

Encaminamiento en redes conmutadas

- 12.1. Encaminamiento en redes de conmutación de circuitos**
- 12.2. Encaminamiento en redes de conmutación de paquetes**
 - Características
 - Estrategias de encaminamiento
 - Ejemplos
- 12.3. Algoritmos de mínimo coste**
 - Algoritmo de Dijkstra
 - Algoritmo de Bellman-Ford
 - Comparación
- 12.4. Lecturas recomendadas**
- 12.5. Términos clave, cuestiones de repaso y ejercicios**
 - Términos clave
 - Cuestiones de repaso
 - Ejercicios



CUESTIONES BÁSICAS

- El encaminamiento en redes de conmutación de circuitos se ha basado tradicionalmente en esquemas estáticos, donde se consideran rutas alternativas para dar respuesta a aumentos de carga. Los esquemas de encaminamiento actuales proporcionan estrategias más adaptables y flexibles.
- La función de encaminamiento de una red de conmutación de paquetes trata de encontrar la ruta de mínimo coste a través de la red, estando el parámetro de coste basado en el número de saltos, el retardo esperado u otras métricas. Los algoritmos de encaminamiento adaptables se fundamentan generalmente en el intercambio de información relativa a las condiciones de tráfico entre los nodos.



Un aspecto clave de diseño de las redes conmutadas, entre las que se encuentran las de conmutación de paquetes, de retransmisión de tramas, las redes ATM y las redes internet, es el relativo al encaminamiento. En términos generales, esta función trata de encontrar rutas a través de la red entre pares de nodos finales comunicantes, de modo que la red se use de forma eficiente.

Este capítulo comienza con un breve estudio de los aspectos involucrados en el encaminamiento en redes de conmutación de circuitos. Seguidamente, se verá el encaminamiento en redes de conmutación de paquetes y se examinarán los algoritmos de mínimo coste, parte fundamental en el encaminamiento en redes de conmutación de paquetes. Estos dos últimos puntos cubren aspectos que resultan básicos para el encaminamiento en redes de tipo internet.

12.1. ENCAMINAMIENTO EN REDES DE CONMUTACIÓN DE CIRCUITOS

En una red grande de conmutación de circuitos como, por ejemplo, la red telefónica de larga distancia de AT&T, muchas de las conexiones de circuitos necesitan una ruta que atravesase más de un conmutador. Cuando se establece una llamada, la red debe encontrar una ruta desde el abonado llamante hasta el abonado llamado que pase a través de varios conmutadores y enlaces. Existen dos requisitos fundamentales para la arquitectura de red que tienen efecto sobre la estrategia de encaminamiento: eficiencia y flexibilidad. En primer lugar, es deseable minimizar la cantidad de equipos (conmutadores y enlaces) en la red, teniendo en cuenta que debemos ser capaces de gestionar toda la carga esperada. Las necesidades de carga se expresan usualmente en términos de tráfico en horas punta, lo cual se refiere, sencillamente, a la carga promedio esperada durante los periodos de más actividad a lo largo del día. Desde un punto de vista práctico, es necesario ser capaz de gestionar esta cantidad de tráfico; desde el punto de vista del coste, sería deseable gestionar esta carga con el menor equipamiento posible. Otro requisito es la flexibilidad. Aunque la red se puede dimensionar teniendo en cuenta el tráfico en horas punta, es posible que la carga supere temporalmente este nivel (por ejemplo, durante una gran tormenta). Puede darse también el caso de que, ocasionalmente, los conmutadores y las líneas fallen y se encuentren momentáneamente inaccesibles (puede que, desgraciadamente, coincidiendo con la propia tormenta). Sería deseable, por tanto, que la red proporcionase un nivel razonable de servicio incluso bajo tales circunstancias.

El aspecto clave de diseño que determina la naturaleza del compromiso entre eficiencia y flexibilidad es la estrategia de encaminamiento. Tradicionalmente, la función de encaminamiento en redes públicas de telecomunicaciones ha sido bastante simple. Esencialmente, los conmutadores de



flexibilidad=>
capacidad Adicional

=>Redundancia y
Capacidad Extra.

Condiciones
Cambiantes?
Que hago?

Solución=>
Encaminamiento
Dinámico.

Que se analiza en
cada instante?
Condiciones
de Tráfico

Una ruta? : NO
Cinjunto de Rutas: SI

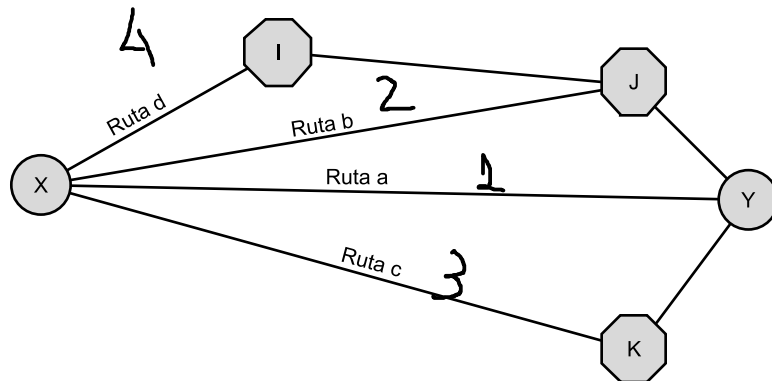
una red se organizaban en una estructura en árbol o jerarquía, estableciéndose una ruta a través del árbol, comenzando en el abonado llamante, hasta el primer nodo común, y después hasta el abonado llamado. Para proporcionar cierta flexibilidad a la red, se incluían en el árbol enlaces de alta capacidad adicionales para conectar entre sí centrales con altos volúmenes de tráfico. En general, esta aproximación es de naturaleza estática. La incorporación de enlaces de alta capacidad proporciona redundancia y capacidad extra, pero persisten las limitaciones en términos de eficiencia y de flexibilidad. Dado que este esquema de encaminamiento no es capaz de adaptarse a condiciones cambiantes, la red debe diseñarse para dar servicio en condiciones típicas de alta carga. Para ofrecer un ejemplo de los problemas a los que da lugar esta aproximación, téngase en cuenta que las horas punta para el tráfico este-oeste no coinciden con las del tráfico norte-sur y plantean, además, diferentes demandas al sistema. Es difícil analizar los efectos de estas variables, que pueden dar lugar a un sobredimensionamiento y, en consecuencia, provocar ineficiencia. En términos de flexibilidad, la estructura jerárquica fija con enlaces adicionales puede responder pobremente ante la ocurrencia de fallos. En general, la consecuencia de un fallo en estos casos es la aparición de una congestión local importante cerca del lugar donde se origina el fallo.

Para hacer frente a la creciente demanda de las redes públicas de telecomunicaciones, la práctica totalidad de los proveedores han pasado de una aproximación jerárquica estática a la adopción de una aproximación dinámica. En un esquema de encaminamiento dinámico las decisiones de encaminamiento están influenciadas en cada instante de tiempo por las condiciones de tráfico actuales. Generalmente, los nodos de conmutación de circuitos mantienen una relación de igual a igual entre sí, en lugar de una jerárquica como la presente en el esquema estático. Todos los nodos están capacitados para realizar las mismas funciones. Esta arquitectura de encaminamiento es más compleja y, a la vez, más flexible. Más compleja porque la arquitectura no proporciona una ruta o conjunto de rutas «natural» basándose en la estructura jerárquica, pero al mismo tiempo es más flexible debido a que hay más rutas alternativas.

Como ejemplo, veamos una forma de encaminamiento en redes de conmutación de circuitos llamada encaminamiento alternativo. La base de los esquemas de encaminamiento alternativo reside en que las posibles rutas entre dos centrales finales se encuentran predefinidas, siendo responsabilidad del conmutador origen seleccionar el camino adecuado para cada llamada. Cada conmutador dispone de un conjunto de rutas prefijadas, en orden de preferencia, para cada destino; si existe una conexión directa entre dos conmutadores, ésta suele ser la elección preferida; si no está disponible esta línea se prueba con la segunda alternativa, y así sucesivamente. Las secuencias de encaminamiento (conjunto de rutas intentadas) reflejan un análisis basado en patrones de tráfico conocidos y se diseñan para optimizar la utilización de los recursos de la red.

Si sólo se define una secuencia de encaminamiento para cada pareja origen-destino, el esquema se conoce como esquema de encaminamiento alternativo fijo. No obstante, es más frecuente el uso de un esquema de encaminamiento alternativo dinámico, en el cual se utiliza un conjunto diferente de rutas preplanificadas en instantes distintos de tiempo, con objeto de aprovechar las distintas condiciones de tráfico en las diferentes franjas horarias y en los distintos periodos de un día. En consecuencia, la decisión de encaminamiento se basa tanto en el estado del tráfico actual (una ruta se descartará si está ocupada) como en patrones de tráfico conocidos (que determinan la secuencia de rutas a considerar).

En la Figura 12.1 se muestra un ejemplo sencillo. El conmutador origen, X, tiene cuatro posibles rutas hacia el conmutador destino, Y. Siempre se intentará en primer lugar la ruta directa *a*; si este enlace no está disponible (ocupado o fuera de servicio), se intentarán las otras rutas en un orden dado dependiendo de la hora de que se trate. Por ejemplo, durante las mañanas del fin de semana la siguiente ruta en probarse será la *b*.



Ruta a: X → Y
 Ruta b: X → J → Y
 Ruta c: X → K → Y
 Ruta d: X → I → J → Y

○ = Central final
 ⬡ = Nodo de conmutación intermedio

(a) Topología

| Periodo de tiempo | Primera ruta | Segunda ruta | Tercera ruta | Cuarta y última ruta |
|-------------------|--------------|--------------|--------------|----------------------|
| Mañana | a | b | c | d |
| Tarde | a | d | b | c |
| Noche | a | d | c | b |
| Fin de semana | a | c | b | d |

(b) Tabla de encaminamiento

Figura 12.1. Rutas alternativas desde la central final X hasta la central final Y.

12.2. ENCAMINAMIENTO EN REDES DE CONMUTACIÓN DE PAQUETES

Uno de los aspectos más complejos y cruciales del diseño de redes de conmutación de paquetes es el relativo al encaminamiento. Este apartado comienza con una revisión de las principales características que se pueden usar para clasificar las estrategias de encaminamiento. Tras esto se discutirán algunos esquemas concretos.

Los principios descritos en esta sección son también aplicables al encaminamiento en la interconexión de redes, discutida en la Parte V del texto.

CARACTERÍSTICAS

La función principal de una red de conmutación de paquetes es aceptar paquetes procedentes de una estación emisora y enviarlos hacia una estación destino. Para ello se debe determinar una ruta a través de la red, siendo posible generalmente la existencia de más de una. Así pues, se debe realizar una función de encaminamiento, entre cuyos requisitos se encuentran los siguientes:

- Exactitud
- Simplicidad
- Robustez
- Estabilidad
- Imparcialidad
- Optimización
- Eficiencia

Las dos primeras características mencionadas se explican por sí mismas. La **robustez** está relacionada con la habilidad de la red para enviar paquetes de alguna forma ante la aparición de sobrecargas y fallos localizados. Idealmente, la red puede reaccionar ante estas contingencias sin sufrir pérdidas de paquetes o caída de circuitos virtuales. **No obstante, la robustez puede implicar cierta inestabilidad.** Las técnicas que reaccionan ante condiciones cambiantes presentan una tendencia no deseable a reaccionar demasiado lentamente ante determinados eventos o a experimentar oscilaciones inestables de una situación extrema a otra. Por ejemplo, la red puede reaccionar ante la aparición de congestión en un área desplazando la mayor parte de la carga hacia una segunda zona. Ahora será la segunda región la que estará sobrecargada y la primera infrautilizada, produciéndose un segundo desplazamiento del tráfico. Durante estos desplazamientos puede ocurrir que los paquetes viajen en bucles a través de la red.

También existe un compromiso entre la **característica de imparcialidad** y el hecho de que el **encaminamiento trate de ser óptimo.** Algunos criterios de funcionamiento pueden dar prioridad al intercambio de paquetes entre estaciones vecinas frente al intercambio realizado entre estaciones distantes, lo cual puede maximizar la eficiencia promedio pero será injusto para aquella estación que necesite comunicar principalmente con estaciones lejanas.

Finalmente, una técnica de **encaminamiento implica cierto coste de procesamiento** en cada nodo y, en ocasiones, también un coste en la transmisión, impidiéndose en ambos casos el funcionamiento eficiente de la red. Este coste debe ser inferior a los beneficios obtenidos por el uso de una métrica razonable, como la mejora de la robustez o la imparcialidad.

Con estos requisitos en mente estamos en condiciones de evaluar los distintos elementos de diseño involucrados en un esquema de encaminamiento. En la Tabla 12.1 se listan estos elementos. Algunos de ellos se solapan o dependen de otros, pero un estudio acerca de los mismos clarificará y permitirá organizar los conceptos de encaminamiento.

Tabla 12.1. Elementos de diseño en las técnicas de encaminamiento en redes de conmutación de paquetes.

| | |
|---|---|
| <p>Criterios de rendimiento</p> <ul style="list-style-type: none"> Número de saltos Coste Retardo Eficiencia <p>Instante de decisión</p> <ul style="list-style-type: none"> Paquete (datagrama) Sesión (circuitos virtuales) <p>Lugar de decisión</p> <ul style="list-style-type: none"> Cada nodo (distribuido) Nodo central (centralizado) Nodo origen (fuente) | <p>Fuente de información de la red</p> <ul style="list-style-type: none"> Ninguna Local Nodo adyacente Nodos a lo largo de la ruta Todos los nodos <p>Tiempo de actualización de la información de la red</p> <ul style="list-style-type: none"> Continuo Periódico Cambio importante en la carga Cambio en la topología |
|---|---|

Criterios de rendimiento

La elección de una ruta se fundamenta generalmente en algún criterio de rendimiento. El más simple **consiste en elegir el camino con menor número de saltos** (aquel que atraviesa el menor número

Los nros. indicados en cada enlace son los costos en ese sentido.

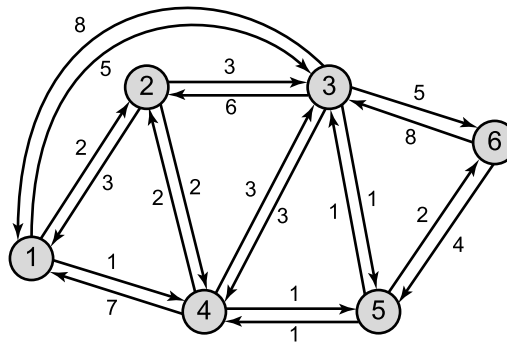


Figura 12.2. Ejemplo de red de conmutación de paquetes.

de nodos) a través de la red¹. Éste es un criterio que se puede medir fácilmente y que debería minimizar el consumo de recursos de la red. Una generalización del criterio de menor número de saltos lo constituye el encaminamiento de mínimo coste. En este caso se asocia un coste a cada enlace y, para cualesquiera dos estaciones conectadas, se elige aquella ruta a través de la red que implique el coste total mínimo. Por ejemplo, en la Figura 12.2 se muestra una red en la que las dos líneas con flecha entre cada par de nodos representan un enlace entre ellos, y los números asociados indican el coste actual del enlace en cada sentido. El camino más corto (menor número de saltos) desde el nodo 1 hasta el 6 es 1-3-6 (coste = 5 + 5 = 10), pero el de mínimo coste es 1-4-5-6 (coste = 1 + 1 + 2 = 4). La asignación de los costes a los enlaces se hace en función de los objetivos de diseño; por ejemplo, el coste podría estar inversamente relacionado con la velocidad (es decir, a mayor velocidad menor coste) o con el retardo actual de la cola asociada al enlace. En el primer caso, la ruta de mínimo coste maximizaría la eficiencia, mientras que en el segundo minimizaría el retardo.

Tanto en la técnica de menor número de saltos como en la de mínimo coste, el algoritmo para determinar la ruta o camino óptimo entre dos estaciones es relativamente sencillo, siendo el tiempo de procesamiento aproximadamente el mismo en ambos casos. Dada su mayor flexibilidad, el criterio de mínimo coste es más utilizado que el de menor número de saltos.

Existen varios algoritmos de mínimo coste de uso común, los cuales se describen en la Sección 12.3.

Instante y lugar de decisión

Las decisiones de encaminamiento se realizan de acuerdo con algún criterio de rendimiento. Dos cuestiones importantes en la toma de esta decisión son el instante temporal y el lugar en que se toma la decisión.

El instante de decisión viene determinado por el hecho de que la decisión de encaminamiento se hace en base a un paquete o a un circuito virtual. Cuando la operación interna de la red se basa en datagramas, la decisión de encaminamiento se toma de forma individual para cada paquete. En

¹ El término *salto* se usa con cierta libertad en la bibliografía. La definición más común, usada en este texto, es que el número de saltos a lo largo de una ruta entre un origen y un destino dados es el número de nodos de la red (nodos de conmutación de paquetes, conmutadores ATM, dispositivos de encaminamiento, etc.) que encuentra un paquete a lo largo de dicha ruta. El número de saltos es igual a veces al número de enlaces, o terminales de grafo, atravesados. A partir de esta última definición se obtiene un valor superior en uno al conseguido mediante la definición aceptada en nuestro caso.

el caso de **circuítos virtuales internos**, la decisión sólo se realiza en el momento en que se establece un **circuito virtual dado**, de modo que, en el caso más sencillo, todos los paquetes siguientes que usan ese circuito virtual siguen la misma ruta. En redes más complejas, la red puede cambiar dinámicamente la ruta asignada a un circuito virtual particular en respuesta a condiciones cambiantes (por ejemplo, sobrecarga o fallos en una parte de la red).

El término **lugar de decisión** hace referencia al **nodo o nodos en la red responsables de la decisión de encaminamiento**. El más común es el **encaminamiento distribuido**, en el que cada nodo de la red tiene la responsabilidad de seleccionar un enlace de salida sobre el que llevar a cabo el envío de los paquetes a medida que éstos se reciben. En el encaminamiento centralizado, la decisión se toma por parte de algún nodo designado al respecto, como puede ser un centro de control de red. El peligro de esta última aproximación es que el fallo del centro de control puede bloquear el funcionamiento de la red; así pues, aunque la aproximación distribuida puede resultar más compleja es también más robusta. Una tercera alternativa empleada en algunas redes es la conocida **como encaminamiento desde el origen**. En este caso, es la estación origen y no los nodos de la red quien realmente toma la decisión de encaminamiento, comunicándosela a la red. **Esto permite al usuario fijar una ruta a través de la red de acuerdo con criterios locales al mismo**.

El instante y el lugar de decisión son variables de diseño independientes. Por ejemplo, supongamos que el lugar de decisión en la Figura 12.2 es cada nodo y que los valores especificados son los costes en un instante de tiempo dado, los cuales pueden cambiar. Un paquete desde el nodo 1 al 6 podría seguir la ruta 1-4-5-6, estando cada enlace de la ruta determinado localmente por el nodo transmisor. Supongamos ahora que los valores cambian de forma que 1-4-5-6 ya no es el camino óptimo. En una red de datagramas, el paquete siguiente puede seguir una ruta diferente, de nuevo determinada por cada nodo a lo largo del camino. En una red de circuitos virtuales, cada nodo recuerda la decisión de encaminamiento tomada cuando se estableció el circuito virtual, de modo que se limita a transmitir los paquetes sin tomar decisiones nuevas.

Necesito esto casi siempre..

Fuente de información de la red y tiempo de actualización

La mayor parte de los esquemas de encaminamiento requieren que las decisiones se tomen en base **al conocimiento de la topología de la red, la carga y el coste de los enlaces**. Sorprendentemente, algunas estrategias, como la **de inundaciones y el encaminamiento aleatorio** (descritas más adelante), no hacen uso de ninguna información para la transmisión de los paquetes.

En el **encaminamiento distribuido**, en el que la decisión de encaminamiento se toma en **cada uno de los nodos**, éstos hacen uso de **información local**, como es el coste asociado a los distintos enlaces de salida; también pueden utilizar información de **los nodos adyacentes** (directamente conectados), como la congestión experimentada en ellos. Finalmente, existen algoritmos de uso común que permiten al nodo obtener información de todos los nodos de una potencial ruta de interés. En el caso del **encaminamiento centralizado**, el **nodo central hace uso generalmente de información procedente de todos los nodos**.

Un concepto relacionado es **el de tiempo de actualización de la información**, el cual es función de la fuente de información y de la estrategia de encaminamiento. Es evidente que si no se usa información (como en el método de inundaciones) no existe actualización. Si sólo se utiliza información local, la actualización es esencialmente continua, ya que un nodo individual conoce siempre sus condiciones locales actuales. Para el resto de categorías de fuentes de información (nodos adyacentes, todos los nodos), **el tiempo de actualización depende de la estrategia de encaminamiento**. **Para una estrategia de encaminamiento estático, la información no se actualiza nunca,**



mientras que para una técnica adaptable la actualización se lleva a cabo periódicamente a fin de posibilitar la adaptación de la decisión de encaminamiento a las condiciones cambiantes de la red.

Como cabe esperar, cuanto mayor sea la información disponible y más frecuentemente se actualice ésta, más probable será que las decisiones de encaminamiento tomadas por la red sean buenas. Eso sí, teniendo presente que la transmisión de esta información consume recursos de la red.

ESTRATEGIAS DE ENCAMINAMIENTO

Existen numerosas estrategias de encaminamiento para abordar las necesidades de encaminamiento en redes de conmutación de paquetes. Muchas de ellas son aplicables también al encaminamiento en la interconexión de redes, estudiada en la Parte V del texto. En este apartado se presentan cuatro estrategias principales: estática, inundaciones, aleatoria y adaptable.

Encaminamiento estático

En el encaminamiento estático se configura una única ruta permanente para cada par de nodos origen-destino en la red, pudiéndose utilizar para ello cualquiera de los algoritmos de encaminamiento de mínimo coste descritos en la Sección 12.3. Las rutas son fijas (al menos mientras lo sea la topología de la red), de modo que los costes de enlace usados para el diseño de las rutas no pueden estar basados en variables dinámicas como el tráfico, aunque sí podrían estarlo en tráfico esperado o en capacidad.

La Figura 12.3 sugiere cómo se pueden implementar rutas estáticas. Se crea una matriz de encaminamiento central, almacenada, por ejemplo, en un centro de control de red. Esta matriz específica, para cada par de nodos origen-destino, la identidad del siguiente nodo en la ruta.

Obsérvese que no es necesario almacenar la ruta completa para cada par de nodos; es suficiente conocer, para cada pareja, cuál es el primer nodo en la ruta. Para comprender mejor este hecho, supongamos que la ruta de mínimo coste desde X hasta Y comienza con el enlace $X-A$. Llamemos R_1 al resto de la ruta, es decir, desde A hasta Y , y definamos R_2 como la ruta de mínimo coste de A a Y . Si el coste de R_1 es mayor que el de R_2 , la ruta $X-Y$ mejorará al usar R_2 en lugar de R_1 . Si el coste de R_1 es menor que el de R_2 , entonces esta última ruta no es la de mínimo coste desde A hasta Y ; por tanto, $R_1 = R_2$. Así pues, en cada punto a lo largo del camino sólo es necesario conocer la identidad del nodo siguiente, no la ruta completa. En nuestro ejemplo, la ruta desde el nodo 1 al nodo 6 atraviesa en primer lugar el nodo 4. Consultando de nuevo la matriz, se observa que la ruta del nodo 4 al 6 atraviesa el nodo 5. Por último, la ruta desde el nodo 5 hasta el 6 es un enlace directo entre ambos. Por tanto, la ruta completa desde el nodo 1 al 6 es 1-4-5-6.

A partir de esta matriz se pueden crear y almacenar en cada nodo las tablas de encaminamiento asociadas. Siguiendo el razonamiento del párrafo anterior, cada nodo sólo necesitará almacenar una columna de la tabla de encaminamiento, indicándose en ella el nodo siguiente para cada destino.

En el encaminamiento estático no existe diferencia entre el uso de datagramas y de circuitos virtuales, ya que todos los paquetes procedentes de un origen dado y con un destino concreto siguen la misma ruta. La ventaja del encaminamiento estático es su simplicidad, además de su buen funcionamiento en redes fiables con carga estacionaria. Su desventaja, en cambio, radica en la falta de flexibilidad, ya que no reacciona ante fallos ni ante congestión en la red.

MATRIZ DE ENCAMINAMIENTO CENTRAL

| | | Nodo origen | | | | | |
|--------------|---|-------------|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 |
| Nodo destino | 1 | — | 1 | 5 | 2 | 4 | 5 |
| | 2 | 2 | — | 5 | 2 | 4 | 5 |
| | 3 | 4 | 3 | — | 5 | 3 | 5 |
| | 4 | 4 | 4 | 5 | — | 4 | 5 |
| | 5 | 4 | 4 | 5 | 5 | — | 5 |
| | 6 | 4 | 4 | 5 | 5 | 6 | — |

Tabla del nodo 1

| Destino | Nodo siguiente |
|---------|----------------|
| 2 | 2 |
| 3 | 4 |
| 4 | 4 |
| 5 | 4 |
| 6 | 4 |

Tabla del nodo 2

| Destino | Nodo siguiente |
|---------|----------------|
| 1 | 1 |
| 3 | 3 |
| 4 | 4 |
| 5 | 4 |
| 6 | 4 |

Tabla del nodo 3

| Destino | Nodo siguiente |
|---------|----------------|
| 1 | 5 |
| 2 | 5 |
| 4 | 5 |
| 5 | 5 |
| 6 | 5 |

Tabla del nodo 4

| Destino | Nodo siguiente |
|---------|----------------|
| 1 | 2 |
| 2 | 2 |
| 3 | 5 |
| 5 | 5 |
| 6 | 5 |

Tabla del nodo 5

| Destino | Nodo siguiente |
|---------|----------------|
| 1 | 4 |
| 2 | 4 |
| 3 | 3 |
| 4 | 4 |
| 6 | 6 |

Tabla del nodo 6

| Destino | Nodo siguiente |
|---------|----------------|
| 1 | 5 |
| 2 | 5 |
| 3 | 5 |
| 4 | 5 |
| 5 | 5 |

Figura 12.3. Encaminamiento estático (haciendo uso de la Figura 12.2).

Una mejora al encaminamiento estático, que soportaría la no disponibilidad temporal de nodos y enlaces, consiste en la especificación de nodos siguientes alternativos para cada dirección. Por ejemplo, los nodos alternativos en la tabla del nodo 1 serían 4, 3, 2, 3, 3.

Inundaciones

Otra técnica de encaminamiento sencilla es la de inundaciones, la cual no precisa de ninguna información sobre la red y funciona como sigue. Un nodo origen envía un paquete a todos sus nodos vecinos, los cuales, a su vez, lo transmiten sobre todos los enlaces de salida excepto por el que llegó. Por ejemplo, si el nodo 1 de la Figura 12.2 tiene que enviar un paquete al nodo 6, transmite una copia (con la dirección de destino de 6) a los nodos 2, 3 y 4. El nodo 2 enviará una copia a los nodos 3 y 4; el nodo 4 enviará, a su vez, una copia a los nodos 2, 3 y 5; y así sucesivamente. Dado que, eventualmente, el nodo 6 recibirá varias copias del paquete, éste debe contener un identificador único (por ejemplo, nodo origen y número de secuencia o número de circuito virtual y número de secuencia) para que el nodo destino pueda quedarse con una sola copia y descartar el resto.

A menos que se haga algo para cesar las continuas retransmisiones de paquetes, el número de éstos en circulación para un mismo paquete origen crece sin límite. Una forma de prevenir estas

con el contador y el identificador de cada paquete puedo saber cantidad de salto para cada paquete ya que están identificados

para que no existan paquetes duplicados...(contador)

retransmisiones consiste en que cada nodo recuerde la identidad de los paquetes que ha retransmitido con anterioridad, de manera que se rechazan copias duplicadas. Una técnica más sencilla consiste en incluir un campo de cuenta de saltos en cada paquete. Este contador puede ponerse inicialmente a un valor máximo, como es por ejemplo el diámetro de la red (longitud de la ruta de menor número de saltos más larga a través de la red)². Cada vez que un nodo transmite un paquete decrementa la cuenta en uno, de modo que cuando el contador alcanza el valor cero se elimina el paquete de la red.

Un ejemplo de esta última técnica se muestra en la Figura 12.4. Supongamos que se envía un paquete desde el nodo 1 al nodo 6 y se le asigna una cuenta de saltos igual a 3. En el primer salto se crean tres copias del paquete; en el segundo salto de estas copias se crea un total de nueve copias. Una de estas copias alcanza el nodo 6, quien, al detectar que el destino es él, no la retransmite. Sin embargo, los otros nodos generan un total de 22 nuevas copias en el tercer y último salto. Obsérvese que si un nodo no guarda el identificador del paquete puede generar múltiples copias

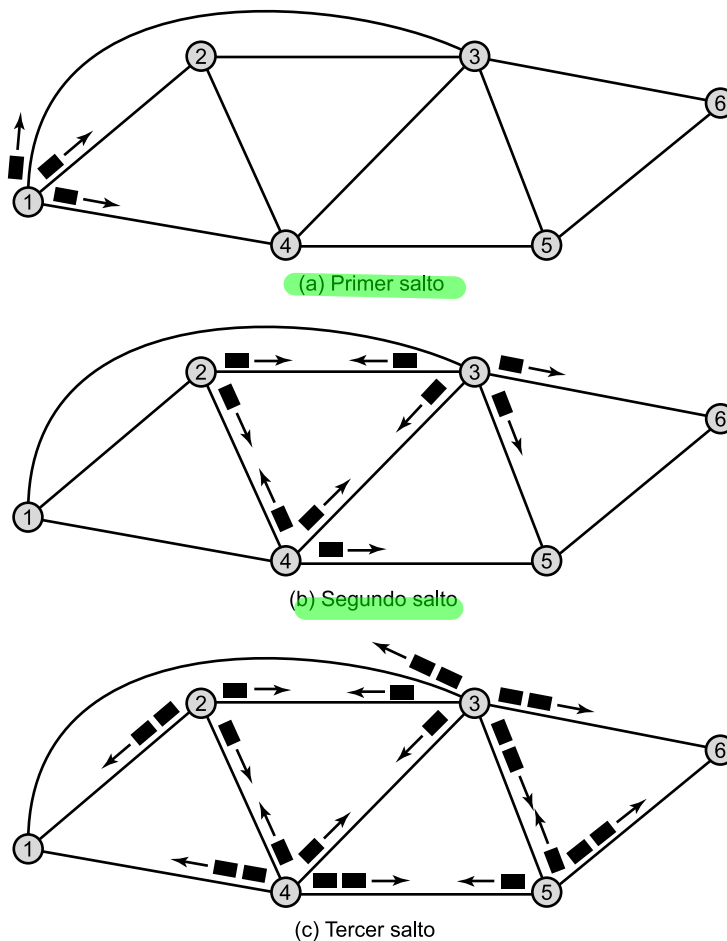


Figura 12.4. Ejemplo de inundaciones (número de saltos = 3).

² Para cada pareja de sistemas finales conectados a la red existe una ruta de menor número de saltos, denominándose diámetro de la red a la longitud de la ruta más larga de entre todas las de menor número de saltos.

en este tercer paso. Todos los paquetes recibidos tras el tercer salto son eliminados, habiéndose recibido en el nodo 6 un total de cuatro copias adicionales del paquete.

La técnica de inundaciones presenta tres propiedades importantes:

- Se prueban todos los posibles caminos entre los nodos origen y destino. De este modo, independientemente de lo que pueda sucederle a un nodo o a un enlace, se garantiza la recepción del paquete siempre que exista, al menos, una ruta entre el origen y el destino.
- Dado que se prueban todos los caminos, al menos una copia del paquete a recibir en el destino habrá usado una ruta de menor número de saltos.
- Se visitan todos los nodos que están directa o indirectamente conectados al nodo origen.

Por la primera propiedad, la técnica de inundaciones resulta muy robusta y puede ser usada para enviar mensajes de alta prioridad. Un ejemplo de aplicación es una red militar que puede sufrir daños importantes. Por la segunda propiedad, la técnica de inundaciones podría emplearse inicialmente para establecer la ruta para un circuito virtual. La tercera propiedad sugiere que la técnica de inundaciones puede resultar útil para llevar a cabo la propagación de información relevante para todos los nodos; ya se verá que se utiliza en algunos esquemas para la propagación de información de encaminamiento.

La principal desventaja de la técnica de inundaciones es la gran cantidad de tráfico que genera, directamente proporcional a la conectividad de la red.

Encaminamiento aleatorio

La técnica de encaminamiento aleatorio presenta, con menor tráfico, la sencillez y robustez de la técnica de inundaciones. En este esquema, un nodo selecciona un único camino de salida para retransmitir un paquete entrante; el enlace de salida se elige de forma aleatoria, excluyendo el enlace por el que llegó el paquete. Si todos los enlaces son igualmente probables de ser elegidos, una implementación sencilla consistiría en seleccionarlos de forma alternada.

Una mejora a esta técnica consiste en asignar una probabilidad a cada uno de los enlaces de salida y llevar a cabo la selección de acuerdo con estas probabilidades. La probabilidad se puede basar en la velocidad de datos, en cuyo caso se tiene

$$P_i = \frac{R_i}{\sum_j R_j}$$

velocidad del enlace i
Velocidad total de todos los enlaces candidatos.

donde

P_i = probabilidad de seleccionar el enlace i .

R_i = velocidad del enlace i .

La suma se realiza para todos los enlaces de salida candidatos. Este esquema proporciona una distribución del tráfico adecuada. Obsérvese que las probabilidades también podrían estar basadas en costes de enlace fijos.

Como en el caso de la técnica de inundaciones, el encaminamiento aleatorio no necesita el uso de información sobre la red. Dado que la ruta se elige de forma aleatoria, ésta no corresponderá en general con la de mínimo coste ni con la de menor número de saltos. Por tanto, la red debe transportar un tráfico superior al óptimo, aunque inferior al de la técnica de inundaciones.

Encaminamiento adaptable

+ usada!!

Prácticamente en todas las redes de conmutación de paquetes se utiliza algún tipo de técnica de encaminamiento adaptable; es decir, las decisiones de encaminamiento cambian en la medida que lo hacen las condiciones de la red. Las principales condiciones que influyen en las decisiones de encaminamiento son:

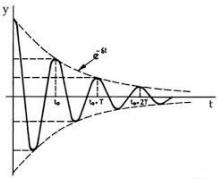
- **Fallos:** cuando un nodo o una línea troncal fallan, no pueden volver a ser usados como parte de una ruta.
- **Congestión:** cuando una parte de la red sufre una congestión importante, es deseable encaminar los paquetes de forma que se rodee la zona congestionada, en lugar de realizar el encaminamiento a través de ella.



Para hacer posible el encaminamiento adaptable es necesario que los nodos intercambien información acerca del estado de la red. El uso de la técnica de encaminamiento adaptable presenta varias desventajas en comparación con el encaminamiento estático:

- La decisión de encaminamiento es más compleja, por lo que aumenta el coste de procesamiento en los nodos de la red. mas procesa => mas \$
- En la mayor parte de los casos, las estrategias adaptables dependen de la información de estado obtenida en una parte de la red pero que es utilizada en otra. Existe un compromiso entre la calidad de la información y la cantidad de datos suplementarios o redundancia utilizada. Cuanta más información se intercambia y más frecuentemente se hace, mejores serán las decisiones de encaminamiento tomadas en cada nodo. Por otro lado, esta información constituye en sí misma tráfico adicional sobre la red, lo que supone cierta degradación de las prestaciones de ésta.

hay que tener cuidado!



- Una estrategia adaptable puede reaccionar demasiado rápidamente, provocando oscilaciones y causando congestión, o demasiado lentamente, en cuyo caso no es válida.



A pesar de estos peligros reales, las estrategias de encaminamiento adaptable son, con mucho, las más utilizadas por dos razones:

- El usuario de la red percibe que las prestaciones mejoran con el uso de estas técnicas.
- Como se discutirá en el Capítulo 13, una estrategia de encaminamiento adaptable puede resultar de ayuda en el control de la congestión: dado que este tipo de técnica tiende a compensar la carga, puede retrasar la aparición de situaciones graves de congestión.

Dependiendo de la validez del diseño y de la naturaleza del tráfico, estas ventajas se pueden constatar o no debido a la complejidad para lograr un funcionamiento correcto. Como demostración de este hecho, la mayor parte de las redes de conmutación de paquetes, como ARPANET y sus sucesoras y varias redes comerciales, han sufrido al menos una revisión en sus técnicas de encaminamiento.

Una clasificación adecuada de las estrategias de encaminamiento adaptable es la realizada de acuerdo con la fuente de la información: local, nodos adyacentes o todos los nodos. Un ejemplo de técnica adaptable basada sólo en información local es aquella en la que cada nodo encamina cada paquete recibido por la línea de salida cuya cola asociada Q sea menor, lo que haría que se compensase la carga entre las distintas líneas de salida. Sin embargo, puede que algunos enlaces de salida no lleven al destino adecuado, por lo que se podría mejorar la técnica, como en el caso del encaminamiento aleatorio, teniendo en consideración la dirección deseada. En este caso, cada enlace de salida tendría un peso B_i para cada destino i . Para cada paquete recibido con destino el

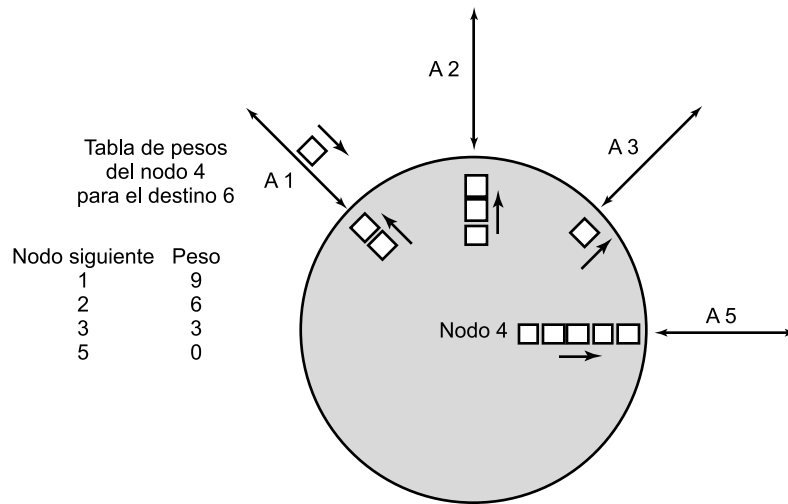


Figura 12.5. Ejemplo de encaminamiento adaptable aislado.

nodo i , el nodo intermedio elegirá aquella línea que minimice $Q + B_i$, de manera que los paquetes se envíen en la dirección correcta considerando los retardos provocados por el tráfico.

Como ejemplo, en la Figura 12.5 se muestra el estado del nodo 4 de la Figura 12.2 en un instante de tiempo dado. Este nodo tiene sendos enlaces a otros cuatro nodos. Al recibirse varios paquetes se produce un exceso, de forma que se crea una cola de paquetes para cada una de las líneas de salida. ¿Hacia qué línea se debería encaminar un paquete recibido desde el nodo 1 con destino al 6? De acuerdo con las longitudes de las colas y la tabla de pesos (B_6) para cada enlace de salida, el valor mínimo de $Q + B_6$ es 4, correspondiente al enlace hacia el nodo 3. Por tanto, el nodo 4 encaminará el paquete hacia dicho nodo.

Los esquemas adaptables basados sólo en información local son raramente utilizados, puesto que no explotan con facilidad la información disponible. Las estrategias basadas en el uso de la información procedente de los nodos adyacentes o de todos los nodos se utilizan más debido a la mejor información acerca de los retardos en los nodos de que se dispone en estos casos. Estas técnicas adaptables pueden ser distribuidas o centralizadas. En el primer caso, cada nodo intercambia información de retardo con otros nodos, de modo que cada nodo trata de estimar el retardo a través de la red a partir de la información recibida y en base a un algoritmo de encaminamiento de mínimo coste. En el caso de una técnica centralizada, cada nodo informa sobre su estado de retardo a un nodo central, quien diseña las rutas de acuerdo con esta información recibida y devuelve la información de encaminamiento a los nodos.

EJEMPLOS

En este apartado se estudiarán varios ejemplos de estrategias de encaminamiento. Todas ellas fueron desarrolladas para ARPANET, que es una red de conmutación de paquetes predecesora de la actual Internet. Resulta instructivo examinar estas estrategias por varias razones. Primero, porque éstas y otras técnicas similares se usan también en otras redes de conmutación de paquetes, incluyendo varias de las que componen Internet. En segundo lugar, hay que decir que los esquemas de encaminamiento basados en el trabajo de ARPANET se han usado también en la interconexión

de redes en Internet y en redes privadas. Por último, porque el esquema de encaminamiento de ARPANET evolucionó de una manera que aclara algunos de los aspectos clave en el diseño de los algoritmos de encaminamiento.

Primera generación

El algoritmo de encaminamiento original, diseñado en 1969, era un algoritmo adaptable distribuido que hacía uso de la estimación de los retardos como criterio de rendimiento y de una versión del algoritmo de Bellman-Ford (véase Sección 12.3). Para este algoritmo, cada nodo mantiene dos vectores:

$$D_i = \begin{bmatrix} d_{i1} \\ \vdots \\ d_{iN} \end{bmatrix} \quad S_i = \begin{bmatrix} s_{i1} \\ \vdots \\ s_{iN} \end{bmatrix}$$

donde

D_i = vector de retardo para el nodo i .

d_{ij} = estimación actual del retardo mínimo desde el nodo i al nodo j ($d_{ii} = 0$).

N = número de nodos en la red.

S_i = vector de nodos sucesores para el nodo i .

s_{ij} = nodo siguiente en la ruta actual de mínimo retardo de i a j .

Periódicamente (cada 128 ms), cada nodo intercambia su vector de retardo con todos sus vecinos. A partir de todos los vectores de retardo recibidos, un nodo k actualiza sus dos vectores como sigue:

$$d_{kj} = \min_{i \in A} [d_{ij} + l_{ki}]$$

$s_{kj} = i$, siendo i el que minimiza la expresión anterior

donde

A = conjunto de nodos vecinos de k .

l_{ki} = estimación actual del retardo desde el nodo k al nodo i .

En la Figura 12.6 se muestra un ejemplo del algoritmo original de ARPANET, usando la red de la Figura 12.7. Ésta es la misma red que la de la Figura 12.2 pero con diferentes costes asociados a los enlaces (y suponiendo el mismo coste en ambos sentidos del enlace). En la Figura 12.6a se muestra la tabla de encaminamiento del nodo 1 en un instante de tiempo que refleja los costes asociados a los enlaces de la Figura 12.7. Para cada destino se especifica un retardo y el nodo siguiente en la ruta que lo produce. En algún momento, los costes de los enlaces cambian a los valores indicados en la Figura 12.2. Supuesto que los vecinos del nodo 1 (nodos 2, 3 y 4) conociesen el cambio antes que él, cada uno de estos nodos actualizará su vector de retardo y enviará una copia a todos sus vecinos, incluyendo el nodo 1 (véase Figura 12.6b). El nodo 1 desecha su tabla de encaminamiento y construye una nueva basándose en los vectores de retardo recibidos y en la propia estimación que él hace del retardo para cada uno de los enlaces de salida a sus vecinos. El resultado obtenido se muestra en la Figura 12.6c.

El retardo de enlace estimado no es más que el tamaño o longitud de la cola para el enlace en cuestión. Así, con la construcción de una nueva tabla de encaminamiento el nodo tiende a favore-

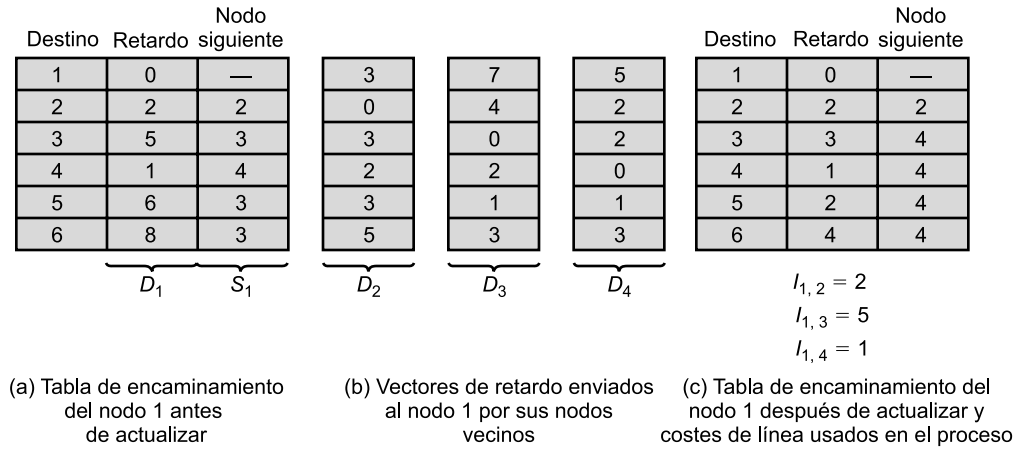


Figura 12.6. Algoritmo de encaminamiento original de ARPANET.

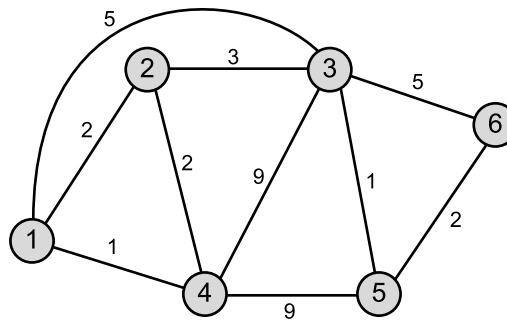


Figura 12.7. Red para el ejemplo de la Figura 12.6a.

cer aquellos enlaces con menores colas, lo que compensa la carga entre las distintas líneas de salida. Sin embargo, dado que el tamaño de las colas varía rápidamente a lo largo del tiempo, la percepción distribuida de la ruta más corta podría cambiar mientras un paquete se encuentra en tránsito. Esto podría provocar una situación en la que un paquete se encamina hacia un área de baja congestión en lugar de hacia el destino.

Segunda generación

Tras años de experiencia y algunas modificaciones sin importancia, el algoritmo de encaminamiento original se reemplazó en 1979 por otro bastante diferente [MCQU80]. Los principales inconvenientes del antiguo algoritmo eran los siguientes:

- No se consideraba la velocidad de las líneas sino sólo su tamaño de cola, por lo que a las líneas de alta capacidad no se les daba el tratamiento de favor que merecían.
- El tamaño de las colas es, en cualquier caso, una medida artificial del retardo, ya que se consume un cierto tiempo de procesamiento desde que el paquete se recibe en un nodo hasta que es puesto en cola.

Veo una foto del pasado!!

- El algoritmo no era demasiado preciso; de hecho, su respuesta era muy lenta ante aumentos en la congestión y en el retardo.

El nuevo algoritmo es también adaptable distribuido en el que se hace uso del retardo como criterio de rendimiento, pero las diferencias son significativas. En lugar de usar la longitud de la cola como indicador del retardo, éste se mide directamente como sigue. A cada paquete recibido en un nodo se le coloca un sello o marca de tiempo indicando el instante temporal en que llegó. También se almacena el instante en que se transmite. Si se recibe una confirmación positiva, el retardo se calcula como el tiempo de salida menos el de llegada más el tiempo de transmisión y el de propagación. Para ello, el nodo debe conocer la velocidad del enlace y el tiempo de propagación. En cambio, si se recibe una confirmación negativa, se actualiza el tiempo de salida y el nodo vuelve a intentarlo hasta que se consigue con éxito una medida del retardo de transmisión.

El nodo calcula el retardo medio de cada enlace de salida cada 10 segundos. Si se producen cambios significativos en el valor obtenido, se envía la información a los demás nodos mediante el algoritmo de inundaciones. Cada nodo mantiene una estimación del retardo de cada enlace de la red, de modo que cuando recibe nueva información se actualiza la tabla de encaminamiento haciendo uso del algoritmo de Dijkstra (véase Sección 12.3).

Tercera generación

El segundo mejor pero....

La experiencia con este nuevo esquema demostró que era más adecuado y estable que el anterior. El coste derivado del empleo de la técnica de inundaciones era moderado, ya que cada nodo la llevaba a cabo cada 10 segundos; sin embargo, se observó un problema en el funcionamiento de esta nueva estrategia a medida que aumentaba el tráfico en la red, por lo que fue revisada en 1987 [KHAN89].

El problema de la segunda estrategia consistía en la suposición de que el retardo de paquetes estimado para un enlace es un buen indicador del retardo de enlace, una vez que todos los nodos realizan el encaminamiento de su tráfico basándose en dicho retardo. Este mecanismo de encaminamiento resulta efectivo sólo si existe alguna correlación entre los valores estimados y los realmente experimentados una vez realizado el encaminamiento. Esta correlación tiende a ser mayor cuando el tráfico es bajo o moderado, pero cuando existe alta carga la correlación es pequeña. Por tanto, inmediatamente después de que todos los nodos hayan actualizado las tablas, éstas quedan obsoletas.

2da Gen..solo calcula bien para Tráfico moderado!!

Como ejemplo, considérese una red consistente en dos regiones con sólo dos enlaces, A y B, que las conectan (véase Figura 12.8). Cualquier ruta entre dos nodos situados en regiones diferentes debe atravesar uno de estos enlaces. Supóngase una situación tal que la mayor parte del tráfico lo soporta la línea A. Esto implicará que el retardo en dicha línea es elevado, comunicándose este hecho al resto de los nodos en el siguiente instante de tiempo. Esta actualización se recibirá en todos los nodos aproximadamente al mismo tiempo, actualizándose inmediatamente sus tablas de encaminamiento. Es probable que este nuevo retardo para el enlace A sea lo suficientemente elevado para hacer que el enlace B sea ahora el elegido por la mayoría de las rutas, si no todas, entre ambas regiones. Dado que todos los nodos actualizan sus tablas al mismo tiempo, la mayor parte del tráfico entre las dos regiones se desplaza simultáneamente hacia la línea B. Esto provocará que sea ahora esta línea la que presente un retardo elevado, por lo que el tráfico se desplazará de nuevo hacia la línea A. Esta oscilación persistirá mientras lo haga el volumen de tráfico.

Existen varias razones por las que dicha oscilación resulta indeseable:

- Una parte importante de la capacidad disponible no se utiliza precisamente cuando más se necesita: en condiciones de alta carga.



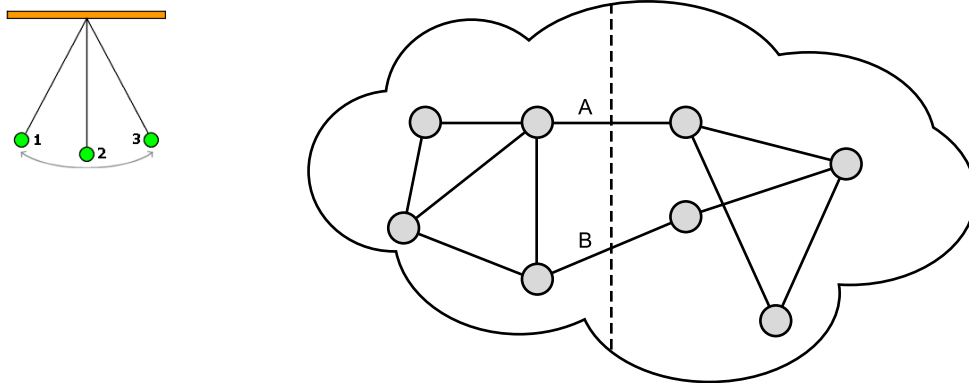


Figura 12.8. Red de conmutación de paquetes sujeta a oscilaciones.

- La utilización excesiva de algunos enlaces puede provocar congestión en la red (esto se verá cuando se estudie la congestión en el Capítulo 13).
- Las oscilaciones en los valores de retardo obtenidos hacen necesaria una actualización más frecuente de las tablas de encaminamiento. Este hecho incrementa el tráfico en la red justo cuando ésta ya presenta alta carga.

Los diseñadores de ARPANET concluyeron que el problema radicaba en el hecho de que todos los nodos estaban tratando de obtener la ruta óptima para todos los destinos, lo que provocaba conflictos. Se concluyó que, para alta carga, el objetivo del encaminamiento debería consistir en la obtención de una ruta promedio, en lugar de intentar la determinación de todos los caminos mejores.

Los diseñadores decidieron que era innecesario cambiar todo el algoritmo; el cambio de la función que determinaba el coste de los enlaces bastaba para evitar las oscilaciones en el encaminamiento y reducir su coste. El cálculo comienza midiendo el retardo medio en los últimos 10 segundos. Este valor se transforma como se indica a continuación:

1. Haciendo uso de un sencillo modelo de colas con un único servidor, el retardo medido se transforma en una estimación de la utilización de la línea. Por teoría de colas, la utilización se puede expresar en función del retardo como sigue:

$$\rho = \frac{2(T_s - T)}{T_s - 2T} \quad \text{Ruta promedio S!!!!}$$

donde

ρ = utilización del enlace.

T = retardo medido.

T_s = tiempo de servicio.

El tiempo de servicio se hace igual al tamaño medio de los paquetes en la red (600 bits) dividido entre la velocidad de la línea.

2. El resultado se suaviza promediándolo con la utilización estimada previamente:

$$U(n + 1) = 0,5 \times \rho(n + 1) + 0,5 \times U(n)$$

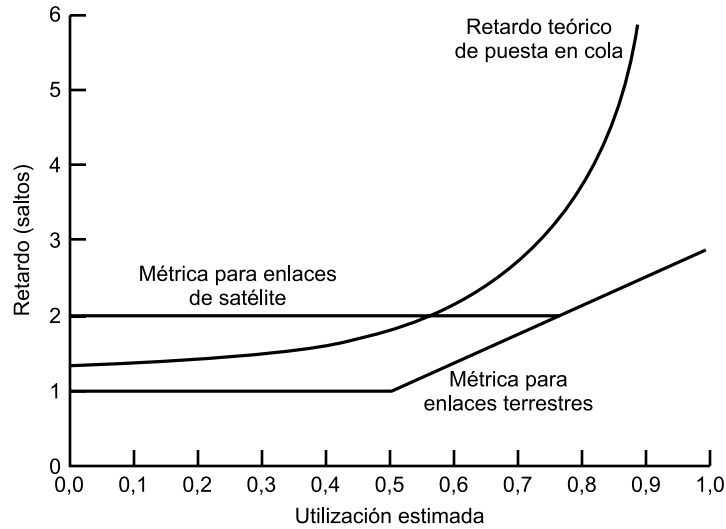


Figura 12.9. Métricas de retardo en ARPANET.

donde

$U(n)$ = utilización media calculada en el instante de muestreo n .

$\rho(n)$ = utilización del enlace en el instante de tiempo n .

El valor promedio incrementa el periodo de las oscilaciones en el encaminamiento, lo que reduce el coste adicional de este último.

3. El coste del enlace se establece como una función de la utilización media, pensada para proporcionar una estimación razonable del coste sin provocar oscilación. En la Figura 12.9 se indica la forma de convertir la estimación de la utilización en un valor de coste, valor que se obtiene a partir del retardo.

En la Figura 12.9 mencionada se normaliza el retardo al valor alcanzado en una línea desocupada, el cual corresponde al tiempo de propagación más el tiempo de transmisión. Cada curva en la figura indica la forma en que el retardo real depende de la utilización; el incremento en el retardo se debe al retardo de cola en el nodo. Para el nuevo algoritmo, el coste se mantiene al valor mínimo hasta que se alcanza un nivel de utilización dado, lo que tiene el efecto de reducir el coste del encaminamiento cuando el tráfico es reducido. Por encima de un cierto nivel de utilización, se permite que el coste alcance un valor máximo igual a tres veces el valor mínimo. El efecto de este valor máximo es establecer que el tráfico no debe ser encaminado alrededor de una línea con alta carga más que en dos saltos adicionales.

Obsérvese que el umbral mínimo es superior para enlaces satélite, lo que potencia el uso de los enlaces terrestres para condiciones de baja carga, dado que éstos presentan un retardo de propagación inferior. Nótese también que la curva de retardo real es mucho más pronunciada que las curvas de transformación para altos niveles de utilización. Esta pendiente en el coste del enlace provoca que el tráfico en un enlace se distribuya, lo que causa la aparición de oscilaciones en el encaminamiento.

En resumen, la función de coste estudiada está más orientada a la utilización que al retardo. La función actúa de forma similar a una métrica basada en retardo cuando la carga es baja y a una métrica basada en la capacidad en condiciones de alta carga.

12.3. ALGORITMOS DE MÍNIMO COSTE

Prácticamente todas las redes de conmutación de paquetes y todas las redes de tipo internet basan sus decisiones de encaminamiento en algún criterio de mínimo coste. Si el criterio consiste en minimizar el número de saltos, cada enlace tendrá asociado un valor igual a 1. Normalmente, el valor asociado al enlace es inversamente proporcional a su capacidad, proporcional a su carga actual o alguna combinación de ellos. En cualquier caso, el coste de las líneas se emplea como entrada a un algoritmo de encaminamiento de mínimo coste, que establece que:

Dada una red de nodos conectados entre sí por enlaces bidireccionales, donde cada enlace tiene un coste asociado en cada sentido, se define el coste de una ruta entre dos nodos como la suma de los costes de los enlaces atravesados. Así, para cada par de nodos se obtiene el camino de mínimo coste.

Obsérvese que el coste de un enlace puede ser diferente para cada uno de los dos sentidos. Esto sería cierto, por ejemplo, si el coste de un enlace fuese igual a la longitud de la cola de paquetes esperando ser transmitidos sobre el enlace por cada uno de los dos nodos.

La mayor parte de los algoritmos de encaminamiento de mínimo coste utilizados en las redes de conmutación de paquetes y en las redes internet son variantes de uno de los dos algoritmos más comunes: el de Dijkstra y el de Bellman-Ford. Este apartado presenta una breve descripción de ambos.

ALGORITMO DE DIJKSTRA

El algoritmo de Dijkstra [DIJK59] se puede enunciar como sigue: encontrar las rutas más cortas entre un nodo origen dado y todos los demás nodos, desarrollando los caminos en orden creciente de longitud. El algoritmo actúa en etapas. Tras el paso o etapa k -ésima se han determinado los caminos más cortos a los k nodos más cercanos (de menor coste) al nodo origen especificado; estos nodos se almacenan en el conjunto T . En el paso $(k + 1)$ se añade a la lista T aquel nodo que presente el camino más corto desde el nodo origen y que no se encuentre ya incluido en dicha lista. A medida que se incorporan nuevos nodos a T , se define su camino desde el origen. El algoritmo se puede describir formalmente como sigue. Definamos:

N = conjunto de nodos de la red.

s = nodo origen.

T = lista o conjunto de nodos añadidos o incorporados por el algoritmo.

$w(i, j)$ = coste del enlace desde el nodo i al nodo j ; $w(i, i) = 0$; $w(i, j) = \infty$ si los dos nodos no se encuentran directamente conectados; $w(i, j) \geq 0$ si los dos nodos están directamente conectados.

$L(n)$ = coste en curso obtenido por el algoritmo para el camino de mínimo coste del nodo s al nodo n ; al finalizar el algoritmo, este coste corresponde al del camino de mínimo coste de s a n en el grafo.

El algoritmo consta de tres pasos, repitiéndose los pasos 2 y 3 hasta que $T = N$; es decir, hasta que las rutas finales han sido asignadas a todos los nodos en la red:

1. [Inicialización]

$T = \{s\}$ El conjunto de nodos incorporados sólo consta del nodo origen s .

$L(n) = w(s, n)$, para $n \neq s$ El coste inicial de las rutas a los nodos vecinos es el asociado a los enlaces.

- 2. [Obtención del siguiente nodo]** Se busca el nodo vecino que no esté en T con el camino de menor coste desde s y se incorpora a T ; también se incorporará el enlace desde ese nodo hasta un nodo de T que forma parte del camino. Esto se puede expresar como

$$\text{Encontrar } x \notin T \text{ tal que } L(x) = \min_{j \notin T} L(j)$$

Añadir x a T , incorporando también el enlace desde x que contribuye a $L(x)$ como la componente de menor coste (es decir, el último salto en la ruta).

3. [Actualización de los caminos de mínimo coste]

$$L(n) = \min [L(n), L(x) + w(x, n)] \quad \text{para todo } n \notin T$$

Si el último término es el mínimo, el camino desde s hasta n es ahora el camino desde s hasta x concatenado con el enlace desde x hasta n .

El algoritmo concluye cuando todos los nodos han sido añadidos a T . Al final, el valor $L(x)$ asociado a cada nodo x es el coste (longitud) de la ruta de mínimo coste de s a x . Además, T define la ruta de mínimo coste desde s hasta cualquier otro nodo.

Cada iteración de los pasos 2 y 3 incorpora un nuevo nodo a T y define el camino de mínimo coste desde s hasta ese nodo, atravesando dicha ruta sólo nodos incluidos en T . Para comprender mejor esto considérese el siguiente razonamiento. Tras k iteraciones existen k nodos en T , habiéndose obtenido además el camino de mínimo coste desde s hasta cada uno de esos nodos. Consideremos ahora todos los caminos posibles desde s hasta los nodos no incluidos en T . Entre estos caminos existe uno de mínimo coste que pasa exclusivamente a través de nodos en T (véase Ejercicio 12.4), terminando con un enlace directo entre algún nodo en T y un nodo no incluido en esta lista. Este nodo se añade a T y se define el camino asociado como la ruta de mínimo coste para ese nodo.

En la Tabla 12.2a y en la Figura 12.10 se muestra el resultado de aplicar el algoritmo al grafo de la Figura 12.2 con $s = 1$. Los enlaces sombreados definen el árbol de expansión correspondiente al grafo, mientras que los valores que aparecen rodeados por un círculo corresponden a la estimación actual de $L(x)$ para cada nodo x . Los nodos sombreados representan la incorporación de éstos a T . Obsérvese que en cada etapa se obtiene el camino a cada nodo, así como el coste total asociado al mismo. Tras la última iteración se dispone del camino de mínimo coste a cada nodo y del coste asociado. El mismo procedimiento se puede utilizar considerando como nodo origen el 2, y así sucesivamente.

ALGORITMO DE BELLMAN-FORD

El algoritmo de Bellman-Ford [FORD62] se puede enunciar así: encontrar los caminos más cortos desde un nodo origen dado con la condición de que éstos contengan a lo sumo un enlace; a continuación encontrar los caminos más cortos con la condición de que contengan dos enlaces como

Tabla 12.2. Ejemplo de algoritmos de encaminamiento de mínimo coste (haciendo uso de la Figura 12.2).

a) Algoritmo de Dijkstra ($s = 1$)

| Iteración T | $L(2)$ | Ruta | $L(3)$ | Ruta | $L(4)$ | Ruta | $L(5)$ | Ruta | $L(6)$ | Ruta | |
|---------------|--------------------|------|--------|------|---------|------|--------|----------|--------|----------|---------|
| 1 | {1} | 2 | 1-2 | 5 | 1-3 | 1 | 1-4 | ∞ | — | ∞ | — |
| 2 | {1, 4} | 2 | 1-2 | 4 | 1-4-3 | 1 | 1-4 | 2 | 1-4-5 | ∞ | — |
| 3 | {1, 2, 4} | 2 | 1-2 | 4 | 1-4-3 | 1 | 1-4 | 2 | 1-4-5 | ∞ | — |
| 4 | {1, 2, 4, 5} | 2 | 1-2 | 3 | 1-4-5-3 | 1 | 1-4 | 2 | 1-4-5 | 4 | 1-4-5-6 |
| 5 | {1, 2, 3, 4, 5} | 2 | 1-2 | 3 | 1-4-5-3 | 1 | 1-4 | 2 | 1-4-5 | 4 | 1-4-5-6 |
| 6 | {1, 2, 3, 4, 5, 6} | 2 | 1-2 | 3 | 1-4-5-3 | 1 | 1-4 | 2 | 1-4-5 | 4 | 1-4-5-6 |

b) Algoritmo de Bellman-Ford ($s = 1$)

| h | $L_h(2)$ | Ruta | $L_h(3)$ | Ruta | $L_h(4)$ | Ruta | $L_h(5)$ | Ruta | $L_h(6)$ | Ruta |
|-----|----------|------|----------|---------|----------|------|----------|-------|----------|---------|
| 0 | ∞ | — | ∞ | — | ∞ | — | ∞ | — | ∞ | — |
| 1 | 2 | 1-2 | 5 | 1-3 | 1 | 1-4 | ∞ | — | ∞ | — |
| 2 | 2 | 1-2 | 4 | 1-4-3 | 1 | 1-4 | 2 | 1-4-5 | 10 | 1-3-6 |
| 3 | 2 | 1-2 | 3 | 1-4-5-3 | 1 | 1-4 | 2 | 1-4-5 | 4 | 1-4-5-6 |
| 4 | 2 | 1-2 | 3 | 1-4-5-3 | 1 | 1-4 | 2 | 1-4-5 | 4 | 1-4-5-6 |

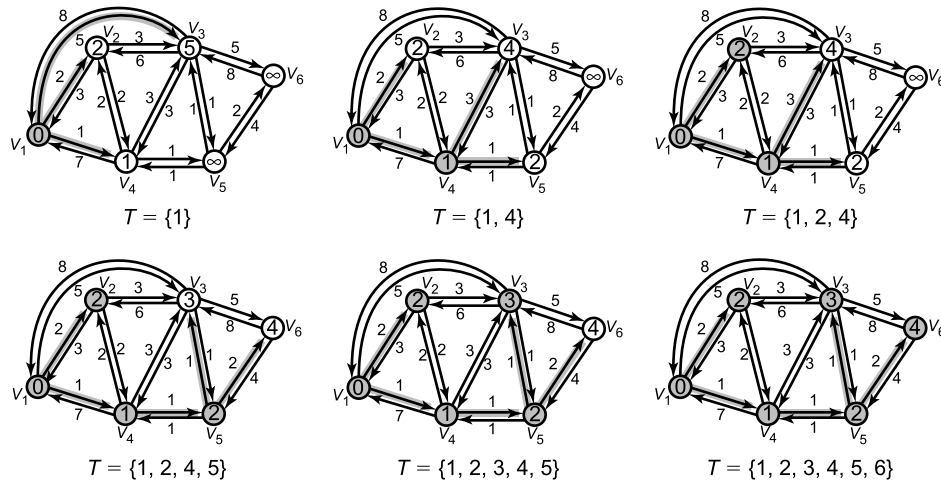


Figura 12.10. Algoritmo de Dijkstra aplicado al grafo de la Figura 12.2.

máximo, y así sucesivamente. Este algoritmo actúa también en pasos, pudiéndose describir formalmente como sigue. Definamos:

s = nodo origen.

$w(i, j)$ = coste del enlace desde el nodo i al nodo j ; $w(i, i) = 0$; $w(i, j) = \infty$ si los dos nodos no se encuentran directamente conectados; $w(i, j) \geq 0$ si los dos nodos están directamente conectados.

h = número máximo de enlaces en un camino en el paso actual del algoritmo.

$L_h(n)$ = coste del camino de mínimo coste desde el nodo s hasta el nodo n con la condición de que no haya más de h enlaces.

1. [Inicialización]

$$L_0(n) = \infty, \forall n \neq s$$

$$L_h(s) = 0, \forall h$$

2. [Actualización]

Para cada sucesivo $h \geq 0$:

Para cada $n \neq s$, calcular

$$L_{h+1}(n) = \min_j [L_h(j) + w(j, n)]$$

Conectar n con el nodo predecesor j de mínimo coste y eliminar todas las conexiones de n con un nodo predecesor diferente obtenido en una iteración anterior. El camino de s a n finaliza con el enlace de j a n .

Para la iteración del paso 2 con $h = K$, y para cada nodo de destino n , el algoritmo compara las rutas potenciales de longitud $K + 1$ desde s hasta n con el camino existente al final de la iteración anterior. Si el camino más corto previo tiene un coste inferior, se guarda; en caso contrario, se define un nuevo camino de longitud $K + 1$ de s a n , el cual consiste en una ruta de longitud K de s a algún nodo j más un salto directo desde el nodo j hasta el nodo n . En este caso, el camino de s a j considerado corresponde a la ruta de K saltos para j definida en la iteración anterior (véase Ejercicio 12.5).

En la Tabla 12.2b y en la Figura 12.11 se muestra el resultado de aplicar este algoritmo a la Figura 12.2 usando $s = 1$. En cada etapa se determinan las rutas de mínimo coste con un número máximo de enlaces igual a h . Tras la última iteración se conoce el camino de mínimo coste a cada nodo y el coste asociado. El mismo procedimiento se puede usar tomando como nodo origen el nodo 2, y así sucesivamente. Obsérvese que los resultados coinciden con los obtenidos por el algoritmo de Dijkstra.

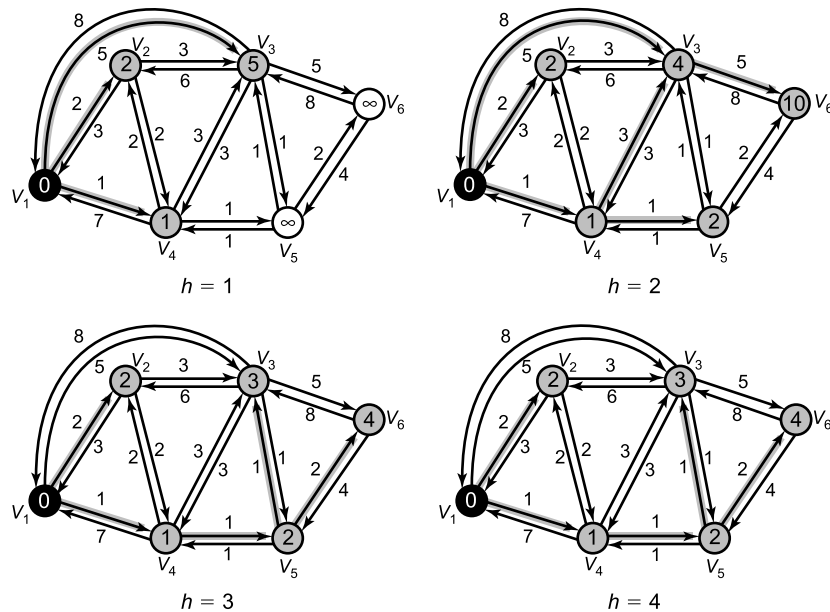


Figura 12.11. Algoritmo de Bellman-Ford aplicado al grafo de la Figura 12.2.

COMPARACIÓN

Una comparación interesante entre estos dos algoritmos hace referencia a la información necesaria a considerar. Veamos en primer lugar el algoritmo de Bellman-Ford. En el paso 2 del proceso, el cálculo para el nodo n requiere conocer el coste de los enlaces a todos los nodos vecinos de n —es decir, $w(j, n)$ — además del coste total del camino a cada uno de estos nodos desde un nodo origen particular s —es decir, $L_h(j)$ —. Cada nodo puede mantener un conjunto de costes y rutas asociadas para cada uno de los otros nodos en la red e intercambiar periódicamente esta información con sus vecinos directos. Por tanto, cada nodo puede hacer uso de la expresión dada en el paso 2 del algoritmo de Bellman-Ford, basándose sólo en la información dada por sus vecinos y en el conocimiento del coste de las líneas asociadas, para actualizar los caminos y sus costes. Consideremos ahora el algoritmo de Dijkstra. En el paso 3 parece necesitarse que cada nodo disponga de la información completa acerca de la topología de la red; es decir, cada nodo debe conocer todos los enlaces y los costes asociados a ellos. Así, en este algoritmo, la información se debe intercambiar con todos los demás nodos.

En general, en la evaluación de las ventajas relativas de ambos algoritmos se debe considerar el tiempo de procesamiento de los algoritmos y la cantidad de información a obtener del resto de nodos de la red o de la internet, dependiendo dicha evaluación de la implementación específica.

Por último, se ha de reseñar que ambos algoritmos convergen, y lo hacen hacia la misma solución bajo condiciones estáticas de la topología y del coste de los enlaces. Si el coste de los enlaces varía a lo largo del tiempo, el algoritmo tratará de reflejar estos cambios; sin embargo, si el coste de los enlaces depende del tráfico, que a su vez depende de las rutas elegidas, se produce una realimentación que puede provocar una situación de inestabilidad.

12.4. LECTURAS RECOMENDADAS

En [GIRA90] se ofrece un estudio adecuado acerca del encaminamiento en redes de conmutación de circuitos. [CORM01] contiene un análisis detallado de los algoritmos de mínimo coste presentados en el capítulo, discutiéndose también en detalle estos algoritmos en [BERT92].

BERT92 Bertsekas, D. y Gallager, R. *Data Networks*. Upper Saddle River, NJ: Prentice Hall, 1992.

CORM01 Cormen, T., et al., *Introduction to Algorithms*. Cambridge, MA: MIT Press, 2001.

GIRA90 Girard, A. *Routing and Dimensioning in Circuit-switching Networks*. Reading, MA: Addison-Wesley, 1990.

12.5. TÉRMINOS CLAVE, CUESTIONES DE REPASO Y EJERCICIOS

TÉRMINOS CLAVE

algoritmo de Bellman-Ford
algoritmo de Dijkstra
algoritmos de mínimo coste
encaminamiento adaptable

encaminamiento aleatorio
encaminamiento alternativo
encaminamiento estático
inundaciones


```

        pred[n]:=1
    end;
L[s]:=0; final[s]:=true; {el nodo s se etiqueta permanentemente con 0}
recent:=s; {el nodo más reciente para ser etiquetado
            permanentemente es s}

path:=true;
{inicialización}

while final[t]=false do
begin
    for n:=1 to N do {encontrar nueva etiqueta}
        if (w[recent,n] < ∞) AND (NOT final[n]) then
            {para cada sucesor inmediato de recent que no se
             encuentra permanentemente etiquetado, hacer}
            begin {actualizar las etiquetas temporales}
                newlabel:=L[recent]+w[recent,n];
                if newlabel < L[n] then
                    begin L[n]:=newlabel; pred[n]:=recent end
                    {se etiqueta de nuevo n si existe un camino
                     más corto a través del nodo recent y se hace
                     recent el predecesor de n en el camino más corto desde s}
                end;
            temp:=∞;
            for x:=1 to N do {encontrar el nodo con la etiqueta temporal menor}
                if (NOT final[x]) AND (L[x] < temp) then
                    begin y:=x; temp:=L[x] end;
            if temp < ∞ then {existe una ruta} then
                begin final[y]:=true; recent:=y end
                {y, el siguiente nodo más cercano a s, se
                 etiqueta permanentemente}
            else begin path:=false; final[t]:=true end
            end
    end
end

```

En este programa se le asigna temporalmente una etiqueta inicial a cada nodo. Cuando se obtiene una ruta final a un nodo, se le asigna una etiqueta permanente igual al coste del camino desde s . Escriba un programa similar para el algoritmo de Bellman-Ford. *Sugerencia:* el algoritmo de Bellman-Ford se conoce a veces como método de corrección de etiquetas, frente al método de fijado de etiquetas seguido en el algoritmo de Dijkstra.

- 12.4. En la descripción del algoritmo de Dijkstra dada en la Sección 12.3 se dice que en cada iteración se añade un nuevo nodo a T y que la ruta de mínimo coste para ese nuevo nodo sólo atraviesa nodos ya incluidos en T . Demuestre que esto es cierto. *Sugerencia:* comience por el principio. Muestre que el primer nodo añadido a T debe tener un enlace directo al nodo origen, el segundo nodo en T debe tener un enlace directo con el nodo origen o con el primer nodo incluido en T , y así sucesivamente. Recuerde que los costes de todas las líneas se suponen no negativos.
- 12.5. En la descripción del algoritmo de Bellman-Ford se establece que en la iteración para la que $h = K$, si hay definida alguna ruta de longitud $K + 1$, los primeros K saltos de este camino forman una ruta definida en la iteración anterior. Demuestre que es cierto.

- 12.6. Los valores del camino de mínimo coste en el paso 3 del algoritmo de Dijkstra sólo se actualizan para nodos no incluidos aún en T . ¿No es posible encontrar una ruta de mínimo coste para un nodo en T ? Si es así, demuéstrelo con un ejemplo. En caso contrario, justifique razonadamente el motivo.
- 12.7. Haciendo uso del algoritmo de Dijkstra, genere un camino de mínimo coste para los nodos del 2 al 6 con el resto de nodos de la Figura 12.2. Muestre los resultados como en la Tabla 12.2a.
- 12.8. Repita el Ejercicio 12.7 haciendo uso del algoritmo de Bellman-Ford.
- 12.9. Aplique el algoritmo de encaminamiento de Dijkstra para las redes de la Figura 12.12. Obtenga una tabla similar a la Tabla 12.2a y una figura análoga a la Figura 12.10.

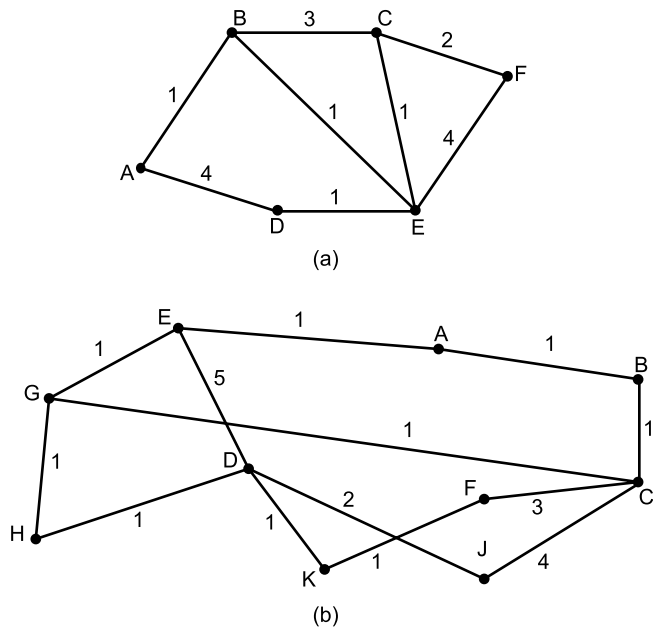


Figura 12.12. Redes de la conmutación de paquetes con costes de enlace asociados.

- 12.10. Repita el Ejercicio 12.9 haciendo uso del algoritmo de Bellman-Ford.
- 12.11. ¿Proporcionan los algoritmos de Dijkstra y de Bellman-Ford los mismos resultados siempre? ¿Por qué sí o por qué no?
- 12.12. Tanto el algoritmo de Dijkstra como el de Bellman-Ford obtienen las rutas de mínimo coste desde un nodo al resto. Por su parte, el algoritmo de Floyd-Warshall obtiene los caminos de mínimo coste entre todos los pares de nodos posibles. Se define:

N = conjunto de nodos en la red.

$w(i, j)$ = coste del enlace del nodo i al nodo j ; $w(i, i) = 0$ y $w(i, j) = \infty$ si los dos nodos no se encuentran directamente conectados.

$L_n(i, j)$ = coste del camino de mínimo coste desde el nodo i al nodo j con la condición de que sólo los nodos $1, 2, \dots, n$ se pueden usar como nodos intermedios en las rutas.

El algoritmo sigue los siguientes pasos:

1. Inicialización:

$$L_0(i, j) = w(i, j), \text{ para todo } i, j, i \neq j$$

2. Para $n = 0, 1, \dots, N - 1$, $L_{n+1}(i, j) = \min[L_n(i, j), L_n(i, n + 1) + L_n(n + 1, j)]$, $\forall i \neq j$ explique el algoritmo con palabras. Demuestre por inducción que funciona correctamente.
- 12.13.** El nodo 1 de la Figura 12.4 envía un paquete al nodo 6 usando inundaciones. Contabilizando la transmisión de un paquete sobre una línea como una carga de uno, indique cuál será el tráfico total generado si:
- a) Cada nodo descarta los paquetes entrantes duplicados.
 - b) Se usa un campo de cuenta de saltos con un valor inicial igual a 5 y no se descartan los paquetes duplicados.
- 12.14.** Ya se vio que el algoritmo de inundaciones se puede utilizar para determinar la ruta con menor número de saltos. ¿Se puede usar también para la obtención del camino con menor retardo?
- 12.15.** El algoritmo de encaminamiento aleatorio sólo permite la existencia de una copia de un paquete en un instante de tiempo dado. A pesar de ello, resulta deseable la utilización de un campo de cuenta de saltos. ¿Por qué?
- 12.16.** Otro esquema de encaminamiento adaptable es el conocido como aprendizaje hacia atrás (*backward learning*). Todo paquete encaminado a través de la red contiene no sólo la dirección de destino, sino también la dirección de origen más un contador de saltos que se incrementa en cada salto. Cada nodo construye una tabla de encaminamiento que especifica el nodo siguiente y el número de saltos para cada destino. ¿Cómo se usa la información contenida en el paquete para construir la tabla? ¿Cuáles son las ventajas y desventajas de esta técnica?
- 12.17.** Construya una tabla de encaminamiento centralizado para las redes de la Figura 12.12.
- 12.18.** Considérese un sistema que emplea la técnica de inundaciones con un contador de saltos que se supone inicialmente igual al «diámetro» de la red. Cuando el contador alcanza el valor cero el paquete se descarta, excepto en el destino. ¿Se asegura siempre así que el paquete alcanzará el destino si existe al menos un camino operativo? Justifique la respuesta.