

ENCAMINAMIENTO DE LOS MENSAJES (*ROUTING*)

Para llegar a su destino, los paquetes deben atravesar la red de comunicación. Pero, ¿cuál es el camino para llegar al destino? Tenemos que resolver dos problemas. Por un lado, cómo indicar la dirección de destino del paquete, y, por otro, qué camino elegir entre los múltiples caminos que enlazan el nodo origen y el destino. En todo caso, y salvo que tuviéramos que resolver algún problema particular, el camino escogido debe ser siempre un camino de distancia mínima (los paquetes nunca se alejan de su destino).

Veamos cómo resolver estos problemas.

1.- El registro de encaminamiento

Cuando se utiliza conmutación de paquetes, el procesador (más exactamente, el interfaz con la red) inyecta los paquetes a transmitir en la red para que éstos lleguen a su destino utilizando por el camino los diferentes recursos de la red (de manera similar a lo que ocurre con el correo habitual). Para llegar desde el nodo origen al destino hay que utilizar la información que lleva el propio paquete. ¿Por dónde avanzar en la red? El proceso de hacer avanzar un paquete en la red se conoce como **encaminamiento** (*routing*). El encaminamiento de los paquetes en la red debe ser sencillo. Dos son las opciones más habituales.

- En la cabecera del paquete se incluye la dirección de destino; así, en cada encaminador de mensajes intermedio se decidirá por dónde debe avanzar el paquete en función de dicha dirección. Para ello pueden usarse dos técnicas diferentes: (a) utilizar una tabla que indique los puertos de salida adecuados para cada dirección de destino; o (b) efectuar una operación sencilla con la dirección que dé como resultado el puerto de salida.
- La cabecera del paquete indica el número de pasos que debe dar el paquete en cada dimensión para llegar al destino, en un campo que se conoce como **registro de encaminamiento** (*routing record*). En cada paso se actualiza el registro de encaminamiento (± 1), y cuando todos sus elementos valgan 0, el paquete ha llegado a su destino.

De una manera o de otra, la elección del camino debe ser lo más rápida posible, ya que no podemos demorar demasiado tiempo los paquetes en los encaminadores intermedios, so pena de que la latencia de la comunicación crezca mucho. Ya que los registros de encaminamiento son, probablemente, la opción más utilizada, veamos cómo calcular el registro de encaminamiento en las topologías de red que hemos analizado en el apartado anterior.

- a) Mallas.** Los procesadores que forman la malla se suelen identificar según sus coordenadas en la misma. En una malla de n dimensiones (k nodos por dimensión), para ir del nodo $X(x_{n-1}, x_{n-2}, \dots, x_1, x_0)$ al nodo $Y(y_{n-1}, y_{n-2}, \dots, y_1, y_0)$ hay que dar $y_i - x_i$ pasos en la dimensión i (si es positivo, en una dirección, y si es negativo, en la contraria). Justamente ésta es la información que se añade en el registro de encaminamiento:

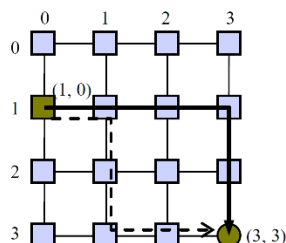
$$RE = [y_{n-1} - x_{n-1}, y_{n-2} - x_{n-2}, \dots, y_1 - x_1, y_0 - x_0]$$

Veamos un ejemplo en dos dimensiones:
del nodo (1, 0) al nodo (3, 3):

$$RE_0 = 3 - 0 = 3$$

$$RE_1 = 3 - 1 = 2$$

$$RE = [2, 3]$$



Según va avanzando el paquete por la red, hay que ir actualizando el registro de encaminamiento en los encaminadores intermedios; si en el ejemplo anterior se escoge el camino remarcado en la figura, el registro de encaminamiento irá modificándose de la siguiente manera en cada paso:

$$\begin{array}{c} [2, 3] \rightarrow [2, 2] \rightarrow [2, 1] \rightarrow [2, 0] \\ \downarrow \\ [1, 0] \\ \downarrow \\ [0, 0] \end{array}$$

En cada “paso” en la red se actualiza un elemento (el de la dimensión correspondiente) del registro de encaminamiento en una unidad. Cuando todos los elementos son 0, el paquete ha llegado a su destino.

- b) Toros.** El procedimiento es el mismo, pero en este caso hay que tener en cuenta que tenemos un anillo en cada dimensión. Y en un anillo siempre hay dos caminos para ir de un punto a otro: por “delante” o por “detrás”; eso sí, escogiendo siempre el camino mínimo. Por tanto, en un anillo de k nodos no se dan nunca más de $d = k/2$ pasos en una dirección, ya que siempre es posible efectuar $k - d$ pasos en la dirección contraria, es decir, recorrer una distancia menor.

Así pues, el registro de encaminamiento se calcula como en el caso anterior, pero se efectúa luego una fase de corrección para buscar el camino mínimo. En un caso general, para ir de $X(x_{n-1}, x_{n-2}, \dots, x_1, x_0)$ a $Y(y_{n-1}, y_{n-2}, \dots, y_1, y_0)$ el registro de encaminamiento se calcula de la siguiente manera:

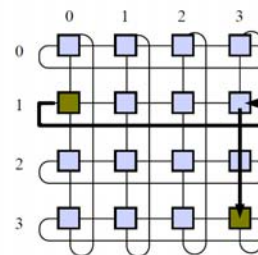
```

for i = 0 to n-1 do           ; para todas las dimensiones
begin
  REi = Yi - Xi             ; elemento i del reg. de encaminamiento
  if (|REi| > k/2) then       ; corregir el camino, es muy largo
    if (REi > 0)
      then REi = REi - k
    else REi = REi + k
end

```

Por ejemplo, en dos dimensiones:

del nodo (1, 0) al nodo (3, 3)

$$\begin{aligned} RE_0 &= 3 - 0 = 3 \\ 3 &> 2 \rightarrow 3 - 4 = -1 \\ RE_1 &= 3 - 1 = 2 \\ RE &= [2, -1] \end{aligned}$$


Como en el caso de las mallas, el registro de encaminamiento se va actualizando en cada paso en la red; en el ejemplo: $[2, -1] \rightarrow [2, 0] \rightarrow [1, 0] \rightarrow [0, 0]$.

- c) Hipercubos.** También en este caso el encaminamiento de los paquetes es sencillo. El hipercubo es un cubo binario, es decir, tiene sólo dos nodos en cada dimensión, con etiquetas 1 y 0. La etiqueta de cada nodo es simplemente el conjunto de etiquetas en cada dimensión.

Para ir de i a j hay que cruzar las dimensiones en las que las etiquetas de cada nodo son diferentes. Por ejemplo, para ir del nodo (0001) al nodo (0100) hay que dar dos pasos en la red, en la primera y en la

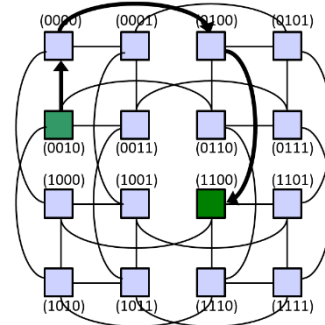
tercera dimensión, ya que las direcciones sólo se diferencian en esos bits. Por tanto, el registro de encaminamiento se calcula simplemente efectuando una operación *xor* bit a bit entre las etiquetas del nodo destino y origen:

$$RE = [\text{origen } \text{xor} \text{ destino}].$$

Por ejemplo, para ir del nodo (0010) al nodo (1100):
 $RE = 0010 \text{ xor } 1100 = [1110]$

Hay que dar tres pasos, en las dimensiones 2, 3 y 4, tal como aparece en la figura. Con cada paso que se efectúa se pone a 0 el bit correspondiente del registro de encaminamiento:

$$[1110] \rightarrow [1100] \rightarrow [1000] \rightarrow [0000]$$



- d) Árboles.** También en este caso es sencillo el encaminamiento de los paquetes. En primer lugar, habrá que decidir hasta qué nivel del árbol hay que subir (mediante una comparación de las direcciones origen y destino), y, después, por qué rama hay que descender para llegar al destino (normalmente utilizando la dirección destino). Dejamos a modo de ejercicio cómo generar y utilizar el registro de encaminamiento en los árboles.

2.- Elección del camino: estático o adaptativo

Cuando un paquete llega a un encaminador de la red se analiza la cabecera del mismo para saber si va destinado al procesador local o a otro procesador, y para ello se utiliza el registro de encaminamiento. Cuando todos los campos del registro de encaminamiento son 0, entonces el paquete ha llegado al destino; si no, debe continuar hacia adelante.

Aunque el registro de encaminamiento esté perfectamente definido, el camino entre origen y destino no es único, y, en general, dispondremos de muchas alternativas. Por ejemplo, si hay que dar dos pasos en ambas direcciones de una malla — $RE = [2, 2]$ —, tenemos 6 posibilidades diferentes de recorrer el camino: x-x-y-y, x-y-x-y, x-y-y-x, y-x-x-y, y-x-y-x, y-y-x-x. ¿Qué camino hay que seguir? ¿Hay que seguir siempre el mismo camino?

- **Encaminamiento estático**

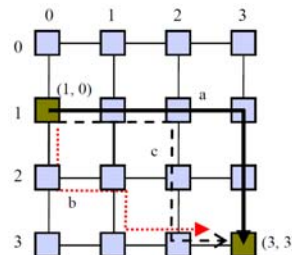
El encaminamiento de los paquetes es estático si todos los paquetes que van del nodo i al nodo j utilizan siempre el mismo camino, sea cual sea la situación de tráfico en la red. La elección más habitual se conoce como **DOR** (*dimension order routing*) o, en el caso de dos dimensiones, como "primero-X-luego-Y". Los paquetes avanzan hacia el destino recorriendo primero todo el camino en una dimensión, luego en la siguiente, etc. Por tanto, en el encaminamiento estático sólo se utiliza uno de todos los caminos posibles para ir del nodo i al nodo j .

Dado que se utiliza un sólo camino, se pierde la tolerancia a fallos de bajo nivel; si no se puede avanzar, porque hay una avería, habrá que utilizar algoritmos de otro nivel para que la comunicación se pueda efectuar.

- **Encaminamiento dinámico o adaptativo**

El encaminamiento es dinámico si los paquetes que van de i a j no siguen siempre el mismo camino, sino que escogen el camino más adecuado en función de la situación concreta de la red. El encaminamiento dinámico es más flexible, y permite utilizar caminos en los que la densidad de tráfico sea menor para evitar zonas de la red que estén temporalmente saturadas.

En la siguiente figura se muestran tres opciones (con línea continua la correspondiente al encaminamiento DOR) para poder ir del nodo $(1, 0)$ al nodo $(3, 3)$. En los tres casos el registro de encaminamiento es $RE = [2, 3]$. Si, por ejemplo, hubiera una congestión de tráfico alrededor del nodo $(1, 1)$, podría ser más adecuado utilizar el camino (b) en lugar del (a). La misma idea se puede aplicar para evitar fallos en la red: aunque se estropee el enlace $(1, 2) \rightarrow (1, 3)$, todavía tenemos posibilidades de llegar al nodo $(3, 3)$, utilizando, por ejemplo, el camino (c).



Existen muchas maneras diferentes de efectuar la adaptación. Una de las más conocidas es la denominada *zigzag*: siempre que sea posible, los paquetes no agotan el recorrido en una dimensión, sino que los van intercambiando. De esa manera se consigue mantener hasta el final la posibilidad de utilizar caminos alternativos frente a posibles problemas de tráfico o de fallos en la red. Para el caso de dos dimensiones, un ejemplo de algoritmo simplificado podría ser el siguiente:

```

si (|REx| > |REy|)
    entonces Sigue_por_el_eje_X           ; X+, X-
    si_no     Sigue_por_el_eje_Y           ; Y+, Y-

si (salida_elegida_ocupada)
    entonces Intenta_la_otra               ; si hay alternativa

```

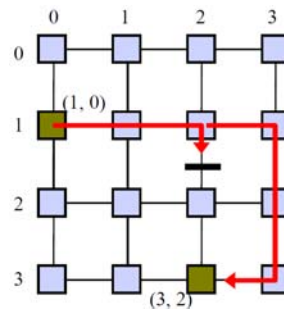
Aunque el encaminamiento dinámico ofrece ciertas ventajas, hay que valorar, como siempre, los posibles inconvenientes que introduce. A menudo, el encaminamiento dinámico es consecuencia de una apuesta; por ejemplo, si en un momento determinado no se puede utilizar una salida porque está ocupada, se decide avanzar por otro camino. Por desgracia, para tomar esa decisión se utiliza sólo información parcial, basada normalmente en el tráfico local, y no se sabe en qué situación se encuentra el nuevo camino elegido (quizás peor que el rechazado). En todo caso, cuando la red no es simétrica o cuando el tráfico se reparte de manera no homogénea, es fácil que surjan zonas de alta densidad de tráfico en la red (*hot spots*), y en esos casos puede resultar efectivo el encaminamiento adaptativo.

Por otra parte, si se divide un mensaje en varios paquetes y éstos utilizan caminos diferentes para llegar al destino, lo más probable es que lleguen en desorden, por lo que habrá que recomponer el mensaje en el destino (y los paquetes deberán incluir más información de control). Además, los encaminadores deberán ser más complejos para poder aplicar encaminamiento adaptativo y, por tanto, el tiempo de procesamiento de los paquetes será mayor. Por último, como analizaremos más adelante, cuando se utiliza encaminamiento adaptativo se pueden producir bloqueos (*deadlock*) en la red: los paquetes no son capaces de avanzar, porque se impiden avanzar unos a otros. Por tanto, hay que evaluar cuidadosamente el uso de encaminamiento adaptativo.

- **Encaminamiento no mínimo**

Aunque el camino recorrido por los paquetes debiera ser de longitud mínima, en algunos casos puede ser necesario o útil utilizar caminos más largos para llegar a destino, para, por ejemplo, rodear una zona de tráfico denso, o, sobre todo, para evitar zonas con fallos en la red.

Por ejemplo, un paquete va desde el procesador (1, 0) al procesador (3, 2), y cuando está en el encaminador del nodo (1, 2) detecta que no puede avanzar porque el encaminador del nodo (2, 2) no responde (se ha estropeado). En se caso, tiene la posibilidad de tomar un camino más largo, como el que aparece en la figura, para poder completar la comunicación.



Salvo las excepciones citadas, y con las precauciones requeridas, los caminos que van a recorrer los paquetes serán siempre de distancia mínima.