-client add interface=etherX disable=no*. Posteriormente, se ha usado *ip address print* para saber a que dirección IP conectarse.

Al acceder al router mediante Winbox, establecemos las IPs necesarias a cada una de las interfaces según se ha mostrado en la topologia de la imagen 4.1. Además se designa una ip para la interfaz de loopback que utilizaremos como host-id. Para que sea más comprensible se ha decidido usar una ip con la forma x.x.x.x donde x es el número de router, por ejemplo, R4 tiene la ip de loopback 4.4.4.4.

En la figura 4.2 podemos ver las ips asignadas en R4. Entre ellas está la de loopback, las direcciones que conectan R4 con R2 y R3, la dirección dentro de la LAN y la dirección IP que nos permite acceder desde nuestra máquina.

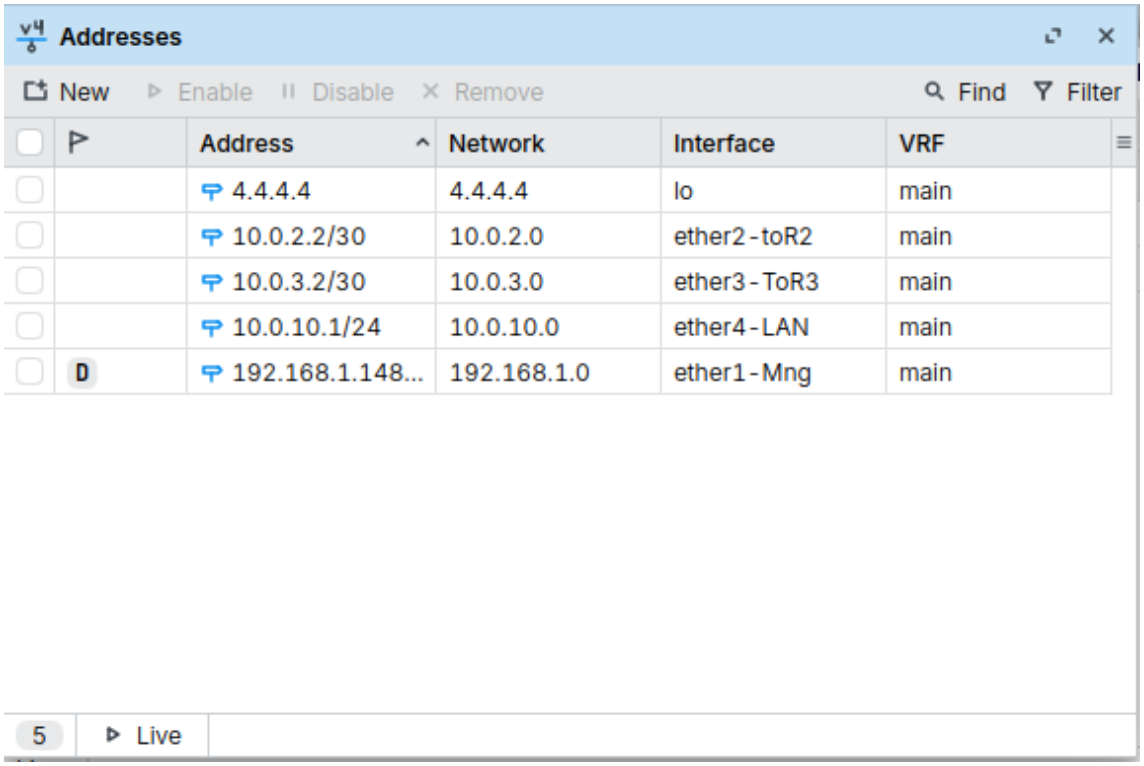

Figura 4.2: Direcciones IP asignadas a las interfaces de R4

En R4 también se ha configurado un DHCP server para poder dar direcciones IP a su red LAN.

4.1.2. Configuración del protocolo OSPF

Para configurar el protocolo OSPF es necesario crear una **instancia OSPF**. Para crearla accedemos a Routing >OSPF >Instances >New y asignamos los parámetros de la nueva instancia. En nuestro caso, la llamaremos ospf-main, usará la versión 2 (direcciones IPv4) y se le asignará manualmente la dirección ip que hemos asignado a la interfaz loopback previamente. En la imagen 4.3 podemos ver el resultado.

Posteriormente creamos un **área** (Routing >OSPF >Area >New). En esta debemos indicar el nombre de la instancia OSPF que acabamos de crear, un identificador para que cada enrutador sepa a que área o áreas pertenece y que tipo de área es. En este caso, lo dejaremos por default ya que no hay más áreas como se puede ver en la figura 4.4.

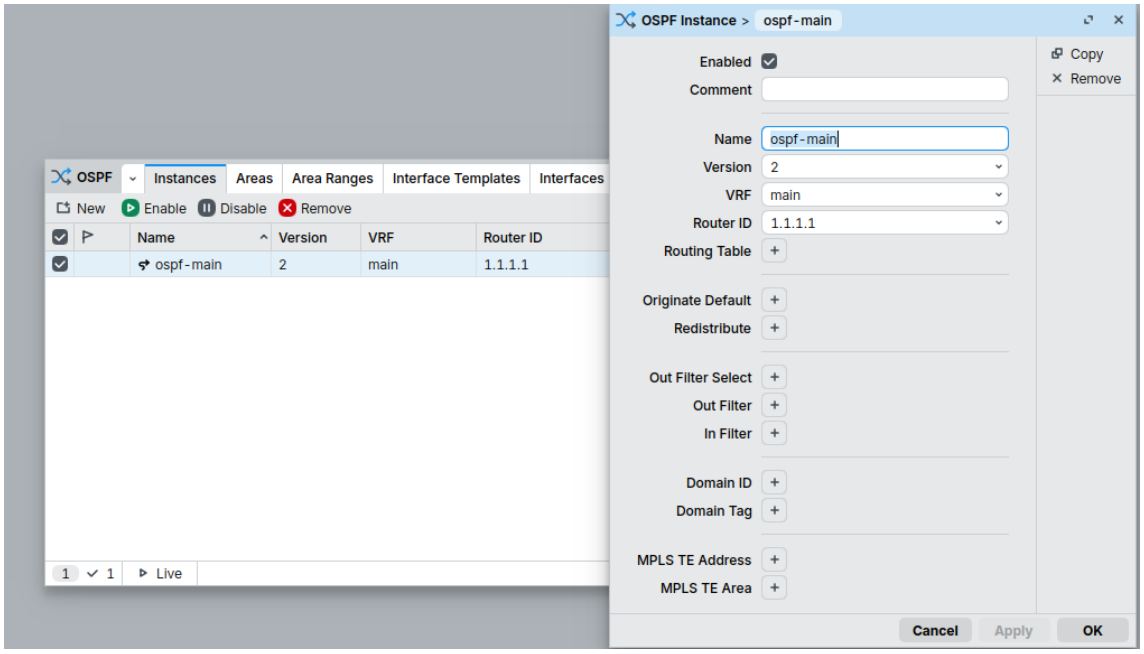

Figura 4.3: Creación de una instancia OSPF

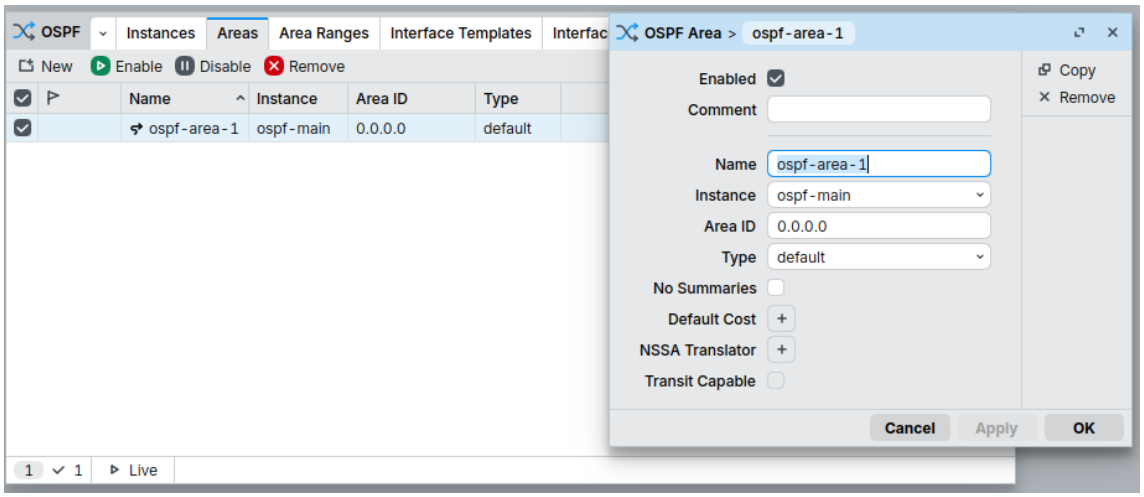

Figura 4.4: Creación de una área OSPF

Por último, para decirle al router que interfaces forman parte del protocolo OSPF y determinar sus características como el coste del enlace, el tipo de enlace o el intervalo de paquetes *hello* entre otros parámetros explicados en el capítulo de *Configuraciones del protocolo OSPF en Mikrotik*. Accederemos a Routing >OSPF >Interface Templates.

Para que los routers puedan reconocerse entre ellos como vecinos, crearemos una plantilla que indique en que interfaces tiene a sus vecinos el router que estamos configurando, como podemos en la figura 4.5.

Para el router R1 se han definido los vecinos por las interfaces ether2 y ether3. Como los routers están conectados con un enlace directo se indica que la conexión es punto a punto para reducir el tiempo del protocolo.

En el caso de R4, queremos que el resto de la organización sea capaz de acceder a nuestra red, por tanto, se debe crear una regla que permita compartir las direcciones ip. Como podemos ver en la figura 4.6 donde se establece que la todas las IPs de la interfaz

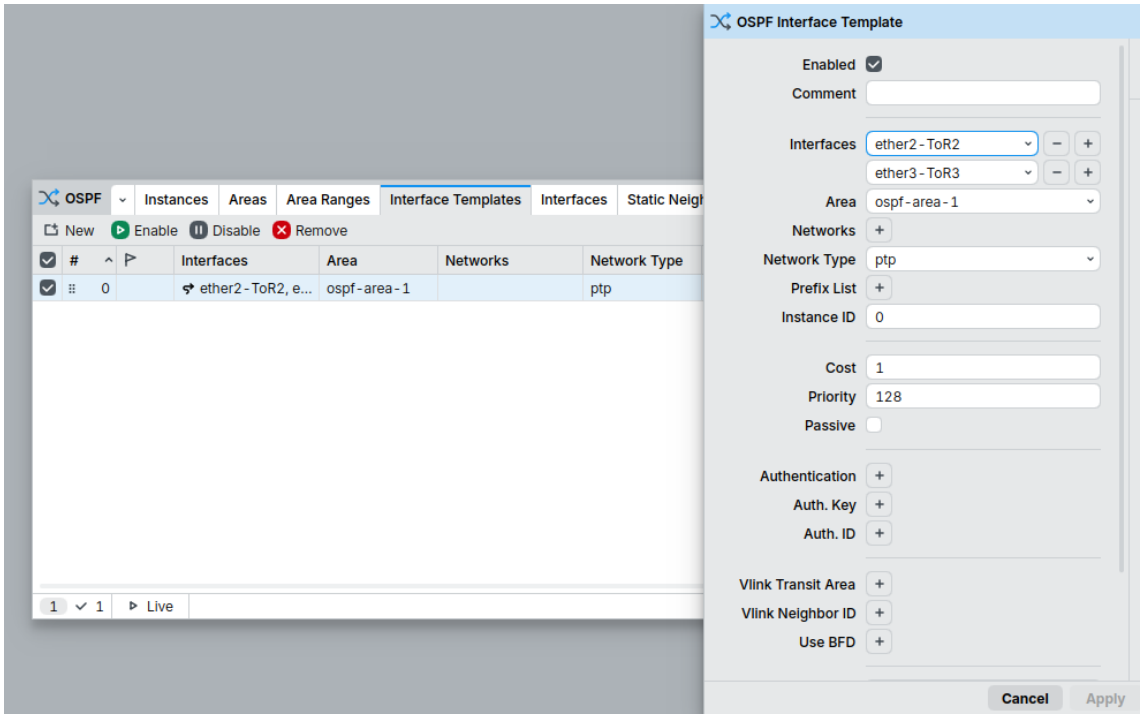

Figura 4.5: Configuración de vecinos de R1

ether4 se den a conocer mediante broadcast pero que lo hagan de manera pasiva, es decir, esta interfaz no recibirá ni enviará paquetes*hello*.

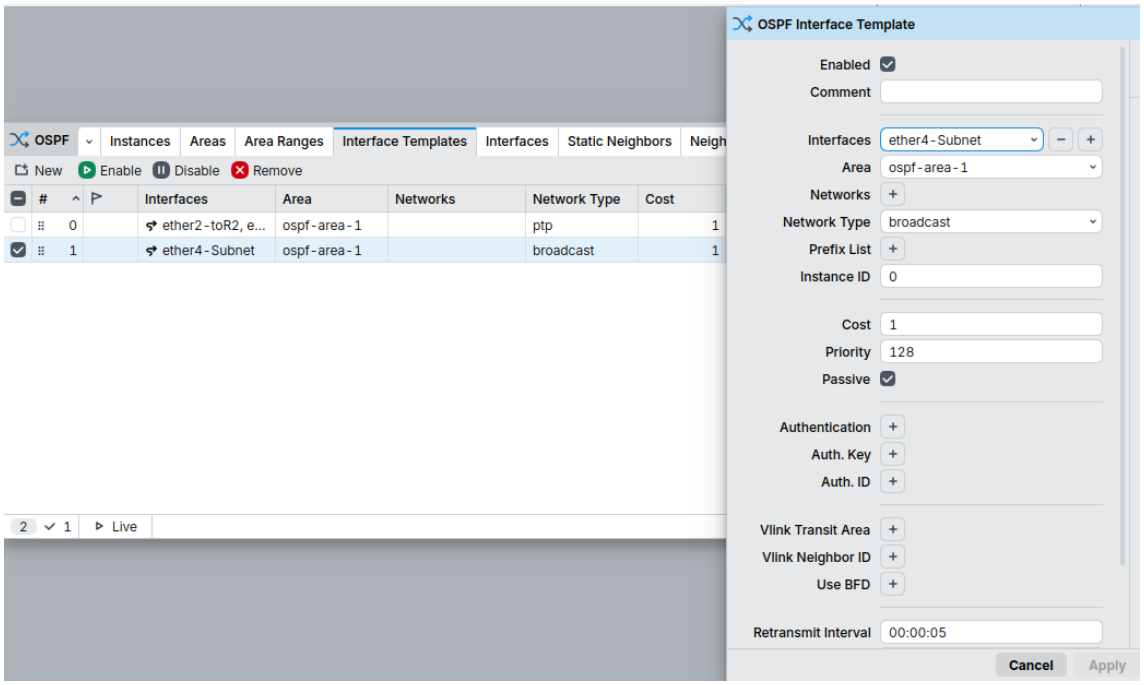

Figura 4.6: Configuración subnet de R4

4.1.3. Comprobaciones

Una vez realizada la configuración del OSPF las redes vecinas (Routing >OSPF >Neighbors) donde nos indica las direcciones IPs de las interfaces adyacentes como podemos ver en la figura 4.7

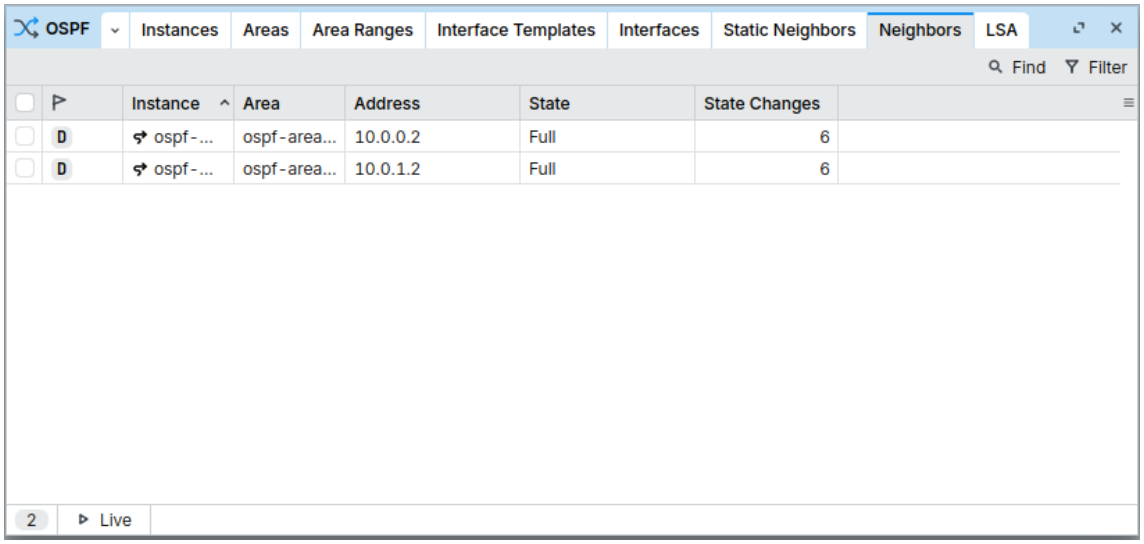

Figura 4.7: Vecinos de R1

También podemos ver los anuncios de estado de enlace (Routing >OSPF >Neighbors) que indican la ID de cada uno de los routers dentro de la misma instancia y àrea a la que pertenece el router siendo configurado. Figura4.8

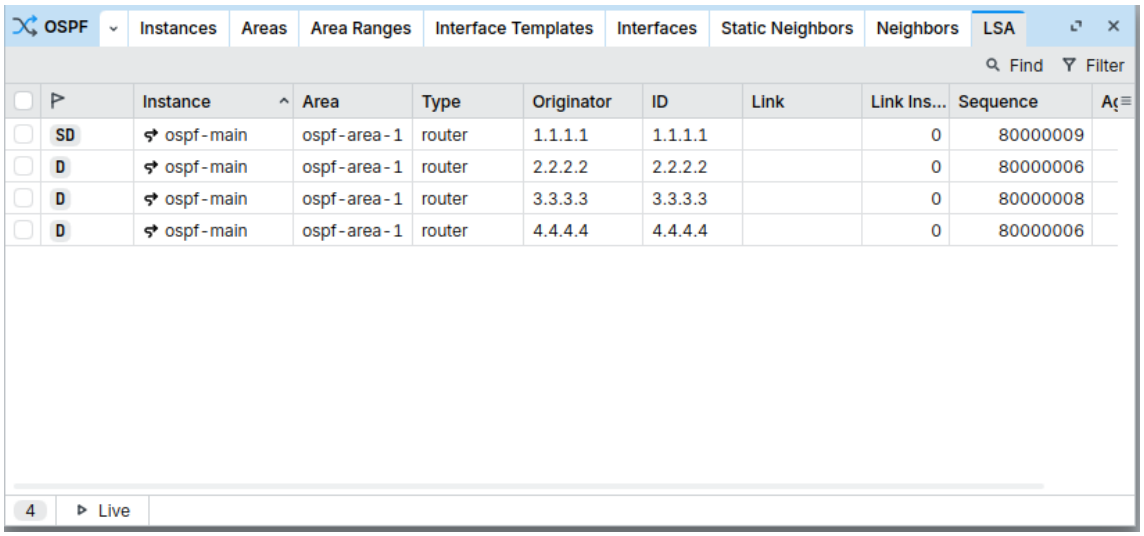

Figura 4.8: Mensajes LSA recibidos por R1

4.1.4. Aumento de costos o caída en un router

En nuestra red, R1 accede al end-point situado en la LAN 10.0.10.0/24 a través de la ruta R1-R2-R4-PC. Hay dos eventos que pueden hacer que esta ruta cambie:

- Aumenta el costo de los enlaces entre R1 y R2 o R2 y R4.
- El router R2 cae.

Al aumentar el costo el cambio es prácticamente instantáneo ya que R2 envía un LSA anunciando dicha actualización. En cambio si R2 cae, la ruta tarda un tiempo en actualizarse. Este tiempo depende del *hello interval* y el *dead interval* que hayamos configurado.

En la imagen 4.9 podemos ver una comparación al realizar un comando traceroute. En el caso de la izquierda, se estaba ejecutando el traceroute al mismo tiempo que se han cambiado los costos y en el caso de la derecha se ha forzado una caída del router, como podemos ver, en este caso ha habido una pérdida de paquetes debido al tiempo de reconexión.

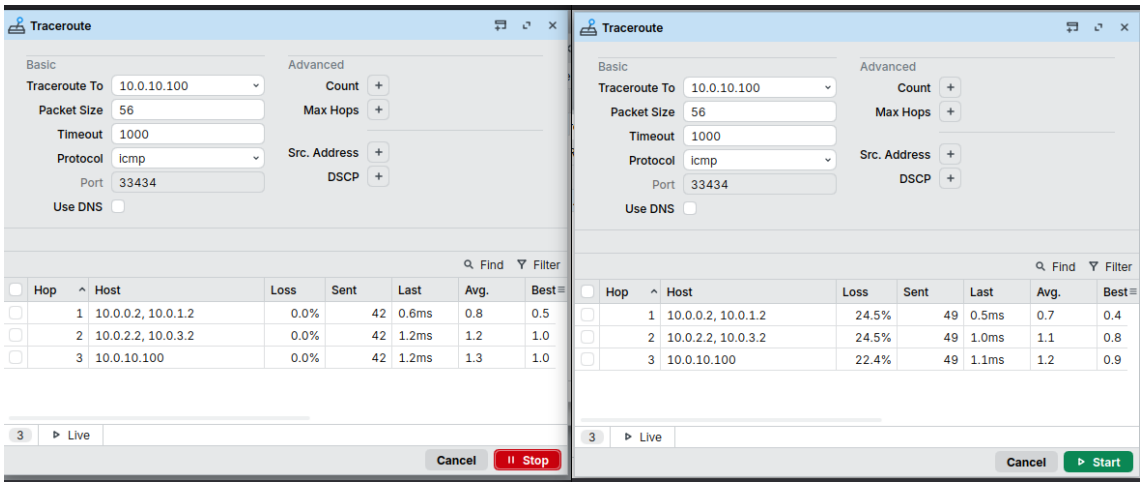

Figura 4.9: Traceroute durante un cambio de costos y una caída de router.

Al realizar este experimento, se ha podido comprobar como el algoritmo OSPF es resiliente a caídas en nuestra red. Si hay redundancia es capaz de actualizar la ruta de forma dinámica.

Por otro lado, también se ha visto la importancia de configurar correctamente los parámetros *hello interval* y el *dead interval*. Si usamos un valor muy bajo de envío de mensajes *hello* podemos reducir el tiempo ante caídas pero también inyectamos más paquetes en la red. Por otro lado, si el valor para designar a un router como caído es muy alto, el tiempo de recuperación también lo es.

4.1.5. Securización

Para esta sección, actuaremos como si R2 no fuese un router de nuestra organización y acutase como un Man-In-The-Middle con un costo inferior a R3 para tener prioridad a la hora de trazar las rutas. Para impedir esto, se van a actualizar las plantillas de R1, R3 y R4 para añadir autenticación. En la sección *Interface Templates* modificaremos las plantillas que tenemos seleccionando un método de cifrado, una contraseña y un identificador de autenticación (Auth. ID) que por defecto es 0 como se puede ver en la imagen 4.10.

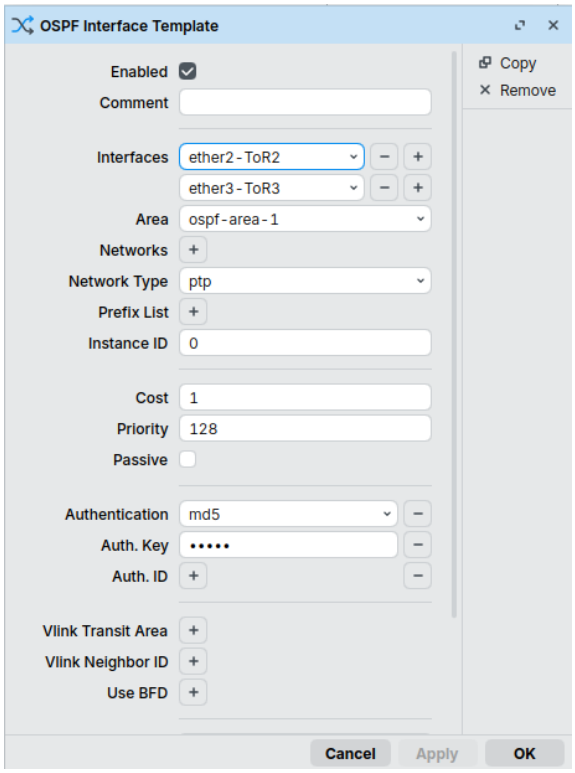

Figura 4.10: Parámetros de autenticación OSPF

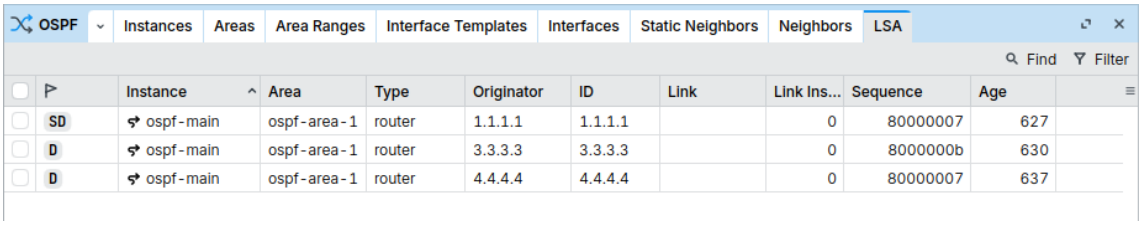

Figura 4.11: Tabla LSA de R1

Si todos los routers autorizados pertenecientes a la red tienen los mismos parámetros de autenticación, al deshabilitar y volver a habilitar la instancia OSPF para volver a iniciar el protocolo, veremos como R2 ya no es un vecino. Figura 4.11

4.1.6. Wireshark

En esta sección se analizará el tráfico capturado mediante Wireshark de la interfaz ether2 de R1 que lo conecta con el router R2. Antes de realizar la captura se ha cambiado la autenticación del protocolo de md5 a simple para recalcar lo ineficiente que es.

En primer lugar se analizará el paquete *hello* que podemos ver en la figura 4.12. En el header podemos ver información que identifica el router y el área a la que pertenece además de la contraseña que se ha utilizado ('admin'). El contenido del paquete proporciona:

- La máscara de red (/30).
- El intervalo de mensajes (10s).
- La prioridad del router.

- El router designado y su backup, en este caso 0.0.0.0 porque están en una conexión ptp. Si el tipo de conexión broadcast sería una de las IPs de la red 10.0.0.0/30 el DR y la otra el BDR (ya que no hay más routers en esa red).
- El vecino activo, como es R2 quien manda el mensaje, su vecino activo es R1 (1.1.1.1).

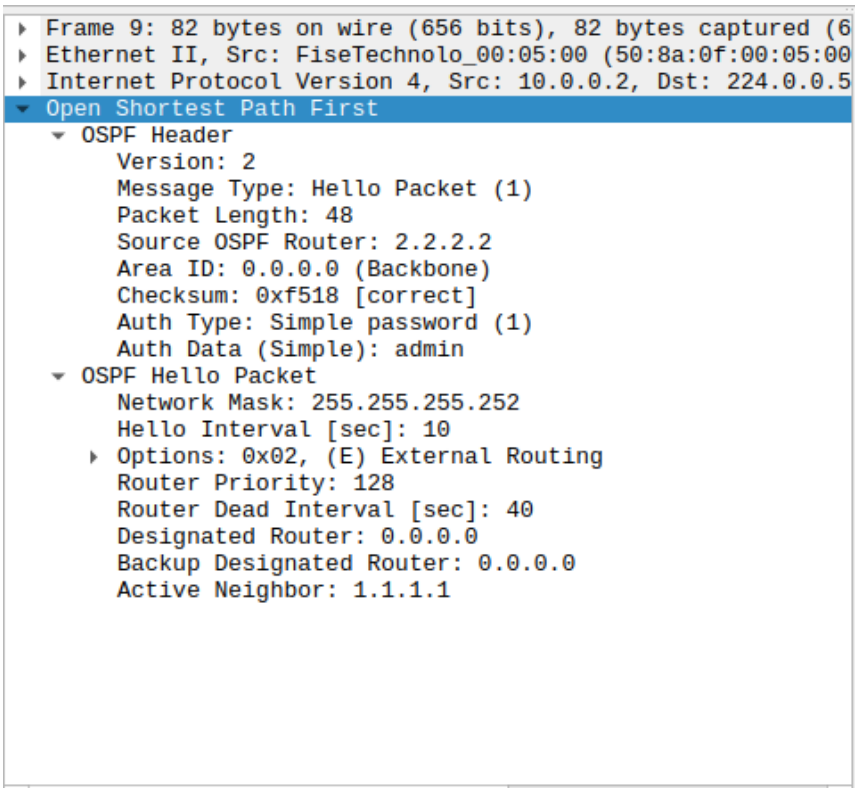

Figura 4.12: Paquete *hello* capturado mediante Wireshark

En la figura 4.13 podemos ver un conjunto de paquetes que se utilizan para la creación de la base de datos LSDB. Entre los paquetes que se muestran, podemos encontrar todos los explicados en el capítulo de explicación del protocolo OSPF. La conversación se inicia con los descriptores de la base de datos y un LS Request en el que se solicita información para completar la LSDB. Seguidamente se envía una secuencia de paquetes LSU y LSAck en la que los dos routers van completando los identificadores de routers y redes que existen en la red. Una vez ambos routers han compartido toda la información necesaria, se envían paquetes *hello* cada 10 segundos.

8	5.553955	10.0.0.1	224.0.0.5	OSPF	78 Hello Packet
9	5.953616	10.0.0.2	224.0.0.5	OSPF	82 Hello Packet
12	5.954454	10.0.0.1	10.0.0.2	OSPF	66 DB Description
13	5.954773	10.0.0.2	10.0.0.1	OSPF	66 DB Description
14	5.955140	10.0.0.1	10.0.0.2	OSPF	86 DB Description
15	5.955460	10.0.0.2	10.0.0.1	OSPF	86 DB Description
16	5.955628	10.0.0.2	10.0.0.1	OSPF	70 LS Request
17	5.955912	10.0.0.1	10.0.0.2	OSPF	66 DB Description
18	5.956090	10.0.0.1	10.0.0.2	OSPF	70 LS Request
19	5.956257	10.0.0.1	10.0.0.2	OSPF	110 LS Update
20	5.956526	10.0.0.2	10.0.0.1	OSPF	110 LS Update
21	5.956807	10.0.0.1	224.0.0.5	OSPF	122 LS Update
22	5.956810	10.0.0.2	224.0.0.5	OSPF	122 LS Update
23	6.457717	10.0.0.2	224.0.0.5	OSPF	78 LS Acknowledge
24	6.458919	10.0.0.1	224.0.0.5	OSPF	78 LS Acknowledge
25	6.964961	10.0.0.2	224.0.0.5	OSPF	122 LS Update
26	7.411709	10.0.0.1	224.0.0.5	OSPF	110 LS Update
27	7.472150	10.0.0.1	224.0.0.5	OSPF	78 LS Acknowledge
28	7.905207	10.0.0.2	224.0.0.5	OSPF	78 LS Acknowledge
30	8.665029	10.0.0.2	224.0.0.5	OSPF	134 LS Update
31	9.166405	10.0.0.1	224.0.0.5	OSPF	78 LS Acknowledge
32	9.932570	10.0.0.2	224.0.0.5	OSPF	182 LS Update
33	9.932860	10.0.0.1	224.0.0.5	OSPF	122 LS Update
34	10.424354	10.0.0.1	224.0.0.5	OSPF	78 LS Acknowledge
37	11.203453	10.0.0.1	224.0.0.5	OSPF	134 LS Update
38	11.683038	10.0.0.2	224.0.0.5	OSPF	134 LS Update
39	11.703545	10.0.0.2	224.0.0.5	OSPF	78 LS Acknowledge
41	12.185423	10.0.0.1	224.0.0.5	OSPF	78 LS Acknowledge
42	13.158393	10.0.0.1	224.0.0.5	OSPF	134 LS Update
44	13.660056	10.0.0.2	224.0.0.5	OSPF	78 LS Acknowledge
45	14.665089	10.0.0.2	224.0.0.5	OSPF	146 LS Update
46	15.166700	10.0.0.1	224.0.0.5	OSPF	78 LS Acknowledge
47	15.558239	10.0.0.1	224.0.0.5	OSPF	82 Hello Packet
48	15.959232	10.0.0.2	224.0.0.5	OSPF	82 Hello Packet
59	25.563189	10.0.0.1	224.0.0.5	OSPF	82 Hello Packet
60	25.954542	10.0.0.2	224.0.0.5	OSPF	82 Hello Packet
63	35.573011	10.0.0.1	224.0.0.5	OSPF	82 Hello Packet
64	35.953847	10.0.0.2	224.0.0.5	OSPF	82 Hello Packet
69	45.565426	10.0.0.1	224.0.0.5	OSPF	82 Hello Packet
70	45.956418	10.0.0.2	224.0.0.5	OSPF	82 Hello Packet

Figura 4.13: Captura de Wireshark donde se muestran distintos paquetes OSPF

4.2 Topología con distintas áreas

Una vez visto como se configura el protocolo OSPF, se ha virtualizado una red para ver como configurar distintas áreas. En la topología de la imagen 4.14 podemos ver una simulación de una corporación dividida en dos zonas A y B, cada una con una red LAN distinta. Para unir ambas zonas se ha creado una área de backbone en la que se incluyen los ABRs (Area Border Router) y el router de backbone.

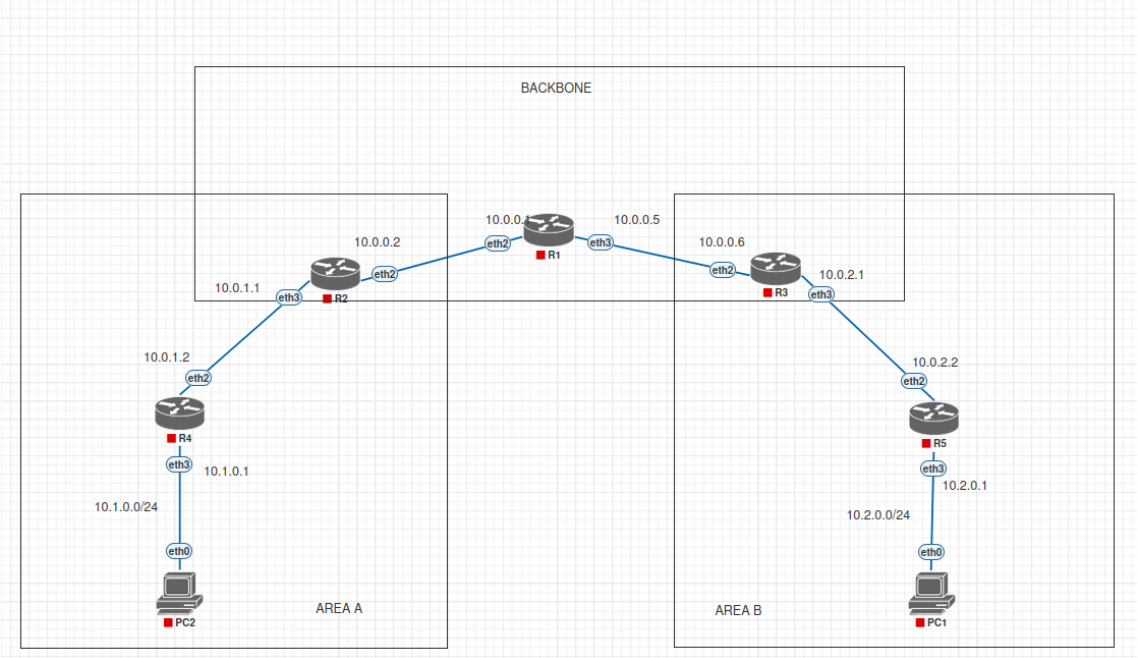

Figura 4.14: Topología con distintas áreas

4.2.1. Configuraciones

Las configuraciones son bastante similares a las de la sección anterior. Pero en este caso, creamos las siguientes áreas:

- **Área A**: Área de la sucursal donde podemos encontrar la subred 10.1.0.0/24 a la que pertenecen los routers R2 y R4.
- **Área B**: Área de la sucursal donde podemos encontrar la subred 10.2.0.0/24 a la que pertenecen los routers R3 y R5.
- **Área backbone**: Área que conecta las dos anteriores, en esta se encuentra el router de backbone R1 y los routers de frontera de área (Area Border Routers o ABR) R2 y R3.

Para cada uno de los routers deberemos asignar las IPs correspondientes a cada interfaz, si es necesario, configurar el servidor DHCP y crear una instancia del protocolo OSPF. La creación de las áreas y las plantillas de interfaces es ligeramente distinto en cada uno de los routers pero se siguen los mismos pasos que hemos explicado en la simulación anterior:

- **R1**: crear área backbone con identificador 0.0.0.0 y una plantilla que indica que las interfaces ether2 y ether3 estan conectadas punto a punto.

- **R2 (ABR)**: crear el área de backbone (0.0.0.0) y el área A con identificador 0.0.0.1 con una plantilla que indica que las interfaces ether2 y ether3 están conectadas punto a punto con R1 y R4.
- **R3 (ABR)**: crear el área de backbone (0.0.0.0) y el área B con identificador 0.0.0.2 con una plantilla que indica que las interfaces ether2 y ether3 están conectadas punto a punto R1 y R5.
- **R4**: crear el área A (0.0.0.1) con una plantilla que indique que R2 está conectado punto a punto mediante ether2 y que en la interfaz ether3 hay una red LAN (tipo de red broadcast con la casilla de passive marcacada).
- **R5**: crear el área B (0.0.0.2) con una plantilla que indique que R3 está conectado punto a punto mediante ether2 y que en la interfaz ether3 hay una red LAN (tipo de red broadcast con la casilla de passive marcacada).

4.2.2. Comprobaciones

Al crear las áreas, un dispositivo en el área A debería ser capaz de comunicarse con otro en el área B. Como podemos ver en la imagen 4.15, el PC de la red 10.2.0.0/24 es capaz de comunicarse con el PC de la red 10.1.0.0/24 estando en áreas distintas.

```
VPCS> show ip

NAME        : VPCS[1]
IP/MASK     : 10.2.0.100/24
GATEWAY     : 10.2.0.1
DNS         :
DHCP SERVER : 10.2.0.1
DHCP LEASE  : 1570, 1800/900/1575
MAC         : 00:50:79:66:68:14
LPORT       : 20000
RHOST:PORT  : 127.0.0.1:30000
MTU         : 1500

VPCS> ping 10.1.0.100

84 bytes from 10.1.0.100 icmp_seq=1 ttl=59 time=2.853 ms
84 bytes from 10.1.0.100 icmp_seq=2 ttl=59 time=2.818 ms
84 bytes from 10.1.0.100 icmp_seq=3 ttl=59 time=3.488 ms
84 bytes from 10.1.0.100 icmp_seq=4 ttl=59 time=2.381 ms
```

Figura 4.15: Comunicación entre áreas

Si observamos las tablas LSA, cada router recibe, mediante mensajes LSA, el identificador de los routers que están dentro de su misma área y las redes de las áreas externas. Como podemos ver en la imagen 4.16, R2 tiene su propio identificador y el de R4 en el área A pero en el área de backbone en cambio tiene el de los routers pertenecientes a esa área (R1 y R3). Por otra parte, en el área A aparecen las redes que estan fuera de está de la misma manera que sucede en el área de backbone.

OSPF | Instances | Areas | Area Ranges | Interface Templates | Interfaces | Static Neighbors | Neighbors | LSA

Find | Filter

Area | contains | Disable

	Instance	Area	Type	Originator	ID	Link	Link Ins...	Sequence	Age
SD	ospf-main	area-A	router	2.2.2.2	2.2.2.2		0	80000004	
SD	ospf-main	area-A	inter-area...	2.2.2.2	10.0.0.0		0	80000002	
D	ospf-main	area-A	router	4.4.4.4	4.4.4.4		0	80000004	
SD	ospf-main	area-A	inter-area...	2.2.2.2	10.0.0.4		0	80000002	
SD	ospf-main	area-A	inter-area...	2.2.2.2	10.0.2.0		0	80000002	
SD	ospf-main	area-A	inter-area...	2.2.2.2	10.2.0.0		0	80000002	
SD	ospf-main	backbone	router	2.2.2.2	2.2.2.2		0	80000004	
SD	ospf-main	backbone	inter-area...	2.2.2.2	10.0.1.0		0	80000002	
D	ospf-main	backbone	router	1.1.1.1	1.1.1.1		0	80000005	
SD	ospf-main	backbone	inter-area...	2.2.2.2	10.1.0.0		0	80000002	
D	ospf-main	backbone	router	3.3.3.3	3.3.3.3		0	80000004	
D	ospf-main	backbone	inter-area...	3.3.3.3	10.0.2.0		0	80000002	
D	ospf-main	backbone	inter-area...	3.3.3.3	10.2.0.0		0	80000001	1

13 | 13 | Live

Figura 4.16: Tabla LSA de R2

CAPÍTULO 5
Conclusión

Durante el desarrollo de este trabajo se ha llevado a cabo un estudio detallado del protocolo de enrutamiento OSPF, lo que ha permitido adquirir una comprensión tanto teórica como práctica de su funcionamiento. Mediante la implementación del protocolo en routers MikroTik y la simulación de distintos escenarios de red en un entorno virtualizado, se ha podido comprobar el comportamiento real de OSPF ante situaciones habituales en infraestructuras de red corporativas.

Las simulaciones realizadas han permitido verificar la capacidad de OSPF para calcular rutas óptimas de forma dinámica, adaptándose de manera eficiente ante cambios en la topología, como el aumento de costes en los enlaces o la caída de uno de los routers. Asimismo, se ha evidenciado la importancia de la redundancia en el diseño de redes, ya que permite mantener la conectividad y mejorar la resiliencia del sistema frente a fallos.

Por otra parte, el uso de una topología multi-área ha mostrado como utilizar un diseño jerárquico de OSPF, facilitando la segmentación lógica de una red corporativa. El papel del backbone y de los routers de frontera de área (ABR) resulta fundamental para garantizar la comunicación entre distintas zonas sin comprometer la escalabilidad del protocolo.

En cuanto a la seguridad, se ha demostrado la necesidad de implementar mecanismos de autenticación en OSPF para evitar la incorporación no autorizada de routers o dispositivos que actúen como tal capaces de alterar el cálculo de rutas o comprometer la estabilidad de la red. El análisis del tráfico OSPF mediante Wireshark ha permitido observar en detalle el intercambio de mensajes del protocolo, reforzando la comprensión de su funcionamiento interno y de los riesgos asociados a una configuración insegura.

Como conclusión final, OSPF se presenta como un protocolo de enrutamiento robusto, flexible y ampliamente utilizado en redes de tamaño medio y grande. Aunque este trabajo se centra en OSPFv2 y en escenarios IPv4, como línea de trabajo futura se podría ampliar el estudio a OSPFv3, a la redistribución de rutas con otros protocolos o a topologías de mayor escala, con el fin de profundizar aún más en su comportamiento y aplicaciones reales.