Encaminamiento en Internet
1. Introducción
Redes-I

Departamento de Sistemas Telemáticos y Computación (GSyC)

Octubre 2010

Contenidos

Contenidos

Terminología

Algoritmo de encaminamiento

Procedimiento por el cual 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: Protocolo de encaminamiento.
- Muchas veces se utiliza el término Protocolo de Encaminamiento como sinónimo de Algoritmo de Encaminamiento.
- El resultado de los algoritmos de encaminamiento es generar en cada encaminador una Tabla de encaminamiento.

Terminología

Tabla de encaminamiento

Tabla que consulta el encaminador cada vez que recibe un paquete y tiene que encaminarlo.

Destino	Encaminador vecino	Máscara	Interfaz
D1	V1	255.255.255.0	eth0
D2	V2	255.255.255.0	eth1
...	...	...	...

Objetivos de un algoritmo de encaminamiento

- Minimizar el espacio de la tabla de encaminamiento para poder buscar rápidamente
- Minimizar la información adicional de encaminamiento a almacenar a fin de ejecutar el algoritmo
- Minimizar el número y frecuencia de mensajes que se envían a otros encaminadores a fin de ejecutar el algoritmo
- Robustez: evitar agujeros negros, evitar bucles, evitar oscilaciones en las rutas
- Generar caminos óptimos, es decir, de mínimo coste, definiendo el coste en base a uno o más de estos parámetros:
  - número de encaminadores intermedios
  - retardo
  - coste económico
  - aprovechamiento de la capacidad de la red

Contenidos

Algoritmo de inundación

- Es un algoritmo simple que a veces se utiliza cuando no hay ninguna información de encaminamiento disponible (por ejemplo, al arrancar algún otro algoritmo):
  1. Cada paquete recibido por un nodo es reenviado a todos los vecinos excepto al que se lo envió a él.
  2. Los paquetes van etiquetados y numerados.
  3. Si un nodo recibe un paquete que ya ha reenviado, lo descarta.

Algoritmo de aprendizaje

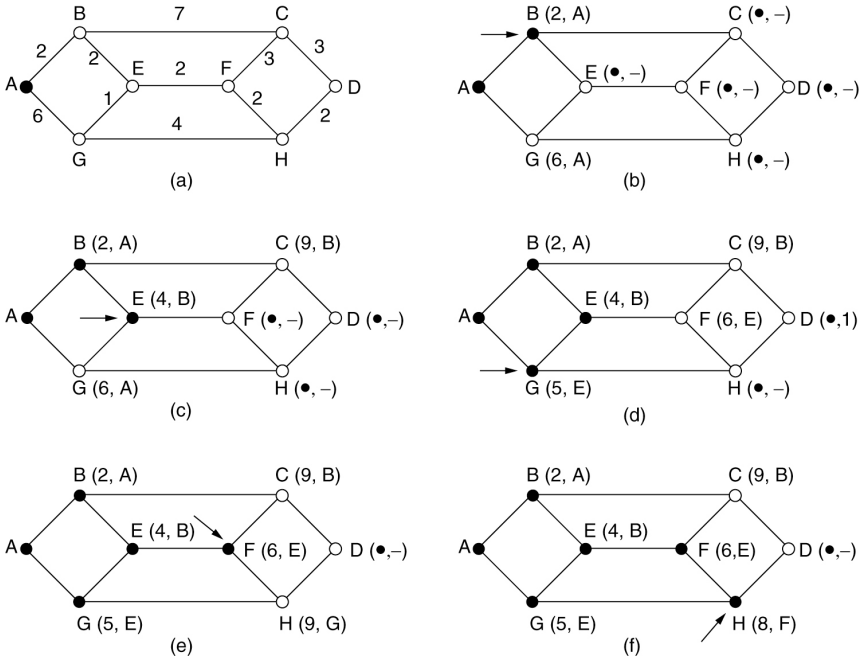
- Es un algoritmo simple, que mejora el de inundación. Es el utilizado por los *bridges*.
  1. Cada nodo mantiene una tabla con parejas **(Destino, enlace por el que encamino)** que va actualizando según los paquetes que va recibiendo.
  2. Al recibir un paquete, se fija en el nodo origen y en el enlace 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.

Algoritmo de Dijkstra

- Algoritmo que encuentra caminos de distancia mínima de un nodo al resto. Cada nodo ejecuta el algoritmo para encontrar caminos desde él al resto.
- Requiere conocer todas las distancias entre nodos adyacentes.
- Algoritmo:
  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 se recalcula su distancia al nodo inicial:
     - si no está directamente conectado a ningún nodo de P, su distancia al nodo inicial es infinita
     - en caso contrario, se elige la menor entre la distancia calculada en un paso anterior y la suma entre la distancia calculada para el último nodo añadido a P y la distancia directa de ese nodo a éste
  4. El nodo de T que presente una menor distancia se pasa a P. Si aún quedan nodos en T, se repite el paso anterior.

Algoritmo de Dijkstra: Ejemplo

La figura muestra los 5 primeros pasos utilizados en calcular el camino más corto desde A a D. La flecha indica el nodo sobre el que se está actuando:



Contenidos

Familias de Protocolos de Encaminamiento

- Los protocolos de encaminamiento más usados en redes TCP/IP pueden clasificarse en dos grupos:
  - Protocolos de Vector de Distancia *(Distance Vector Protocols)*
  - Protocolos de Estado de Enlace *(Link State Protocols)*

Contenidos

Funcionamiento básico

1. Cada nodo conoce o estima el coste para llegar a sus nodos vecinos.
2. Cada nodo mantiene su tabla de encaminamiento con triplas de la forma:

   (Destino, Coste, Vecino por el que encamino)

   para todos los destinos de la red
3. Cada nodo envía periódicamente **a sus vecinos** su Vector de Distancia **a todos los destinos**, formado por los pares:

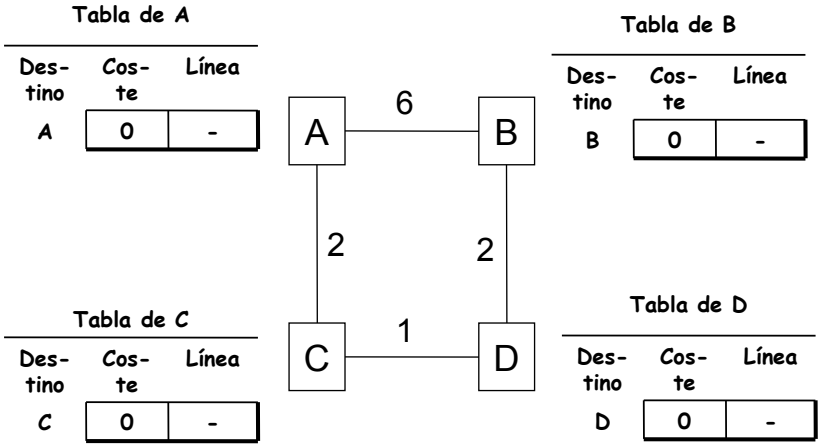
   (Destino, Coste)
4. Cada nodo estudia los vectores de distancia que recibe de sus vecinos para seleccionar para cada destino el vecino por el que tendrá menor coste, y actualiza su tablas de encaminamiento consecuentemente.

- El protocolo RIP pertenece a esta familia de protocolos.

Ejemplo

INICIALMENTE

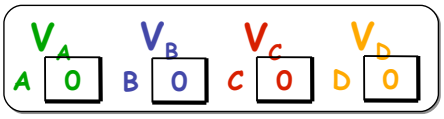
Inicialmente cada nodo sólo se conoce a sí mismo.



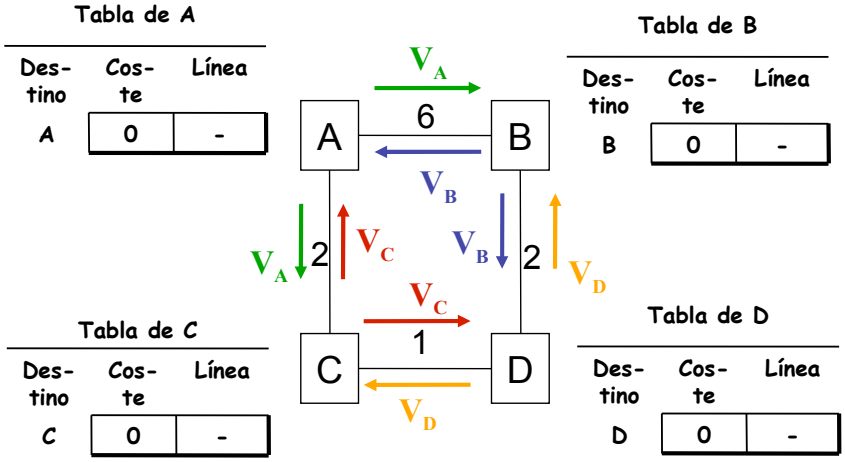
Ejemplo

INTERCAMBIO 1

Cada nodo intercambia su vector distancia con sus vecinos.



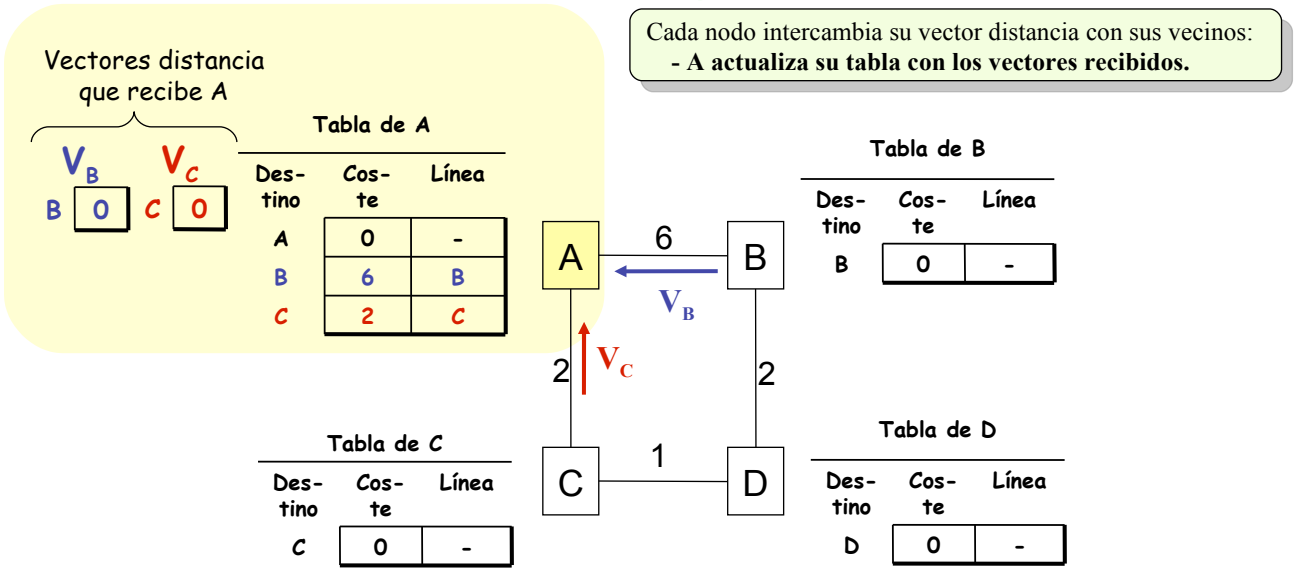
Vectores distancia iniciales que envía cada una de las máquinas



Ejemplo

INTERCAMBIO 1

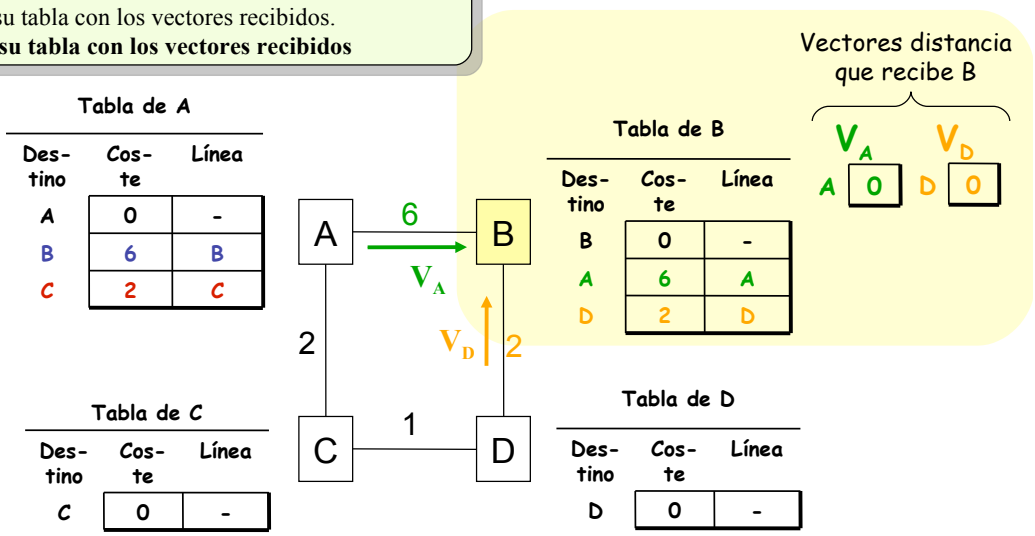
Cada nodo intercambia su vector distancia con sus vecinos:
- **A actualiza su tabla con los vectores recibidos.**



Ejemplo

INTERCAMBIO 1

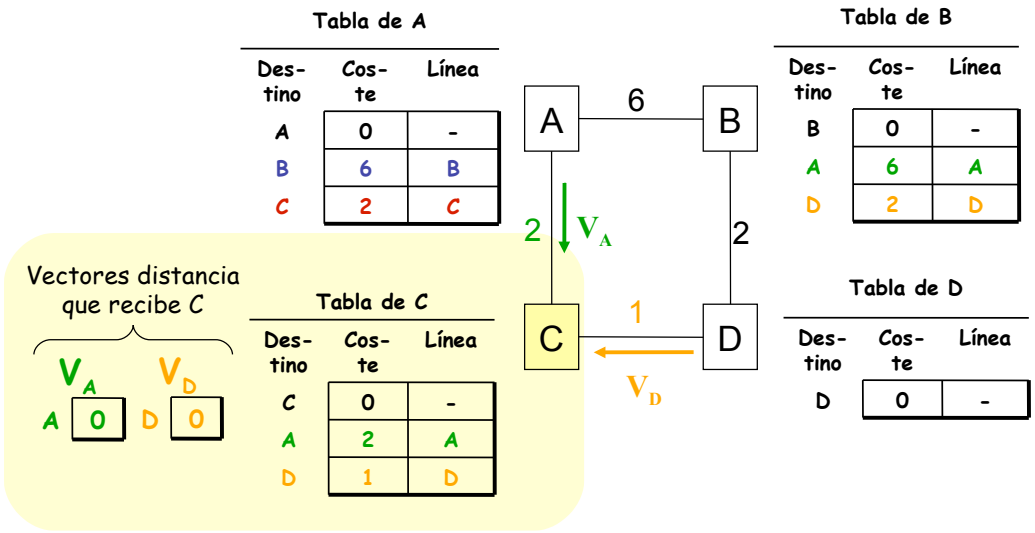
Cada nodo intercambia su vector distancia con sus vecinos:
- A actualiza su tabla con los vectores recibidos.
- **B actualiza su tabla con los vectores recibidos**



Ejemplo

INTERCAMBIO 1

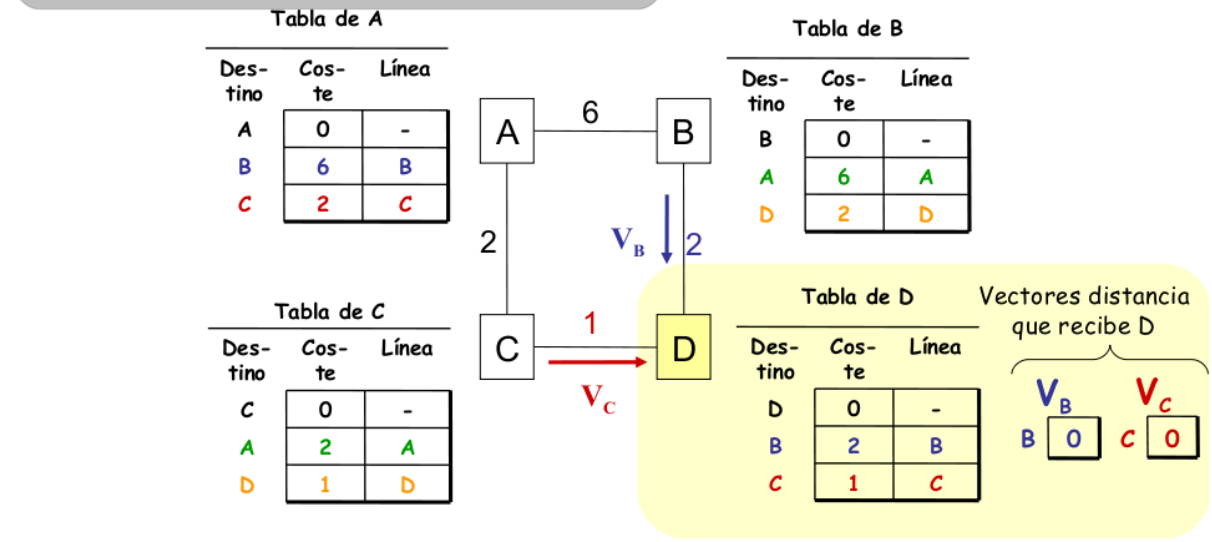
Cada nodo intercambia su vector distancia con sus vecinos:
- A actualiza su tabla con los vectores recibidos.
- B actualiza su tabla con los vectores recibidos.
- **C actualiza su tabla con los vectores recibidos**



Ejemplo

INTERCAMBIO 1

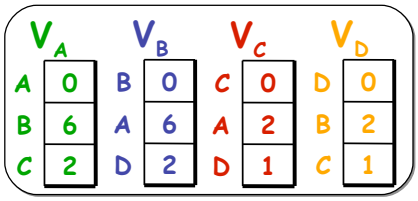
Cada nodo intercambia su vector distancia con sus vecinos:
- A actualiza su tabla con los vectores recibidos.
- B actualiza su tabla con los vectores recibidos.
- C actualiza su tabla con los vectores recibidos.
- **D actualiza su tabla con los vectores recibidos**



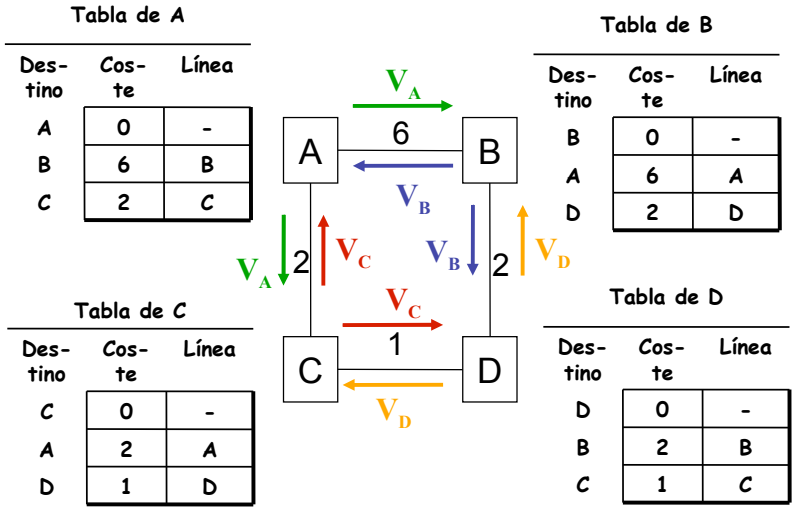
Ejemplo

INTERCAMBIO 2

Cada nodo intercambia su vector distancia con sus vecinos.



Vectores distancia que envía cada una de las máquinas

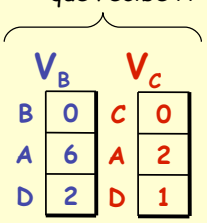


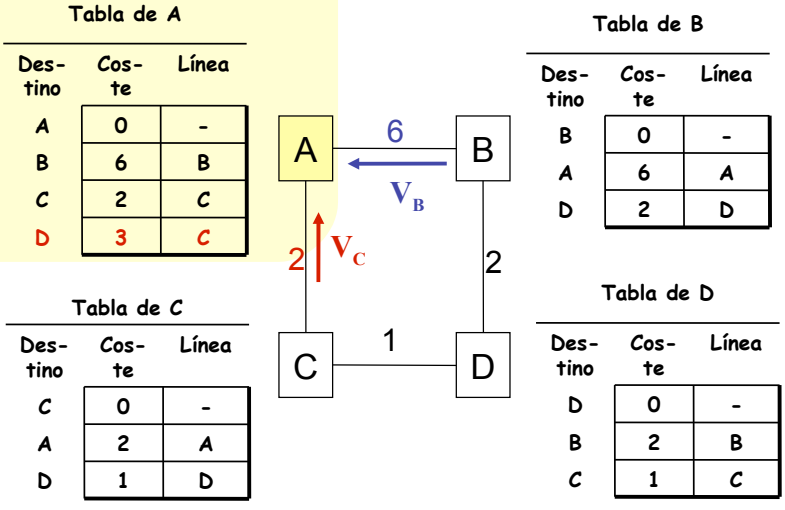
Ejemplo

INTERCAMBIO 2

Cada nodo intercambia su vector distancia con sus vecinos:
- A actualiza su tabla con los vectores recibidos.

Vectores distancia que recibe A

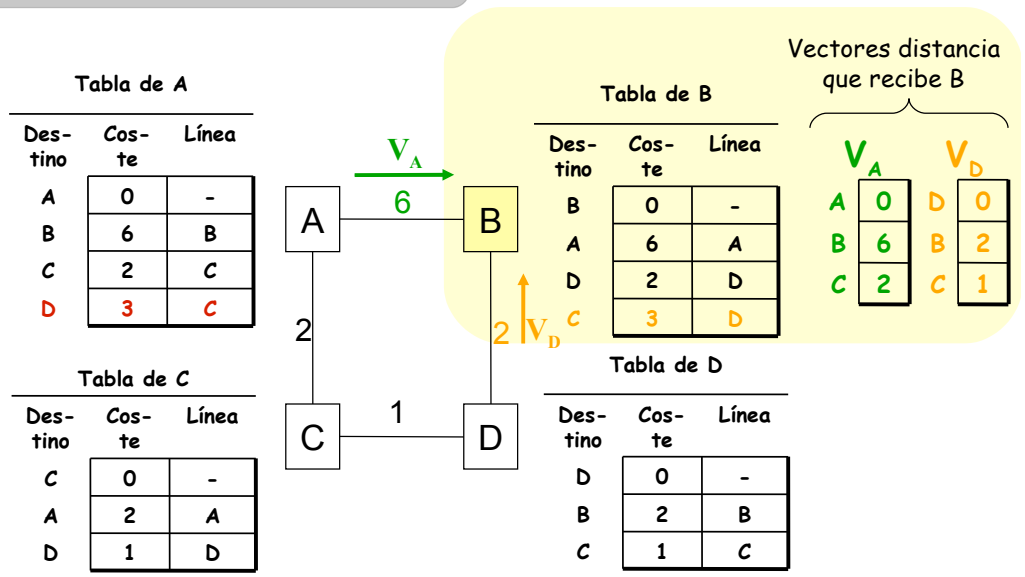




Ejemplo

INTERCAMBIO 2

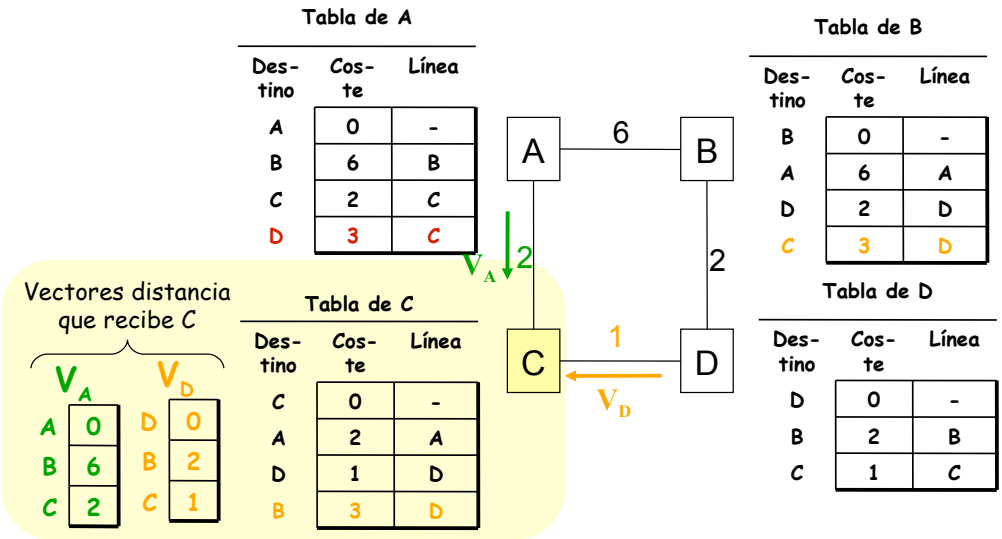
Cada nodo intercambia su vector distancia con sus vecinos:
- A actualiza su tabla con los vectores recibidos.
- **B actualiza su tabla con los vectores recibidos**



Ejemplo

INTERCAMBIO 2

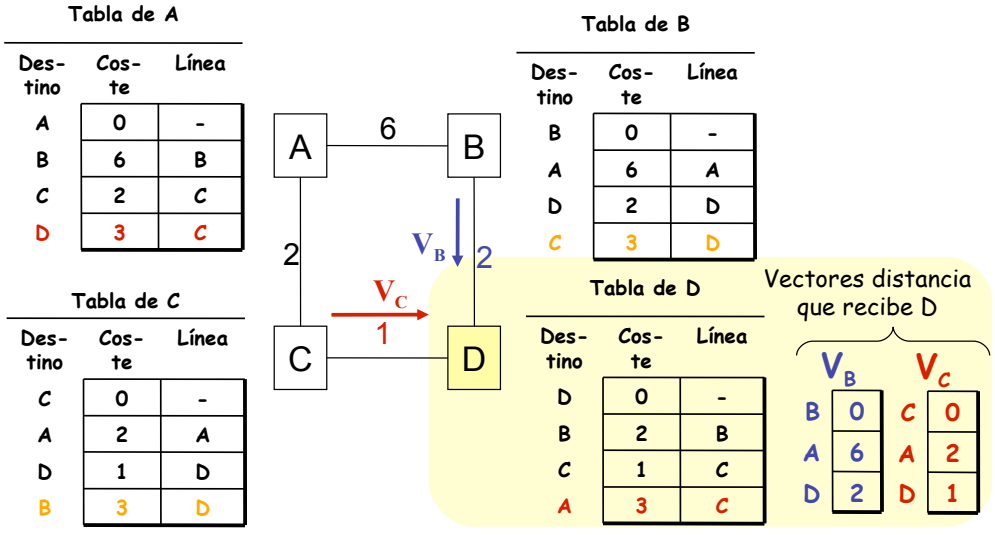
Cada nodo intercambia su vector distancia con sus vecinos:
- A actualiza su tabla con los vectores recibidos.
- B actualiza su tabla con los vectores recibidos.
- **C actualiza su tabla con los vectores recibidos**



Ejemplo

INTERCAMBIO 2

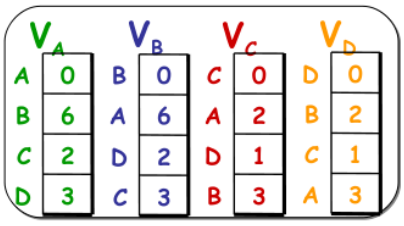
Cada nodo intercambia su vector distancia con sus vecinos:
- A actualiza su tabla con los vectores recibidos.
- B actualiza su tabla con los vectores recibidos.
- C actualiza su tabla con los vectores recibidos.
- **D actualiza su tabla con los vectores recibidos**



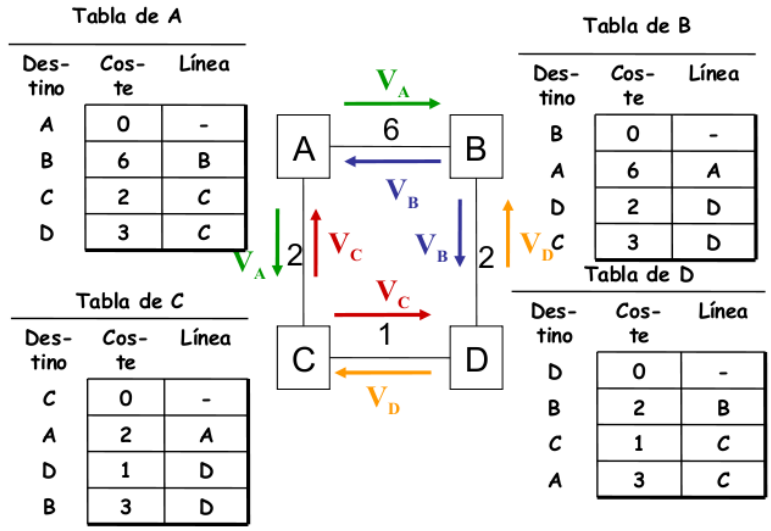
Ejemplo

INTERCAMBIO 3

Cada nodo intercambia su vector distancia con sus vecinos.



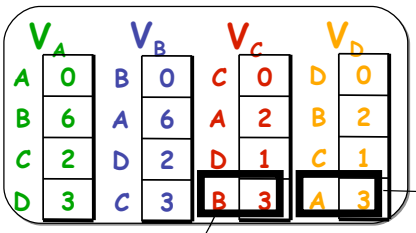
Vectores distancia que envía cada una de las máquinas



Ejemplo

INTERCAMBIO 3

Cada nodo intercambia su vector distancia con sus vecinos:
- **A y B actualizan sus tablas con los vectores recibidos**
- **C y D no necesitan actualizarlas**



Vectores distancia que envía cada una de las máquinas

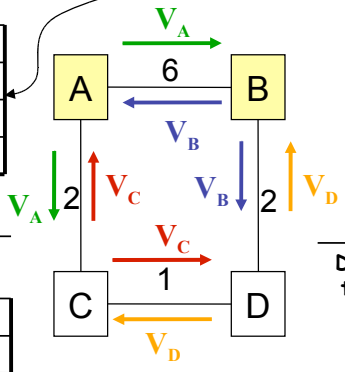


Tabla de A

Destino	Coste	Línea
A	0	-
B	5	C
C	2	C
D	3	C

Tabla de B

Destino	Coste	Línea
B	0	-
A	5	D
D	2	D
C	3	D

Tabla de C

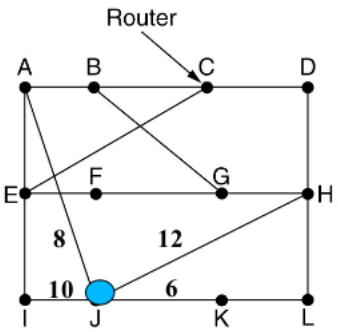
Destino	Coste	Línea
C	0	-
A	2	A
D	1	D
B	3	D

Tabla de D

Destino	Coste	Línea
D	0	-
B	2	B
C	1	C
A	3	C

Ejercicio

En la siguiente figura, J recibe los vectores distancia de sus nodos vecinos A, I, H y K. Según los datos que aparecen en la figura, actualizar los costes para llegar a todos los destinos en el nodo J:



(a)

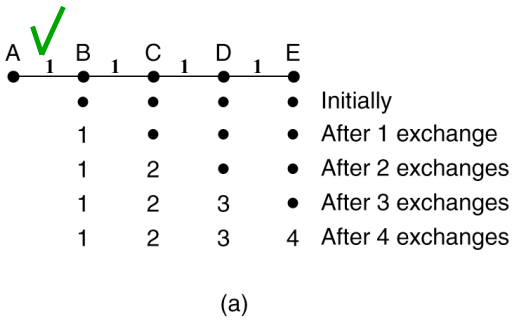
New estimated delay from J

To	A	I	H	K	Line
A	0	24	20	21	A
B	12	36	31	28	A
C	25	18	19	36	I
D	40	27	8	24	H
E	14	7	30	22	I
F	23	20	19	40	I
G	18	31	6	31	H
H	17	20	0	19	H
I	21	0	14	22	I
J	9	11	7	10	–
K	24	22	22	0	K
L	29	33	9	9	K
	JA delay is 8	JI delay is 10	JH delay is 12	JK delay is 6	New routing table for J

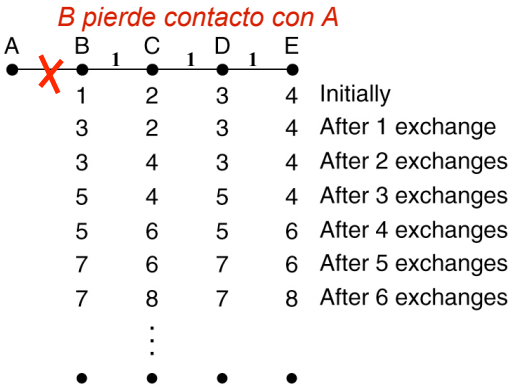
Vectors received from J's four neighbors

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.
  - En la figura (a); B, C, D y E van aprendiendo secuencialmente el coste hacia A.
- Pero las malas noticias (se cae un enlace o un encaminador) tardan en llegar.
  - En la figura (b): B pierde el contacto con A, sin embargo B aprende con el vector de distancia de C que puede alcanzar a A a través de C (cuyo coste será el recibido en el vector de distancia más el coste entre B y C). A continuación C recibe el vector de distancia de B cuyo coste para llegar a A ha aumentado y C actualiza el coste para llegar a A a través de B, etc,



(a)



(b)

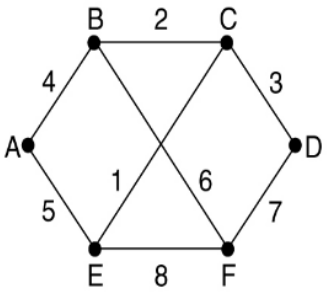
Contenidos

Funcionamiento básico

1. Cada encaminador conoce o estima el coste para llegar a sus nodos vecinos y construye un paquete de Estado de Enlace (LSP, *Link State Packet*) con esta información.
2. Cada encaminador envía periódicamente **a todos** los encaminadores de la red el paquete de Estado de Enlace con el coste para llegar **a sus vecinos**. 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)

- El protocolo OSPF pertenece a esta familia de protocolos.

Ejemplo



(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)

(a) Ejemplo de subred

(b) Paquetes de estado de enlace para esa subred

Ejemplo

- Cada nodo recibe el LSP de cada uno de los demás nodos.
- Cada nodo aplica Dijkstra con la información recibida y construye su tabla de encaminamiento

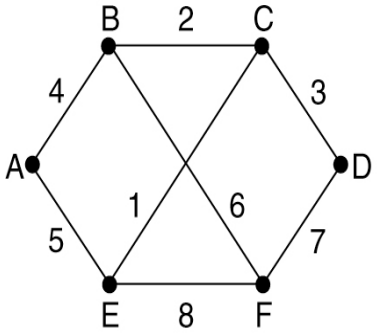


Tabla de A

Destino	Coste	Gateway
A	0	–
B	4	B
C	6	B
D	9	E
E	5	E
F	10	B

Ventajas

- Convergen más rápido y sin bucles.
- Permiten usar varias métricas para calcular el mejor camino.
- Permiten obtener varias rutas alternativas para un mismo destino: balanceo de tráfico...
- Representan mejor las “rutas hacia el exterior” en redes con muchos nodos.

Problemas

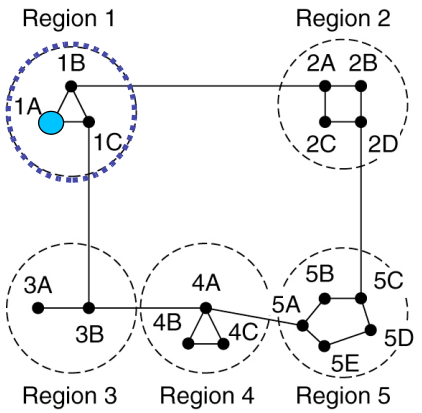
- Son mucho más complejos que los de Vector de Distancia.
- Es imprescindible asegurar la consistencia de las tablas de encaminamiento. Si distintos nodos llegaran a construir tablas distintas, la situación sería desastrosa.
- La información que se recibe de cada *router* hay que guardarla en una base de datos y ésta puede ser grande. Sobre esa base de datos se tiene que aplicar Dijkstra.
- Pueden necesitarse muchos mensajes para propagar la información de esa base de datos.
- Necesitan abordar problemas de seguridad.

Contenidos

Idea

- 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.
  - Las rutas entre gateways se calculan con otro algoritmo de encaminamiento.

Ejemplo



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

Hierarchical table for 1A

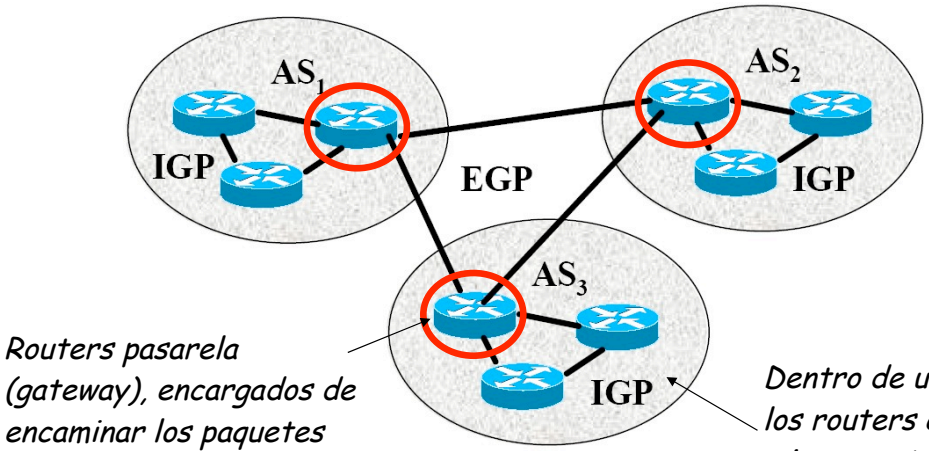
Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

Protocolos de encaminamiento en Internet: Protocolos Interiores y Exteriores

- A principios de los 80 Internet era una sola red desde el punto de vista administrativo. Las tablas mantenían entradas para todas las subredes. Problemas:
  - Escalabilidad
  - Autonomía administrativa
- En 1982 se decide agrupar subredes en Sistemas Autónomos (AS) y eliminar la centralización administrativa:
  - Uno de los AS es el backbone y a él se conecta al menos un router de cada uno de los otros AS (representante).
  - Cada AS ejecuta Protocolos Interiores de Encaminamiento (IGPs, Interior Gateway Protocol) para sus subredes:
    - RIP (Routing Information Protocol)
    - OSPF (Open Shortest Path First)
  - Los routers representantes de cada AS ejecutan un Protocolo Exterior de Encaminamiento (EGPs, Exterior Gateway Protocol) para la conexión de los mismos:
    - BGP (Border Gateway Protocol).

AS, IGP y EGP

- Relación entre sistemas autónomos y protocolos interiores y exteriores de encaminamiento:



Routers pasarela (gateway), encargados de encaminar los paquetes hacia otros AS.

Dentro de un AS, todos los routers ejecutan el mismo protocolo de encaminamiento

Contenidos

Referencias

- Andrew S. Tanenbaum, **Redes de Computadores**, Prentice Hall, 4ª edición: apartado 5.2.
- J.J. Kurose y K.W. Ross, **Redes de Computadores: un enfoque descendente basado en Internet**, Pearson Educación, 2ª edición: apartados 4.2, 4.3.