Encaminamiento en redes con infraestructura

Departamento de Sistemas Telemáticos y Computación (GSyC)

gsyc-profes (arroba) gsyc.es

Noviembre de 2013

Introducción

El nivel de red se ocupa de que los paquetes que salen del transmisor lleguen a su destino final, aunque el emisor y el receptor no estén "adyacentes" (conectados directamente al mismo medio de transmisión).
Esto normalmente requiere pasar a través de nodos intermedios: encaminadores (*routers*). Es necesario elegir la mejor ruta a seguir.
RECORDATORIO: El nivel de enlace sólo se ocupa de que las tramas viajen entre máquinas "adyacentes".

Misiones del Nivel de Red

- Encaminamiento de paquetes.
- Asignación de direcciones únicas a todas las máquinas de la red, independientes de la tecnología de los niveles de enlace.
- Interconexión en una misma red de subredes con distinto nivel de enlace.
- Control de congestión.

Tipos de Nivel de Red

Según haya o no conexiones de red:

- No orientado a conexión
- Orientado a conexión

Según se encamine cada paquete por separado o no:

- Basado en datagramas
- Basado en circuitos virtuales

Según se ofrezca o no un servicio fiable:

- Fiable
- No fiable

Servicio No Orientado a Conexión

- Cada vez que el nivel superior quiere enviar datos, se compone una unidad de datos (paquete) con ellos y se envía. No hay relación con transmisiones previas o futuras al mismo destino

Servicio Orientado a Conexión

- Antes de enviar el primer byte de datos, origen y destino mantienen un diálogo inicial para establecer ciertas condiciones de la transferencia de información, que se mantienen mientras dure esta transferencia

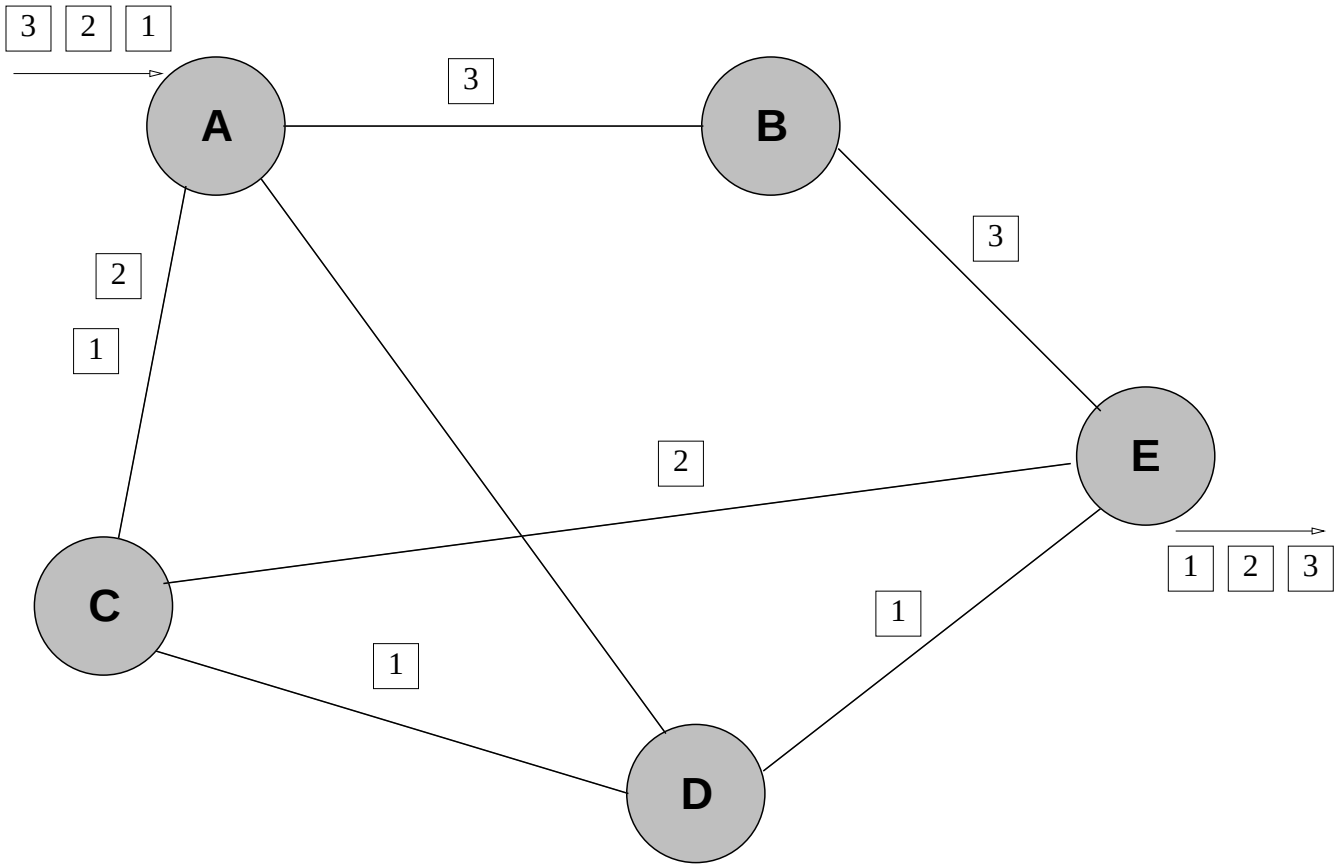
Servicio basado en Datagramas

- La dirección de destino viaja en todos los paquetes de datos.
- El encaminamiento de cada paquete es independiente, por lo que varios paquetes enviados del mismo origen al mismo destino pueden viajar por diferentes rutas (y, tal vez, llegar en desorden).

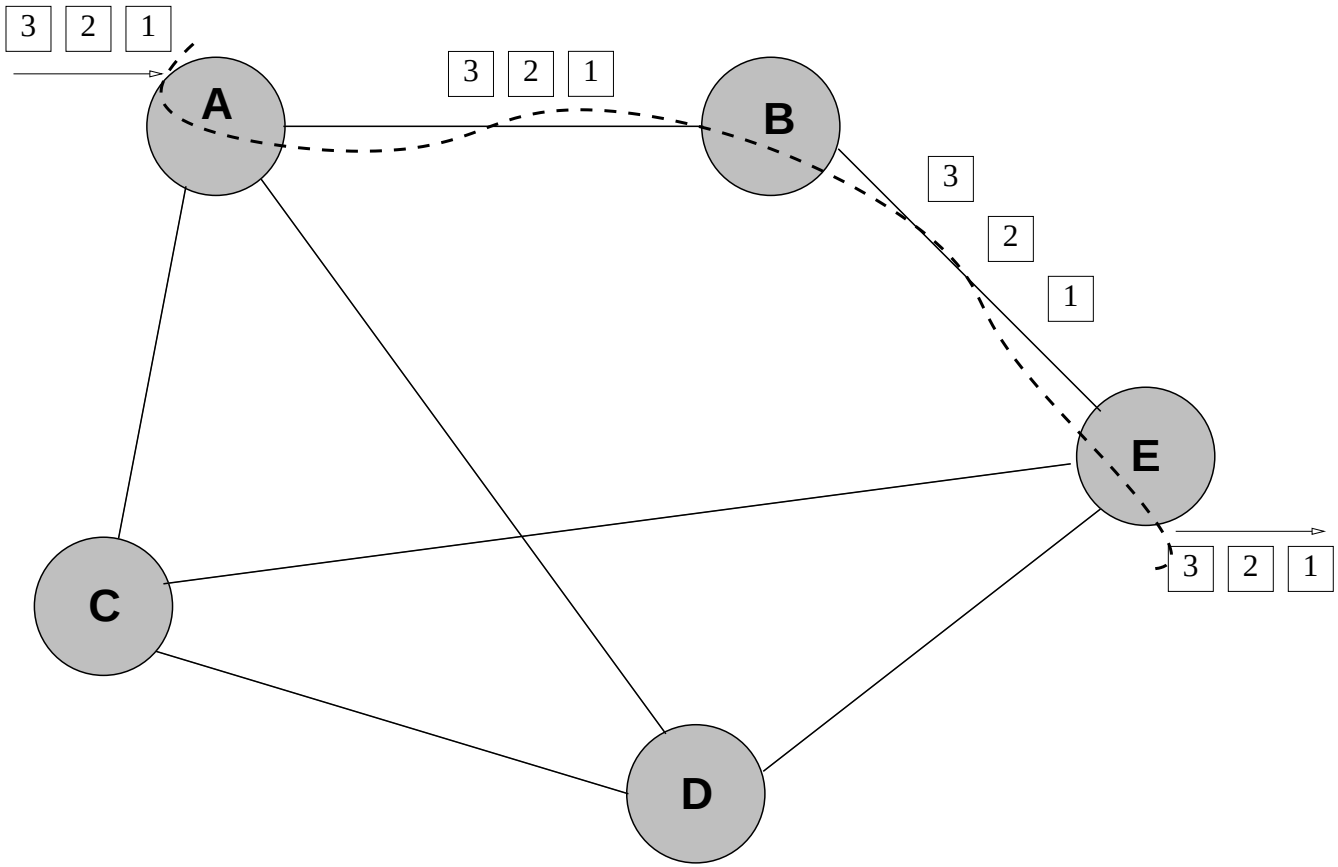
Servicio basado en Circuitos Virtuales

- Al principio se establece un "circuito virtual" por el que viajarán todos los paquetes de datos.
- La dirección de destino viaja sólo en los paquetes que establecen el cirtuito virtual. Los paquetes con datos sólo llevan un identificador del circuito virtual al que pertenecen
- Todos los paquetes pertenecientes a un mismo circuito virtual siguen el mismo camino y llegan en orden.

Servicio basado en datagramas:



Servicio basado en circuitos virtuales:



Servicio Fiable:

- Se garantiza al nivel superior que todos los paquetes llegan a su destino, y que el destino es capaz de reordenarlos si se desordenan en el camino.
- Para ello se numeran los paquetes, y se retransmiten los perdidos

Servicio No Fiable:

- No se garantiza al nivel superior que todos los paquetes lleguen a su destino: pueden perderse paquetes (típicamente por congestión).
- Algún nivel superior deberá ser capaz de detectar y recuperarse de estas pérdidas, si la aplicación lo requiere.

Todas las combinaciones de tipos de servicio de nivel de red son teóricamente posibles, pero no todas se dan en la práctica.
Las combinaciones más frecuentes son:

- Servicio Orientado a Conexión, basado en Circuitos Virtuales y Fiable (ejemplo: X.25).
- Servicio No Orientado a Conexión, basado en Datagramas y No Fiable (ejemplo: IP).

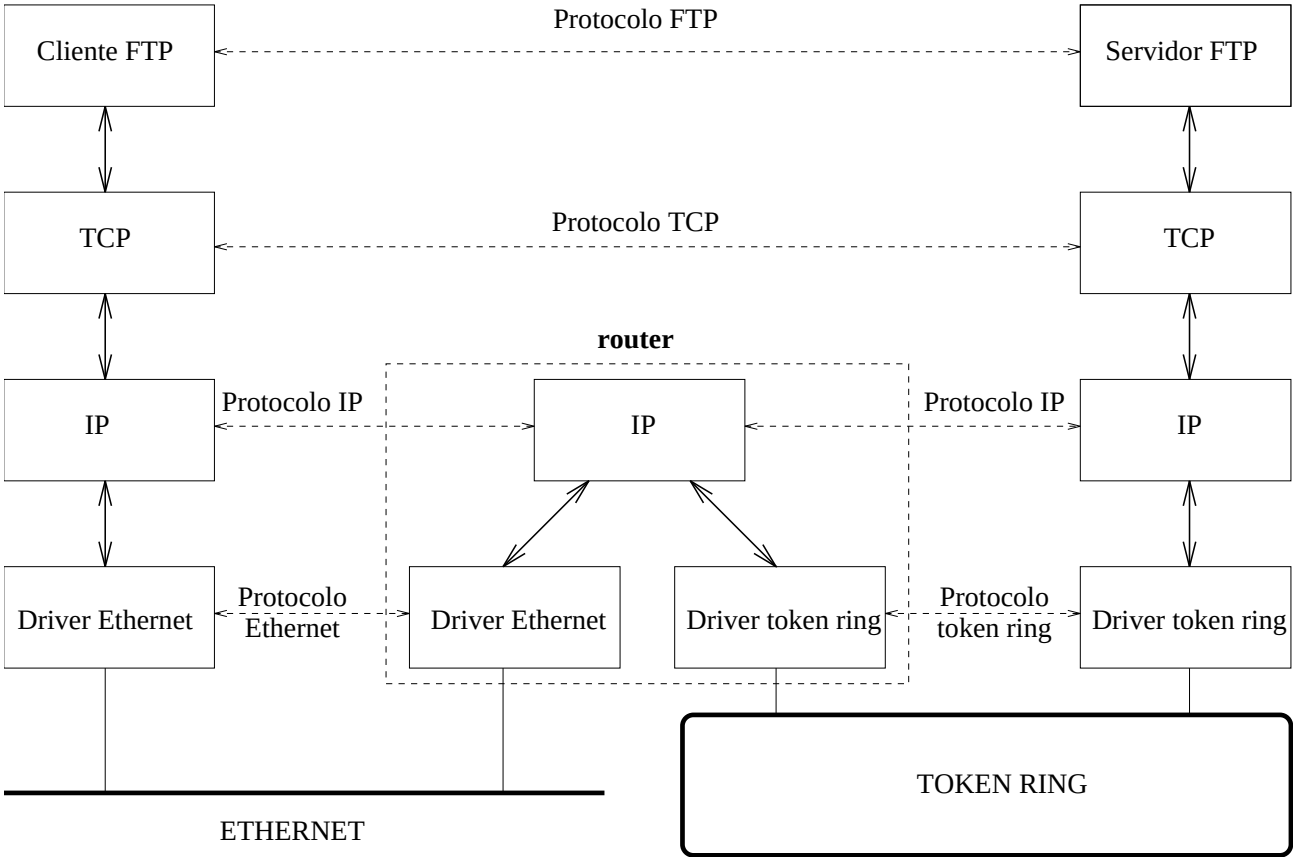
Direcciones de Red

Se necesita un mecanismo de identificación unívoca de todas las máquinas de la red, independientemente de la tecnología del nivel de enlace de cada una.
Existen distintos métodos de direccionamiento según el tipo de redes. Veremos más adelante el formato de las direcciones del nivel de red IP.

Interconexión de subredes con distinto nivel de enlace

- Dependiendo de la arquitectura de red que se trate, puede que se desee integrar en una misma red a subredes con distinto nivel de enlace.
- Cuando así ocurre, es misión de nivel de red hacer esta integración
- Es necesario que el nivel de red del encaminador que une las subredes "entienda" los dos niveles de enlace. Puede tener que resolver problemas de:
  - distintos tamaños de las unidades de datos
  - distintas velocidades de las subredes

Ejemplo


Cliente FTP
Protocolo FTP
Servidor FTP
TCP
Protocolo TCP
TCP
router
IP
Protocolo IP
IP
Protocolo IP
IP
Driver Ethernet
Protocolo Ethernet
Driver Ethernet
Driver token ring
Protocolo token ring
Driver token ring
TOKEN RING
ETHERNET

Encaminamiento

El proceso mediante el cuál se encuentra un camino entre dos puntos cualesquiera de la red
Problemas a resolver: ¿Qué camino escoger? ¿Existe alguno más corto? ¿Qué ocurre si un encaminador o un enlace intermedio se rompen? ...

Algoritmo de encaminamiento: Procedimiento por el cuál los encaminadores (*routers*) alcanzan las decisiones de las mejores rutas para cada destino.
Como parte del algoritmo de encaminamiento, normalmente los encaminadores tienen que enviarse entre sí mensajes de control para conseguir toda la información necesaria.
El resultado de los algoritmos de encaminamiento es generar en cada encaminador su tabla de encaminamiento.
Tabla de encaminamiento: Tabla que consulta el encaminador cada vez que recibe un paquete y tiene que encaminarlo. Esta tabla tiene esta forma:

Destino final	Encaminador vecino al que enviar el paquete
D1	V1
D2	V2
...	...

Muchas veces se utiliza el término Protocolo de Encaminamiento en vez de Algoritmo de Encaminamiento.

Objetivos de un algoritmo de encaminamiento

- Minimizar el espacio de la tabla de encaminamiento para poder buscar rápidamente y para tener menos información a intercambiar con otros encaminadores
- Minimizar el número y frecuencia de mensajes de control
- Robustez: evitar agujeros negros, evitar bucles, evitar oscilaciones en las rutas
- Generar caminos óptimos:
  - menor retardo de tránsito, o
  - camino más corto (en función de una cierta métrica en función de retardo, coste de los enlaces, ...), o
  - máxima utilización de la capacidad de la red

Clasificación de los algoritmos de encaminamiento

- Estáticos: Las tablas de encaminamiento no varían con el tiempo. Para redes en las que el tráfico y la topología no varía.
  - Centralizados: Las tablas se calculan en un sólo nodo de la red.
  - Distribuidos: Las tablas se calculan por cooperación entre todos lo nodos.

Clasificación de los algoritmos de encaminamiento (cont.)

- Dinámicos o Adaptativos: Las tablas de encaminamiento se ajustan a cambios en las condiciones de la red (tráfico o topología).
  - Aislados: Cada nodo toma decisiones independientemente del resto.
  - Centralizados: Un nodo acumula toda la información de la red.
  - Distribuidos: Los nodos cooperan para adaptarse a las condiciones de la red.
  - Mixtos: Un nodo central mantiene información del estado de la red, y aconseja a los otros nodos, que pueden tomar decisiones por su cuenta.

Algoritmo de inundación

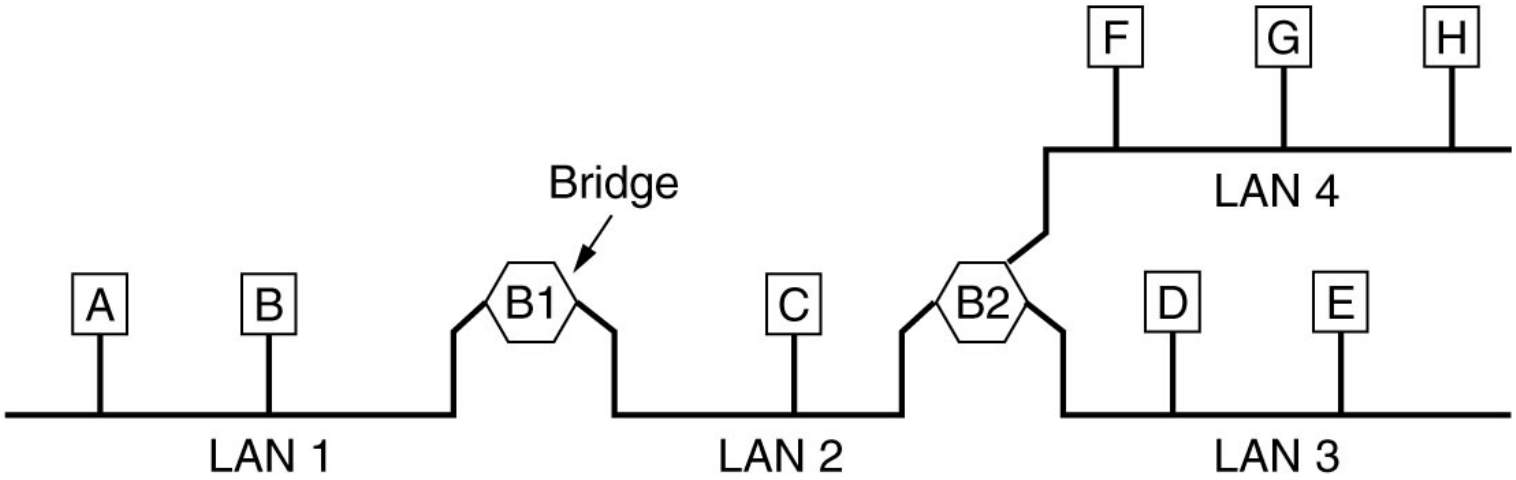
Es un algoritmo simple que a veces se utiliza cuando no hay ninguna información de encaminamiento disponible (al arrancar algún otro algoritmo):

1. Cada paquete recibido por un nodo es encaminado a todos los vecinos (excepto al que lo envió).
2. Los paquetes van etiquetados y numerados.
3. Si un nodo recibe un paquete que ya ha encaminado, lo descarta.

Algoritmo de aprendizaje

Es un algoritmo simple, que mejora el de inundación. Se utiliza también en los *bridges* de nivel de enlace.

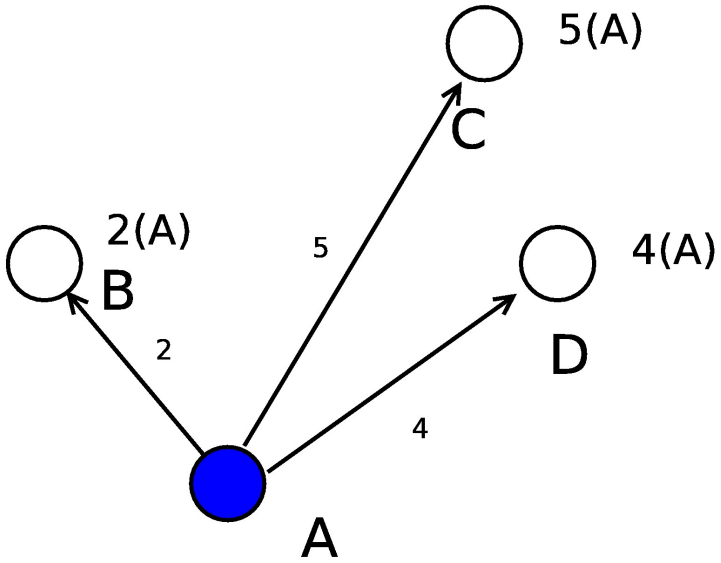
1. Cada nodo mantiene una tabla con entradas
   (Destino, enlace por el que encamino)
   que va actualizando según los paquetes que va recibiendo.
2. Al recibir un paquete, se fija en el nodo origen y enlace por el que le ha llegado, apuntando en la tabla que cuando ese nodo sea destino de un paquete lo encaminará por ese enlace
3. Cuando para un destino no hay entrada en la tabla, se envía por inundación.


F
G
H
Bridge
LAN 4
A
B
B1
C
B2
D
E
LAN 1
LAN 2
LAN 3

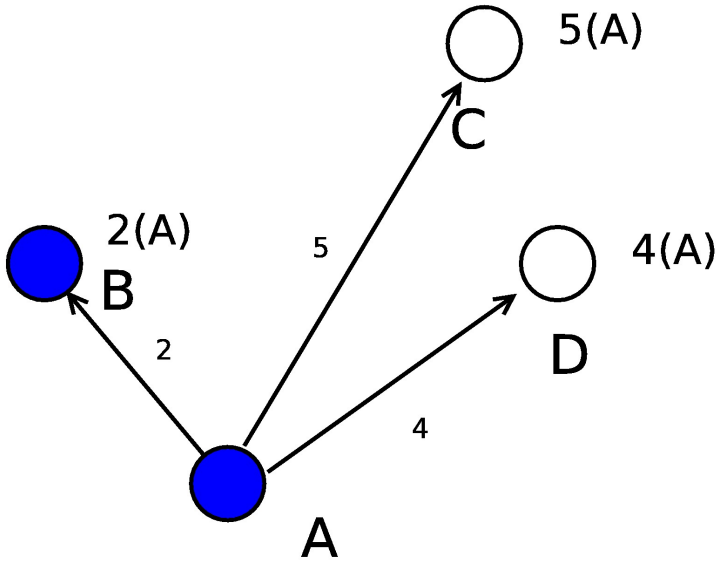
Algoritmo de Dijkstra

Es un algoritmo que encuentra caminos de distancia mínima de un nodo al resto (por lo que cada nodo ejecuta el algoritmo). Requiere conocer todas las distancias entre nodos adyacentes.

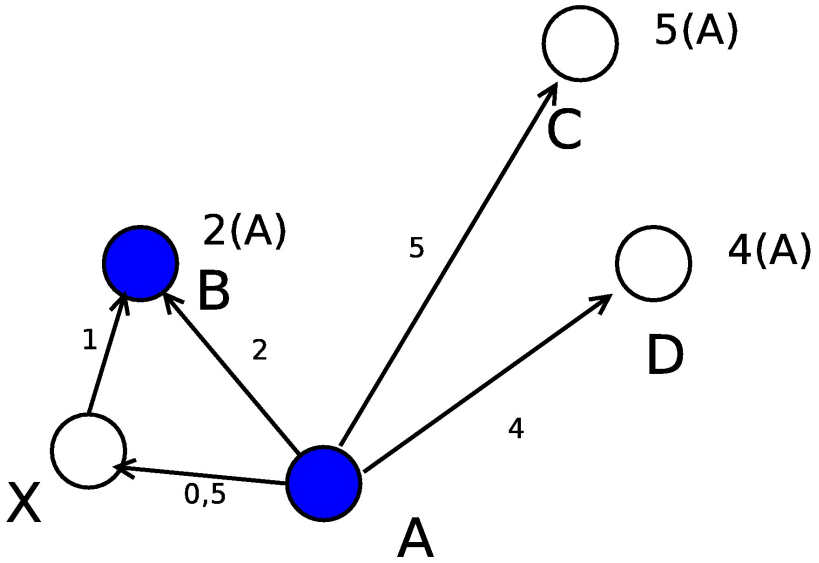
1. Se trabaja con dos conjuntos de nodos:
   - P: Nodos con su encaminamiento ya resuelto (permanentes)
   - T: Nodos aún no resueltos (tentativos)
2. Inicialmente P sólo contiene el nodo inicial
3. Para cada nodo de T:
   - si no está directamente conectado a ningún nodo de P, su distancia al nodo inicial es infinta
   - en caso contrario, se calcula la menor entre la distancia calculada en un paso anterior y las distancias directas entre él y los nodos de P
4. El nodo de T de menor distancia se pasa a P. Si aún quedan nodos en T, se vuelve al paso anterior.



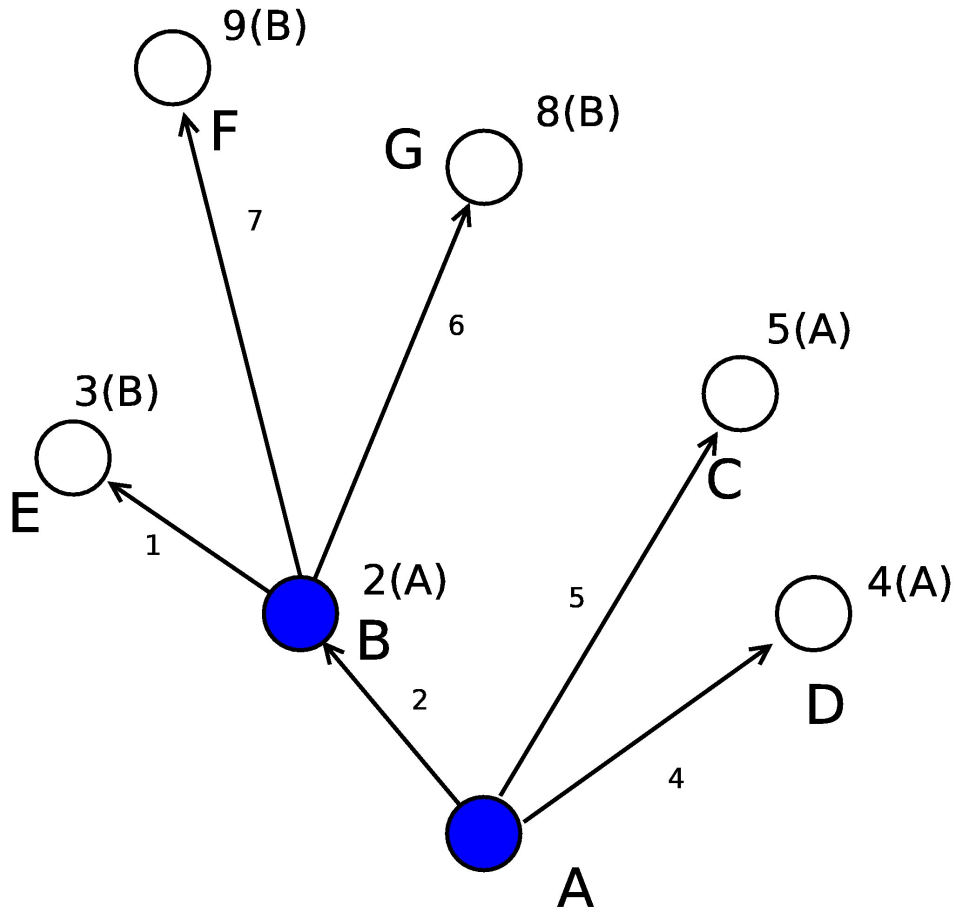
- Indicamos distancia tentativa al origen
- En azul indicamos encaminamiento mínimo ya resuelto



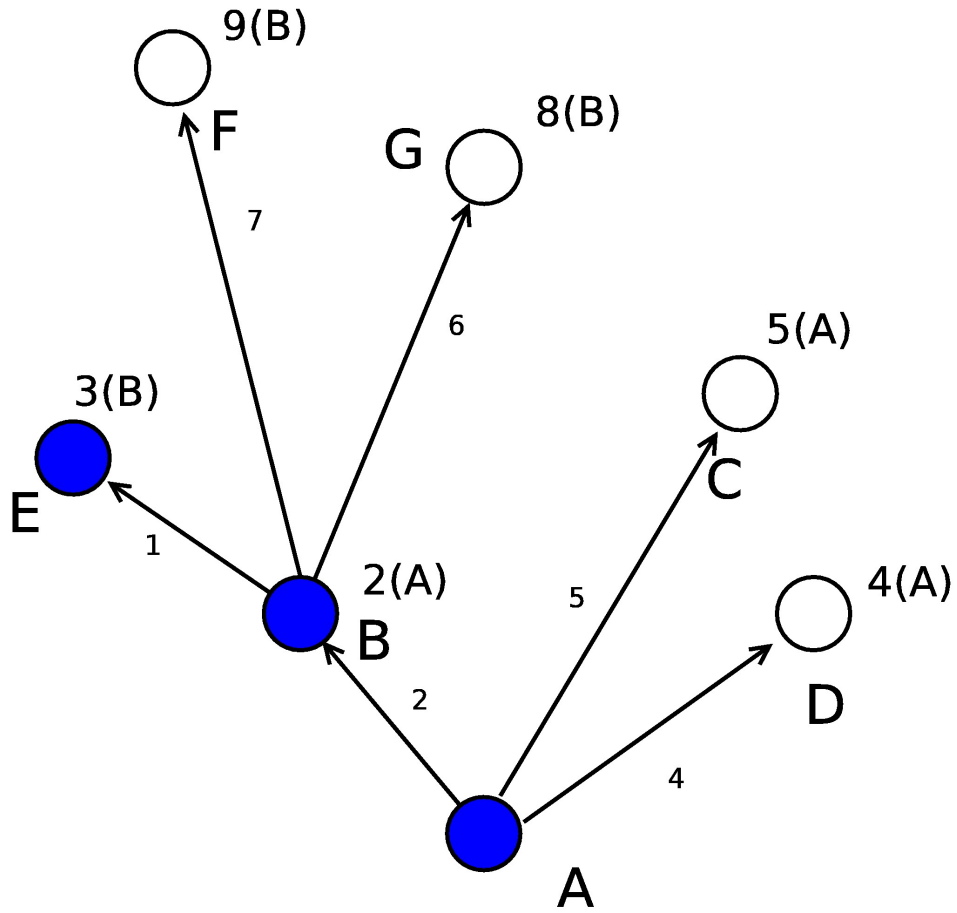
- B tiene la distancia menor, ya es camino mínimo



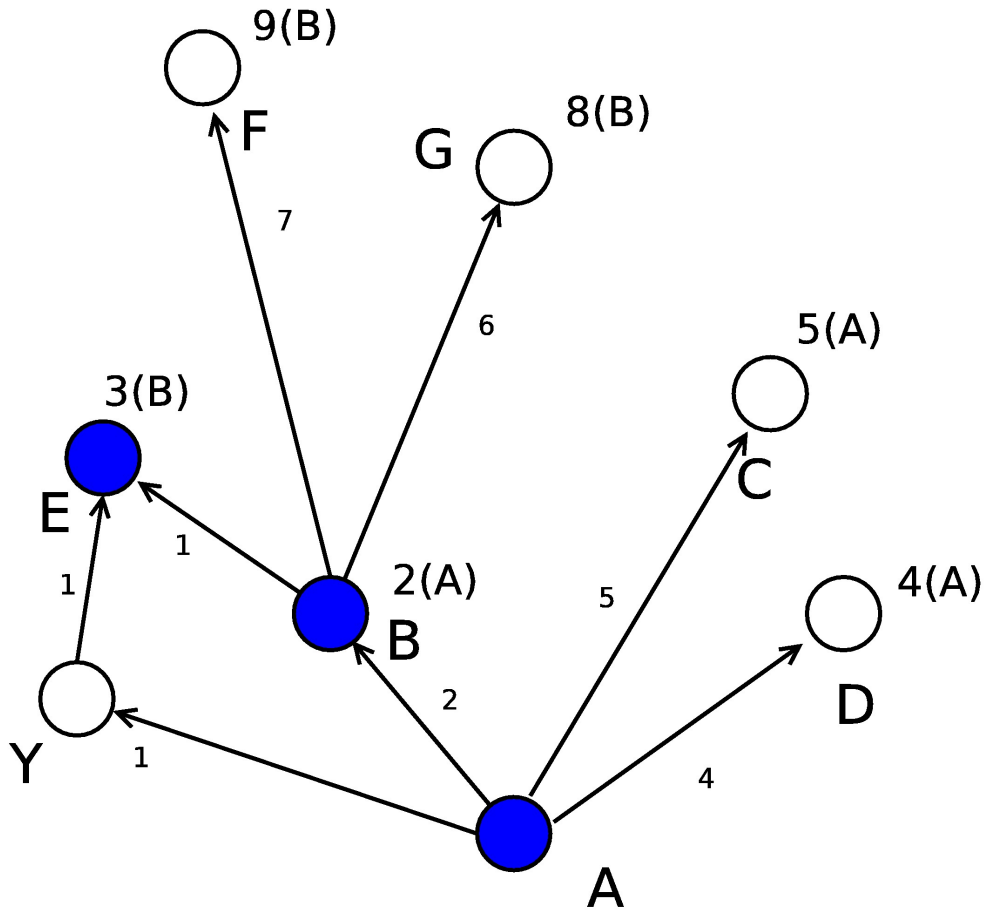
- Supongamos un nodo X que mejore la ruta
- Es imposible. Habríamos considerado el arco AX



- Consideramos B y exploramos todos sus arcos
- Si para algún nodo ya conocíamos un camino mejor, mantenemos el mejor



- Marcamos E como resuelto porque entre los provisionales **de todo el grafo** es el de menor distancia



Supongamos un nodo Y con una ruta menor que aún no conozco

- Si Y estuviera resuelto, ya lo conocería
- Si es tentativo, es mayor

Es imposible que exista Y

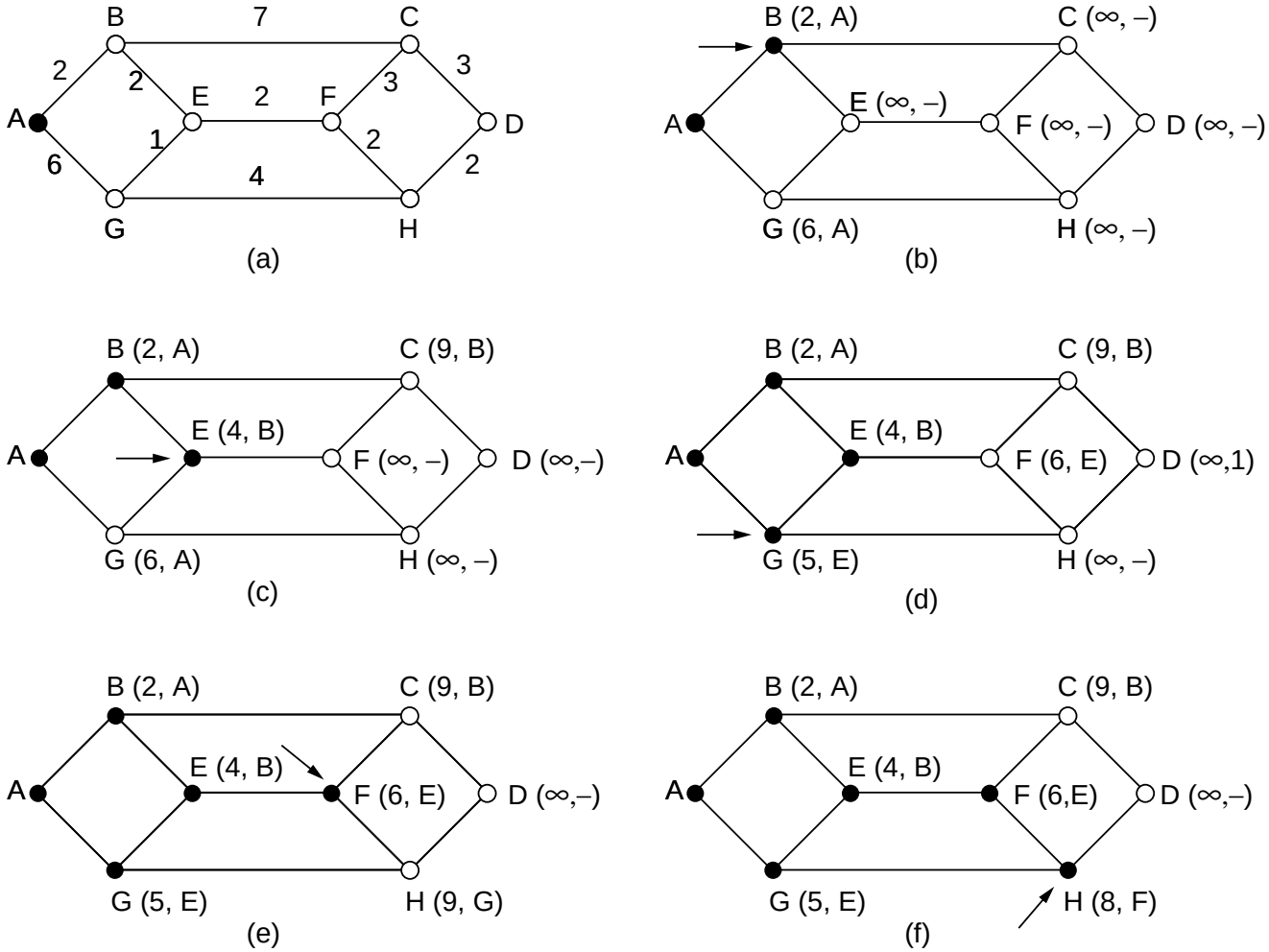


Fig. 5-6. The first five steps used in computing the shortest path from *A* to *D*. The arrows indicate the working node.

Algoritmos basados en Vectores de Distancias

El protocolo RIP (*Routing Information Protocol*), utilizado en Internet, emplea esta técnica.

1. Cada nodo mantiene una tabla de encaminamiento con pares (Destino, Nodo vecino por el que encamino), para todos los destinos de la red.
2. Cada nodo estima el retardo de sus paquetes a los nodos vecinos (enviando periódicamente paquetes de sondeo).
3. Cada nodo envía periódicamente a sus vecinos todos sus pares (Destino, retardo estimado)
4. Cada nodo estudia la información recibida de los vecinos para ver si puede conseguir una ruta de menor retardo enviando a través de otro de sus vecinos, y actualiza sus tablas de encaminamiento consecuentemente

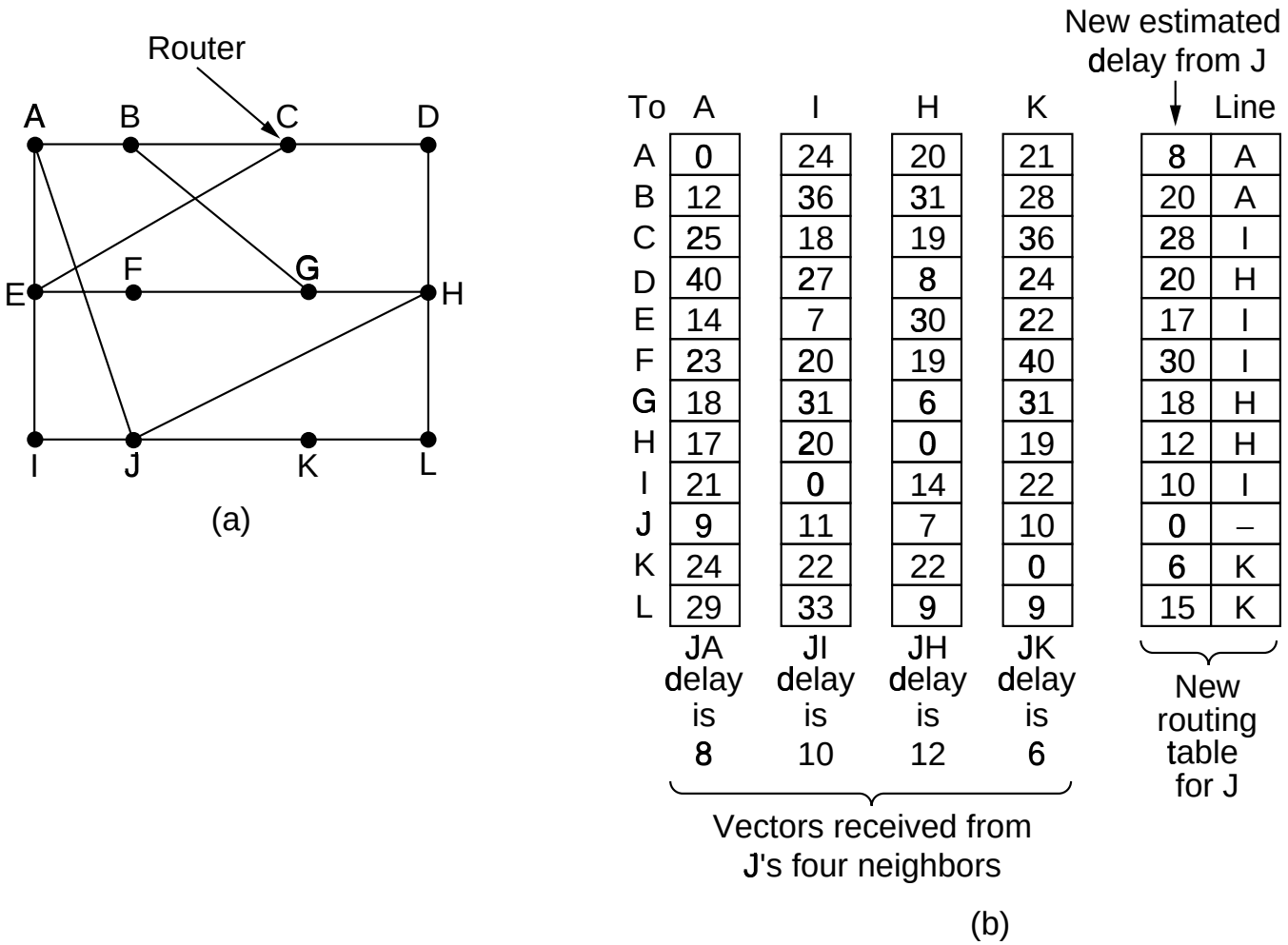


Fig. 5-10. (a) A subnet. (b) Input from *A*, *I*, *H*, *K*, and the new routing table for *J*.

Problema: cuenta al infinito.

La información acerca de mejores rutas se propaga poco a poco, consiguiéndose al cabo de un rato que todos los encaminadores tengan tablas óptimas

Pero las malas noticias (se cae un enlace o un encaminador) tardan en llegar:

	MADRID	15	**LEGANES**	5	**FUENLA.**	10	**GRIÑON**
1	0		∞		∞		∞
2	0		15		∞		∞
3	0		15		20		∞
4	0		15		20		30
5	0	90	25		30		40
6	0		35		40		50
7	0		45		50		60
8	0		55		60		70

La tabla anterior representa un grafo muy sencillo: cuatro nodos alineados.

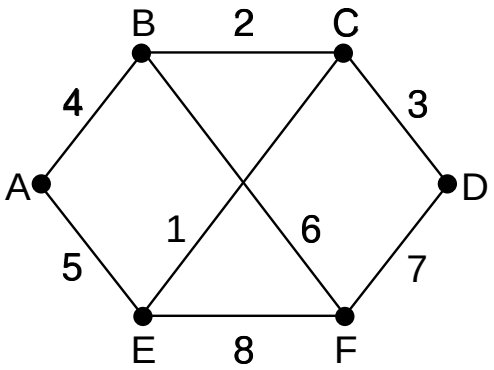
- En el momento t_0, un observador omnisciente ajeno a la red sabría que se tardan 15 minutos en ir desde Madrid a Leganés, 5 minutos desde Leganés a Fuenlabrada y 10 minutos desde Fuenlabrada hasta Griñon.
- En este modelo, suponemos que el tiempo entre A y B es el mismo que entre B y A (lo que en la vida real sería frecuentemente falso)
- El objetivo es que la red conozca en todo momento, para cualquier origen, a qué distancia está Madrid y en qué dirección hay que ir
- En cada iteración del algoritmo, cada nodo intercambia con sus vecinos inmediatos su distancia (en minutos) hasta Madrid y cada nodo calcula sus vecinos inmediatos

1. La información en la red es nula
2. Tras intercambiar un paquete con sus vecinos, Leganés sabe que Madrid está a 15'
3. Tras intercambiar otro paquete de datos, Fuenlabrada sabe que a través de Leganes puede llegar a Madrid en 5+15=20'
4. Griñon sabe que se puede llegar a Madrid, en 30', tomando la dirección de Fuenlabrada
5. La red cambia. Desde Leganés, ahora se tardan 90' en llegar a Madrid. Pero Fuenlabrada, que ignora esto, comunica a Leganés que es el trayecto Fuanlabrada-Madrid dura 20'. Por tanto, Leganés supone que puede llegar a Madrid, tomando la direcciónd de Fuenlabrada, en 5+20=25'
   En este punto, el algoritmo ha fallado
   Fuenlabrada piensa que puede llegar a Madrid en 30', siguiendo la ruta Fuenlabrada-Leganés-Fuenlabrada-Leganés.
   Un bucle sin sentido, y con información erronea
6. La red va empeorando los tiempos, acabará llegando a los valores reales, pero muy lentamente

Algoritmos basados en el Estado de Enlace

El protocolo OSPF (*Open Shortest Path First*), utilizado en Internet, emplea esta técnica.

1. Cada encaminador mide su distancia con cada uno de sus vecinos (enviando paquetes de sondeo) y construye un paquete de estado de enlaces con esta información.
2. Cada encaminador envía perdiódicamente el paquete de estado de enlaces a todos los encaminadores de la red. Estos mensajes se difunden por inundación.
3. Cada encaminador, con la información recibida, conoce la topología completa de la red y calcula el mejor camino a todos sus destinos (aplicando, por ejemplo, el algoritmo de Dijkstra)



(a)

Link State Packets

A	B	C	D	E	F
Seq.	Seq.	Seq.	Seq.	Seq.	Seq.
Age	Age	Age	Age	Age	Age
B 4	A 4	B 2	C 3	A 5	B 6
E 5	C 2	D 3	F 7	C 1	D 7
	F 6	E 1		F 8	E 8

(b)

Fig. 5-15. (a) A subnet. (b) The link state packets for this subnet.

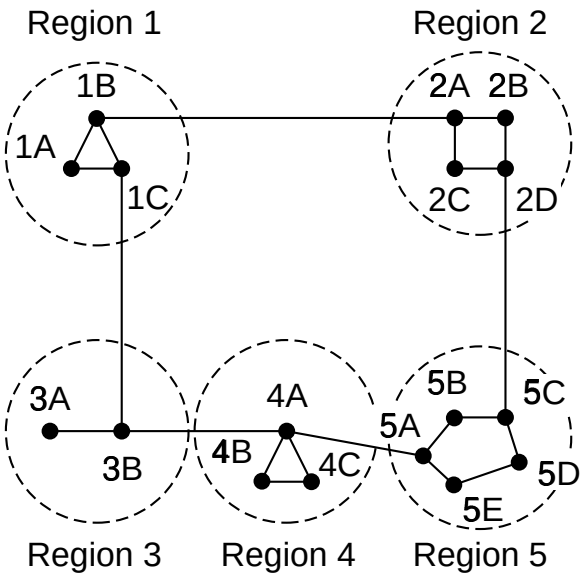
Encaminamiento jerárquico

Si la red es muy grande, las tablas de encaminamiento se hacen inmanejables:

- Se tarda mucho en calcular los caminos óptimos
- Se genera mucho tráfico de control para conseguir difundir la información necesaria para los algoritmos de encaminamiento

Solución: Encaminamiento Jerárquico:

- Se divide la red en dominios
- Dentro de cada dominio se encamina según un algoritmo de los vistos anteriormente
- Los dominios están interconectados mediante pasarelas (*gateways*)
- Las máquinas dentro de un dominio no conocen a las de otro
- Los *gateways* sólo conocen a otros gateways



(a)

Full table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

(b)

Hierarchical table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

(c)

Fig. 5-17. Hierarchical routing.