

## 9.1 Introducción

---

Los protocolos de encaminamiento se utilizan para actualizar las tablas de encaminamiento que se almacenan en los dispositivos de una red. Las tablas de encaminamiento son un conjunto de entradas dónde se define cómo hacer llegar paquetes desde un nodo origen a un nodo destino. Ya se vio en el capítulo 8 que estas tablas de encaminamiento pueden ser estáticas o dinámicas.

Las tablas de encaminamiento estáticas son aquellas cuya información es introducida manualmente por un administrador de red que se encarga de modificarlas y actualizar sus entradas. Generalmente, este tipo de tablas se emplean en redes pequeñas que no suelen cambiar frecuentemente. Por el contrario, las tablas de encaminamiento dinámicas se actualizan periódicamente usando lo que se conocen como protocolos de encaminamiento. De este modo, cada vez que se produce un cambio en la red, por ejemplo el apagado o fallo de un nodo encaminador, el fallo de un enlace, etc., los protocolos actualizan las tablas de forma automática permitiendo el enrutamiento evitando esos enlaces o nodos encaminadores.

En la práctica existen multitud de protocolos estándar de encaminamiento que emplean algoritmos de encaminamiento basados en criterios de mínimo coste como los que se van a comentar en el apartado 9.2. En general, todos los protocolos de encaminamiento cumplen las mismas funciones, que consisten en aprender y determinar cuál es el mejor camino para ir de un nodo origen a un nodo destino.

Los algoritmos de encaminamiento usados en conmutación de paquetes y redes de tipo INTERNET basan sus decisiones en criterios de mínimo coste (apartado 9.3).

## 9.2 Tipos de protocolos de encaminamiento

---

En concreto, es común tipificar los protocolos de encaminamiento en dos grandes grupos en función de su ámbito de uso. Así, los protocolos se clasifican como:

- *Protocolos de 'gateway' interior* (IGP: Interior Gateway Protocol). Son protocolos de encaminamiento que se usan para intercambiar información de enrutamiento dentro de un mismo sistema de red autónomo. Un sistema autónomo, no es más que un conjunto de redes administradas bajo un mismo dominio (es decir, por un mismo administrador). Por lo tanto, este tipo de protocolos de encaminamiento se suelen emplear en redes LAN. Los protocolos IGP que más se emplean son RIP y OSPF.
- *Protocolos de 'gateway' exterior* (EGP: Exterior Gateway Protocol). Son protocolos de encaminamiento que se usan para intercambiar información de enrutamiento entre distintos sistemas de red autónomos. Es decir, generalmente, este tipo de protocolos de encaminamiento se emplean en redes WAN. Los protocolos EGP más conocidos es BGP.

También, existen otros protocolos de encaminamiento IGP y EGP que no se van a tratar en este capítulo por tratarse de protocolos no estandarizados y propietarios de compañías como CISCO, este es el caso de IGRP y EIGRP. En la figura 9-1 se puede observar un ejemplo de ámbito de uso de protocolos IGP y EGP.

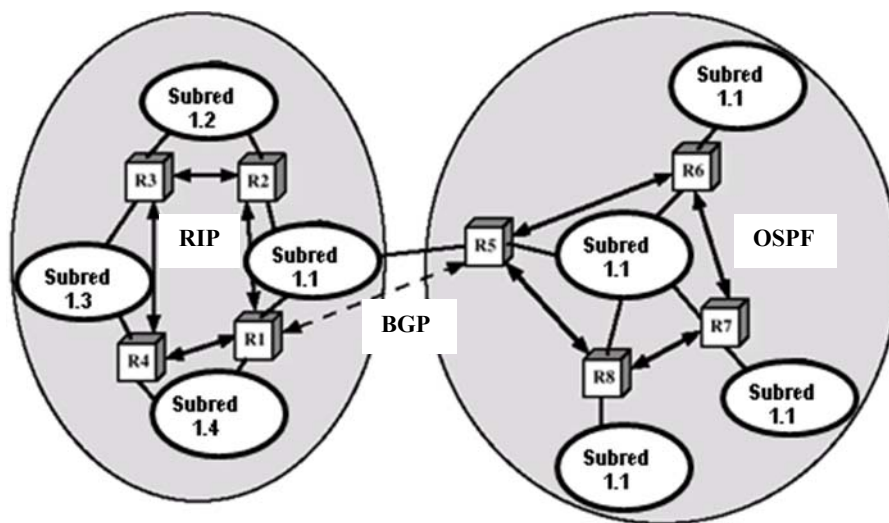


Figura 9-1. Esquema de Sistemas Autónomos empleando protocolos de distintas clases.

Además, existe otra forma de catalogar a los protocolos de encaminamiento en función de cómo se calcule la ruta óptima. Estos son tres:

- *Protocolos basados en vector de distancia.* Éste tipo de protocolos permiten calcular la distancia para llegar a un nodo destino situado en cualquier red, y la dirección necesaria para llegar hasta ese nodo destino. En estos protocolos de encaminamiento se denomina distancia a una medida de longitud, y vector significa una dirección. Existen gran diversidad de protocolos de encaminamiento estándar que están basados en vector de distancia. Algunos de los más representativos son los protocolos RIP e IGRP. En este tipo de protocolos las rutas calculadas por unos routers pueden ser empleadas por otros, ya que los distintos routers de una topología se intercambian periódicamente la información de rutas que disponen. Además, en este tipo de protocolos, todos y cada uno de ellos, se pueden emplear métodos distintos para calcular la ruta óptima. Uno de los algoritmos de cálculo de ruta más representativo basado en vector de distancia es el algoritmo de Bellman-Ford.
- *Protocolos basados en estado del enlace.* Este otro tipo de protocolo se basa en un conocimiento exacto de la topología de red sobre la que se quiere encaminar. Así, se crean tablas de encaminamiento basándose en información de la topología, a partir de paquetes denominados de estado de enlace que se intercambian todos los routers que forman la red para describir y determinar el estado de los distintos enlaces. La ventaja de estos protocolos es que cada router calcula de manera independiente la ruta óptima hacia un nodo destino. Esto permite que la ruta calculada por un router no depende de la calculada por otro. Un ejemplo de este tipo de protocolo es el protocolo OSPF. Y el algoritmo de cálculo de ruta más común es el algoritmo de Dijkstra.
- *Protocolos híbridos.* Protocolos de encaminamiento que combinan aspectos de protocolos de vector de distancia y de estado del enlace. Un ejemplo de este tipo de protocolos es el protocolo EIGRP de CISCO.

Cada uno de estos protocolos puede emplear algoritmos distintos para calcular la ruta óptima. Los algoritmos de encaminamiento más representativos son: el algoritmo de Bellman-Ford y el de Dijkstra. Ambos se comentarán en el apartado 9.3.

## 9.3 Algoritmos de cálculo de ruta

Los algoritmos que permiten calcular la ruta óptima para los distintos protocolos de encaminamiento son muy diversos, y su diversidad viene determinada por la métrica. La métrica no es más que un valor numérico, es decir un peso, que cada dispositivo de enrutamiento asigna a todas y cada una de las posibles rutas calculadas. Por lo general, este peso determina cómo de buena es cada una de las posibles rutas calculadas. La de mejor métrica será la ruta escogida como óptima para ir desde un nodo origen a uno nodo destino.

Las métricas se pueden calcular a partir de uno o varios parámetros. Así, algunos de los parámetros que estos algoritmos suelen considerar para el cálculo de la ruta óptima, son:

- *Número de saltos*: Número de ‘routers’ o encaminadores por los que pasará un paquete.
- *Ancho de banda*: Capacidad de datos del enlace. Es decir, velocidad de transmisión máxima que soporta el enlace.
- *Carga*: Número de paquetes que están gestionando un router o que está soportando un enlace en un determinado instante de tiempo.
- *Coste*: Valor arbitrario que suele basarse en un valor ponderado o no obtenido a partir del ancho de banda del enlace, el coste económico o cualquier otra medida y que es asignado por el administrador del sistema autónomo.
- *Tasa de errores*: Número de bits erróneos que se producen en cada enlace.
- *MTU*: Longitud máxima de datos de la trama de nivel de enlace que puede ser aceptada por todos los enlaces de la ruta.

### 9.2.1 Algoritmo de Dijkstra

El algoritmo de Dijkstra es un algoritmo de cálculo de ruta basado en estado del enlace. Para ilustrar cómo el algoritmo de Dijkstra calcula la ruta óptima, se va a suponer que el estado del enlace viene determinado por una métrica. De ser así, cada enlace debería tener asociado un valor numérico. Generalmente, este valor es inversamente proporcional a la capacidad del enlace y proporcional a la carga de éste o una combinación ponderada de ambos. Por lo tanto, una ruta vendrá determinada por la suma de todas las métricas de todos los enlaces por los que se pase. Y la ruta óptima, será aquella que menor métrica calculada tenga.

Además, es importante indicar que el coste de un enlace puede ser distinto en cada uno de los dos sentidos de circulación de los paquetes. Esto es así, porque en el enlace para intercambio de datos bidireccionales, parámetros como la carga no tienen porque ser similar en ambos sentidos de la comunicación.

Dada una red, el algoritmo de Dijkstra permite, por lo tanto, calcular la ruta óptima entre un nodo origen y todos y cada uno de los demás nodos que constituyen la red. El algoritmo consta de varias etapas o pasos que pueden resumirse en la figura 9.2.

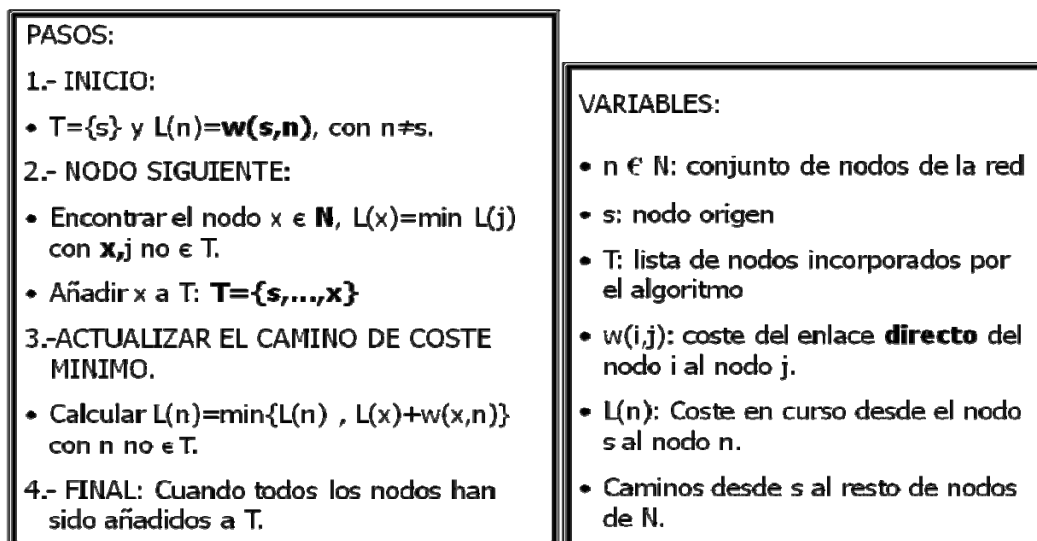


Figura 9-2. Etapas del algoritmo de Dijkstra.

Las variables que maneja el algoritmo (Figura 9.2) son: como  $N$  se define el conjunto de nodos de la red o conjunto de routers,  $s$  representa el nodo inicial origen a partir del cual se quiere calcular la ruta,  $T$  es la lista de nodos hasta los cuales se ha calculado la ruta óptima,  $w(i,j)$  define el coste del enlace entre el nodo  $i$  y el nodo  $j$ , y  $L(n)$  determina el coste mínimo (es decir el coste de la ruta óptima) desde el nodo  $s$  al nodo  $n$ . Además, es importante indicar que si los nodos  $i$  y  $j$  no se encuentran directamente conectados, entonces los costes que se asocian durante la ejecución del algoritmo son infinitos, es decir  $w(i,j)=\infty$ , y si se encuentran directamente conectados el coste será  $w(i,j) \geq 0$ . El algoritmo consiste en ir añadiendo a una lista  $T$  aquellos nodos que formen parte del camino más corto para ir desde el nodo origen hasta el nodo destino, y que no se encuentren ya incluidos en la lista (es decir que no se haya pasado por ellos). Cada vez que se añade un nodo nuevo a  $T$  se está incorporando un nuevo router a la ruta. Además,  $L(n)$  irá actualizándose con nuevo coste tras haber introducido un nuevo nodo.

### Ejemplo 9.1

#### Algoritmo de Dijkstra.

Se pretende hacer uso de algoritmo de Dijkstra para generar el camino de mínimo coste entre el nodo 1 y el resto de nodos de una red LAN, representada en la figura 9.3; de modo que la información obtenida pueda usarse como criterio para construir las tablas de encaminamiento estático para cada nodo de la red. Para ello, completa la siguiente tabla indicando la ruta y el coste mínimo de la ruta para cada iteración  $i$ .

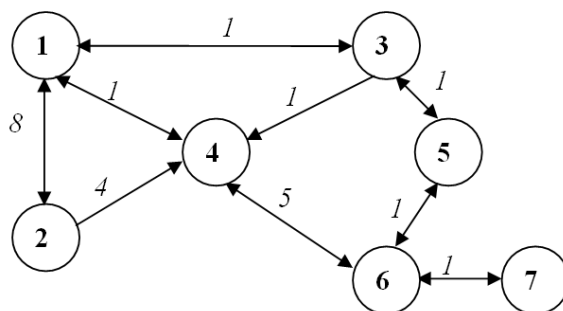


Figura 9-3. Topología de red para el cálculo de rutas por Dijkstra.

i	T	L(2)	Ruta	L(3)	Ruta	L(4)	Ruta	L(5)	Ruta
1	1	8	1-2	1	1-3	1	1-4	$\infty$	-
2	1,3	8	1-2	1	1-3	1	1-4	2	1-3-5
3	1,3,4	5	1-4-2	1	1-3	1	1-4	2	1-3-5
4	1,3,4,5	5	1-4-2	1	1-3	1	1-4	2	1-3-5
5	1,3,4,5,6	5	1-4-2	1	1-3	1	1-4	2	1-3-5
6	1,3,4,5,6,7	5	1-4-2	1	1-3	1	1-4	2	1-3-5
7	1,2,3,4,5,6,7	5	1-4-2	1	1-3	1	1-4	2	1-3-5

L(6)	Ruta	L(7)	Ruta
$\infty$	-	$\infty$	-
$\infty$	-	$\infty$	-
6	1-4-6	$\infty$	-
3	1-3-5-6	$\infty$	-
3	1-3-5-6	4	1-3-5-6-7
3	1-3-5-6	4	1-3-5-6-7
3	1-3-5-6	4	1-3-5-6-7

## 9.2.2 Algoritmo de Bellman-Ford

El algoritmo de Bellman-Ford es un algoritmo de cálculo de ruta basado en camino más corto. En el caso de Bellman-Ford, el cálculo de la ruta óptima se reduce a calcular el camino más corto entre un nodo origen y un nodo destino. Por lo tanto, en este caso no se suele hablar de coste asociado a un enlace, ya que normalmente la métrica que se asocia a un enlace es uno. Al final, el camino más corto tendrá como coste la suma de todas las métricas de cada uno de los enlaces que, por lo general, no será otra que el número de enlaces por los que se ha pasado.

El algoritmo comienza calculando el camino más corto entre un nodo y todos los demás con la única exigencia de que los caminos calculados contengan al menos un enlace. Posteriormente, se repite la misma iteración pero suponiendo que la condición que se exige sea de al menos dos enlaces, y a continuación se volverá a repetir suponiendo tres enlaces, y así sucesivamente.

Por lo tanto, los pasos básicos para calcular el camino más corto se van a ilustrar sobre el ejemplo de la figura 9.4:

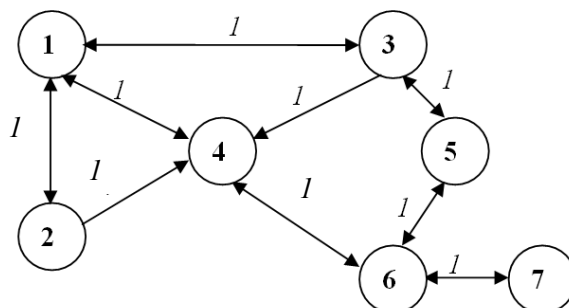


Figura 9-4. Topología de red para el cálculo de rutas por Bellman-Ford.

Paso 1: Inicialmente, cada nodo de la red se inicializa con su una tabla de información que contiene como ir de ese nodo origen a todos los demás nodos de la red. Así, para el ejemplo de la figura 9.5 se muestra la información de inicialización para cada uno de los nodos. En cada nodo, se inicializa el coste del camino a métrica 0 si el nodo destino es el mismo que el origen y se inicializa a métrica,  $\infty$ , si no hay un camino directo entre el nodo origen y el nodo destino. El resto de caminos entre el nodo origen y el nodo destino tendrán como coste 1. En la información de las tablas se almacena cuál es el nodo destino (Des) y cuál es el nodo siguiente para alcanzar ese destino (Sig).

Des	Sig	M.	Des	Sig	M.	Des	Sig	M.	Des	Sig	M.	Des	Sig	M.
1	-	0	1	-	1	1	-	1	1	-	1	1		$\infty$
2	-	1	2	-	0	2		$\infty$	2	-	1	2		$\infty$
3	-	1	3		$\infty$	3	-	0	3	-	1	3	-	1
4	-	1	4	-	1	4	-	1	4	-	0	4		$\infty$
5		$\infty$	5		$\infty$	5	-	1	5		$\infty$	5	-	0
6		$\infty$	6		$\infty$	6		$\infty$	6	-	1	6	-	1
7		$\infty$	7		$\infty$	7		$\infty$	7		$\infty$	7		$\infty$

Nodo '1'	Nodo '2'	Nodo '3'	Nodo '4'	Nodo '5'
----------	----------	----------	----------	----------

Des	Sig	M.	Des	Sig	M.
1		$\infty$	1		$\infty$
2		$\infty$	2		$\infty$
3		$\infty$	3		$\infty$
4	-	1	4		$\infty$
5	-	1	5		$\infty$
6	-	0	6	-	1
7	-	1	7	-	0

Nodo '6'	Nodo '7'
----------	----------

Figura 9-5. Inicialización de las tablas para encaminamiento basado en vector de distancia y Actualización tabla de encaminamiento Nodo '1'.

Paso 2: Se actualizan las tablas de información de rutas calculando cual es la ruta más corta para ir de un nodo origen a un nodo destino considerando dos enlaces (Figura 9-6). Veamos, por ejemplo, cuál es la ruta más corta para ir desde el nodo '1' de la Figura 9.6, a un nodo destino predeterminado empleando dos enlaces. En primer lugar, el nodo '1' consultará en sus tablas cuales son los enlaces adyacentes, que en este caso serían '2', '3' y '4'. A continuación se consultan en las tablas de '2', '3' y '4' cuáles son sus respectivos nodos adyacentes (de 2: '1' y '4', de 3: '1', '4' y '5' y de 4: '1', '2', '3' y '6'). Por lo tanto, se puede determinar que el nodo '1' puede alcanzar a los nodos '5' y '6' a través de dos enlaces. Así observando las tablas de información se observa que la ruta del nodo '1' hasta el nodo '3' empleando dos enlaces tiene métrica 2. Lo mismo ocurre para ir del nodo '1' a los nodos '2' y '4'. Por lo tanto, como en estos casos la métrica obtenida al recorrer dos enlaces es mayor que la que se tenía almacenada en la tabla de la figura 9.5, la ruta y la métrica no se actualiza y se mantienen las del paso 1. Por otro lado, para ir desde el nodo '1' al nodo '5' y '6' con dos enlaces, también, se emplea una métrica de 2 y, sin embargo en este caso si se actualizan las rutas almacenadas en la tabla de la figura 9.5, ya que la métrica obtenida es menor que la almacenada inicialmente. En un principio, para ir del nodo '1' a los nodos '5' y '6' se tenía una métrica  $\infty$ .

Paso 3: Este proceso se repite iterativamente tantas hasta que se alcanza el nodo más alejado.

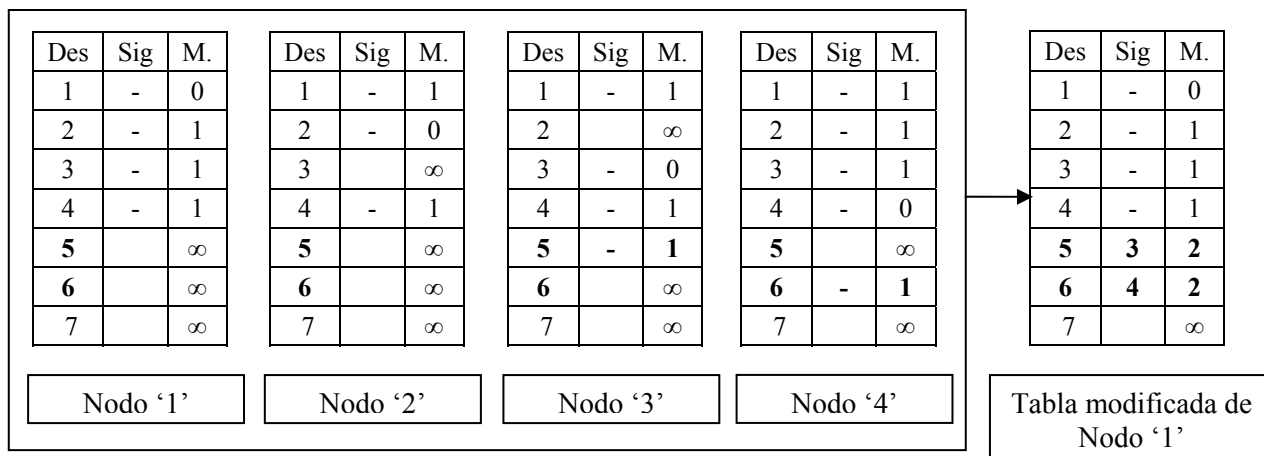


Figura 9-6. Actualización de la tabla de rutas del nodo '1'.

En la figura 9.6, se representa el proceso de actualización. En un instante determinado la tabla del Nodo '1', se actualiza con la información que le proporcionan los nodos '3' y '4'. Un ejemplo práctico de este algoritmo es el protocolo estándar RIP que emplea este algoritmo para calcular las rutas y actualizarlas.

## 9.4 Protocolos IGP: OSPF

El protocolo OSPF (Open Shortest Path First) o protocolo abierto del primer camino más corto, data de finales de los años 80. Este protocolo fue diseñado como protocolo de encaminamiento para solventar algunos de los problemas que el protocolo RIP no contemplaba. Actualmente la versión del protocolo OSPF que se emplea es la versión 2 cuando se usa IPv4, u OSPF versión 3 para IPv6. En cualquier caso, OSPF es el protocolo de encaminamiento más empleado en grandes redes LAN. Además, OSPF permite encriptación con el algoritmo MD5 para autentificar a los nodos antes de actualizar y crear nuevas rutas. También, es importante destacar, que a diferencia de RIP, OSPF no requiere protocolos de nivel de transporte para encapsular los datos de encaminamiento, y que éste opera directamente sobre IP (RIP encapsula sobre paquetes de transporte UDP). La normativa de IETF que define la implementación del protocolo OSPF está publicada en la norma RFC 2328 y 2740, para las versiones 2 y 3 respectivamente. Estas versiones datan de finales de los 90.

OSPF es un protocolo basado en estado del enlace. La principal diferencia entre OSPF y RIP viene determinada por cómo ambos protocolos comunican actualizaciones de rutas. Mientras que RIP permite a un nodo encaminador enviar sus propias rutas únicamente a sus nodos vecinos adyacentes, el protocolo OSPF permite a un nodo enviar publicaciones del estado del enlace, a cualquier nodo perteneciente a la misma jerarquía (Ver más adelante el concepto de área). En ambos casos se emplean paquetes multidifusión de IP. En OSPF la dirección IP de multidifusión es 224.0.0.5.

### 9.4.1 Acciones OSPF para construir encaminamiento

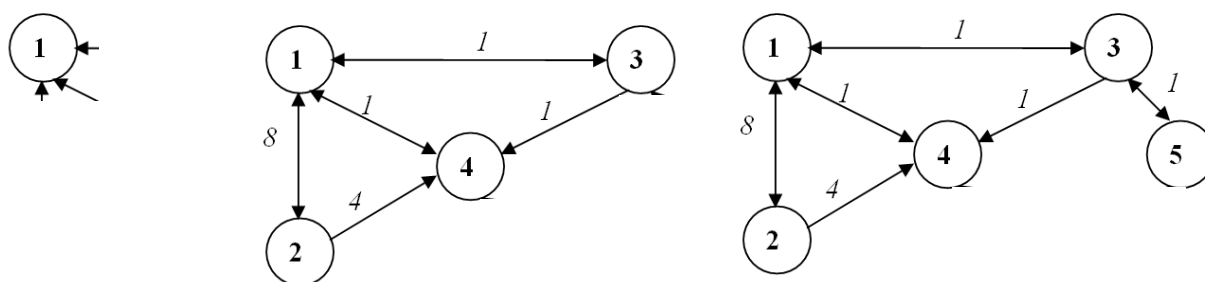
En el encaminamiento basado en estado del enlace, se necesita de un conjunto de acciones que permitan asegurar que cada nodo encaminador tenga una tabla de encaminamiento que disponga de información del nodo de menor coste a partir del cual se alcanza al resto de nodos. Estas acciones se pueden resumir en:

- *Crear estados de los enlaces* a los que conecta cada nodo (Figura 9.3). A partir de estos estados cada nodo construye paquetes LSA (Link-State Advertisement) o paquetes de estado de sus enlaces. Estos paquetes se emplean para que los nodos se intercambian la información de los enlaces a los que están conectados. El paquete de estado de enlace, LSA, suele transportar gran cantidad de información. Por lo general:
  - La identidad de cada nodo y la lista de enlaces a los que está conectado para poder definir la topología de la red del dominio.
  - El número de secuencia y la edad para distinguir los nuevos paquetes de los viejos.

Además, los paquetes LSA sólo se crean cuando se dan alguna de estas situaciones:

- Cambios en la topología de red del dominio. Por lo tanto, se requiere que cada nodo encaminador se adapte a la nueva topología. En este caso los paquetes LSA los generan los nodos que conectan enlaces nuevos o enlaces que han sido modificados. Esos paquetes se enviarán, después, a nodos adyacentes.
  - Siempre de forma periódica. Cada intervalo de tiempo comprendido entre 60 minutos y 2 horas, dependiendo de la configuración. Se crean paquetes LSA que permitan actualizar las entradas de los encaminadores. De este modo se elimina información vieja que pueda quedar almacenada.
- *Enviar paquetes LSA a cada nodo adyacente mediante el método de inundación* por cada interfaz de salida, generalmente cada 10 segundos. Cada nodo que recibirá paquetes LSA, los compara con la copia que pueda tener almacenada, descartando el que sea más viejo de los dos (número de secuencia más pequeño). El más nuevo lo almacena y envía copia de nuevo por cada interfaz por el método de inundación. Una vez se hayan enviado los paquetes LSA, cada nodo tendrá una copia de toda la topología completa, esta copia constituye una base de datos de los enlaces de la misma (LSD, Link-State Database).
  - *Formar un árbol del camino más corto para cada nodo* (Figura 9.5). Puesto que la topología completa almacenada en cada nodo a partir de los LSA no es suficiente para determinar cuál es el camino más corto, el algoritmo de Dijkstra permite determinar el camino mínimo entre un nodo encaminador (nodo raíz en el árbol) y el resto de nodos encaminadores (nodos hojas en el árbol). El árbol representa la ejecución del algoritmo de Dijkstra.
  - *Calcular la tabla de encaminamiento* basada en el árbol de camino más corto. Los nodos encaminadores que tienen activado el protocolo OSPF almacenan información sobre el estado de los enlaces y ejecutan el algoritmo de Dijkstra para calcular la ruta más corta de un nodo a cualquier otro. Así a partir del árbol de Dijkstra para cada nodo, se construye una tabla de encaminamiento empleando la ruta de camino más corto para ir de un nodo a otro.





Establacer el nodo '1' como raíz y moverlo a la lista T. Mover '1' a la RUTA y añadir 2, 3 y 4. Mover 3 a la RUTA y añadir 5 a la lista T.

Figura 9-5. Ejemplo de construcción del árbol de camino mínimo obtenido a partir de Dijkstra.

OSPF para realizar un encaminamiento más eficiente, divide un sistema autónomo (red con un mismo administrador) en unidades jerárquicas de menor tamaño. Estas unidades se conocen con el nombre de áreas. Un *área* no es más que un subconjunto de nodos encaminadores y estaciones conectados dentro de un mismo sistema autónomo. Por lo tanto, un área es una red más pequeña que constituye un subconjunto de aquella a partir de la cual partió la división. Los cambios que se produzcan en la topología de un área no afectan al rendimiento de otras áreas, puesto que OSPF gestiona el encaminamiento de cada área de forma independiente.

Para asignar la métrica o coste de los enlaces a cada nodo encaminador, se pueden emplear diferentes criterios: mínimo retardo, máxima productividad, etc. Además, es frecuente construir varias tablas de encaminamiento para un mismo nodo encaminador empleando cada uno de esos diferentes criterios. Por lo general, es frecuente usar como métrica de un enlace, un valor numérico que se calcula en base al ancho de banda del interfaz.

## 9.4.2 OSPF encaminando a través de múltiples áreas

Cuando un sistema autónomo se jerarquiza y se divide en múltiples áreas para facilitar el encaminamiento por OSPF, cada área puede ser de uno de estos tres tipos o categorías:

- **Backbone:** Constituye el núcleo central de una red encaminada con OSPF. Este área se caracteriza porque mantiene una conexión física o lógica con todas y cada una de las demás áreas en las que se ha jerarquizado la red.
- **Stub:** Hace referencia a todas aquellas áreas que no pueden recibir rutas procedentes de redes externas. Se consideran redes externas todas aquellas que son gestionadas por protocolos de encaminamiento distintos a OSPF y están fuera de la jerarquía de red.
- **Not-so-stubby (NSSA):** Son áreas stub que pueden importar rutas externas y enviarlas a un área de backbone, pero que sin embargo no puede exportar rutas externas procedentes del backbone hacia otras áreas.

Además, se pueden diferenciar los siguientes tipos de nodos encaminadores:

- **Nodo encaminador Interior:** Son aquellos en los que todos sus interfaces pertenecen a una misma área de la jerarquía, y además la base de datos de estados de enlace (LSD) es idéntica.

- Nodo encaminador de Backbone: Son aquellos en los que al menos un interface pertenece al área de Backbone, también conocida como área cero.
- Nodo encaminador de Frontera (ABR: Area Border Router): Son los encaminadores que tienen interfaces conectados a diferentes áreas.

## 9.5 Protocolos EGP: BGP

---

El protocolo BGP (Border Gateway Protocol) o protocolo de pasarela de frontera es un protocolo que se diseñó para encaminamiento entre diferentes sistemas autónomos. Este protocolo es empleado por los proveedores de servicio de Internet, ISPs. BGP no se basa en el estado del enlace sino en el vector de distancia y fue creado a finales de los 80. Actualmente, la única versión en uso del protocolo BGP es la versión 4. Las normas que determinan el funcionamiento de esta versión están publicadas en RFC 4271. Esta versión funciona en Internet desde mediados de los años 90.