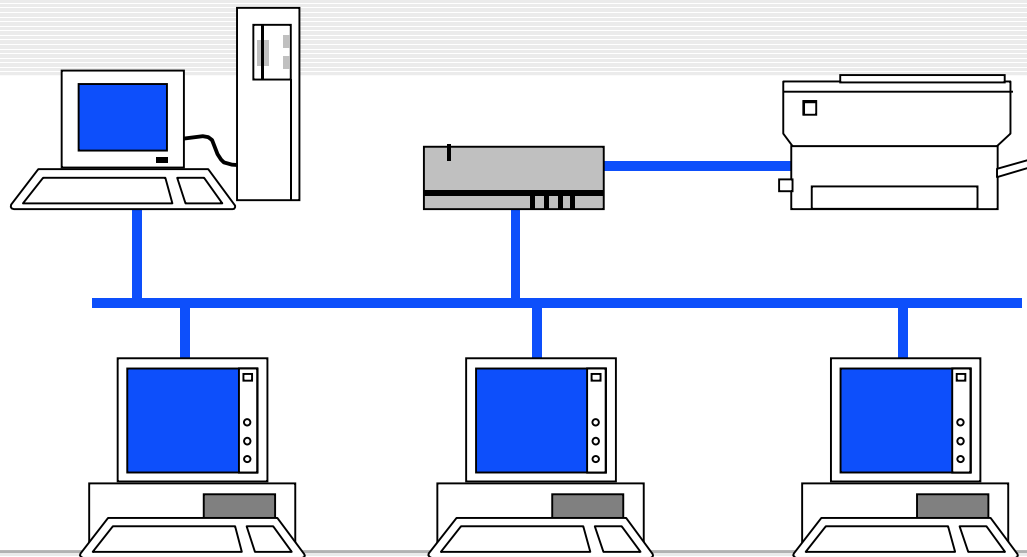


# Redes de Computadores

## Capa de Red





# Algoritmos de Enrutamiento

- **Ruteamiento del Camino más Corto**
- Ruteamiento Multitrayecto
- Ruteamiento Centralizado
- Ruteamiento Aislado
- Inundación
- Ruteamiento basado en el Flujo
- Ruteamiento Distribuido (Vector Distancia)
- Ruteamiento por estado de enlace
- Ruteamiento Jerárquico
- Ruteamiento por Difusión



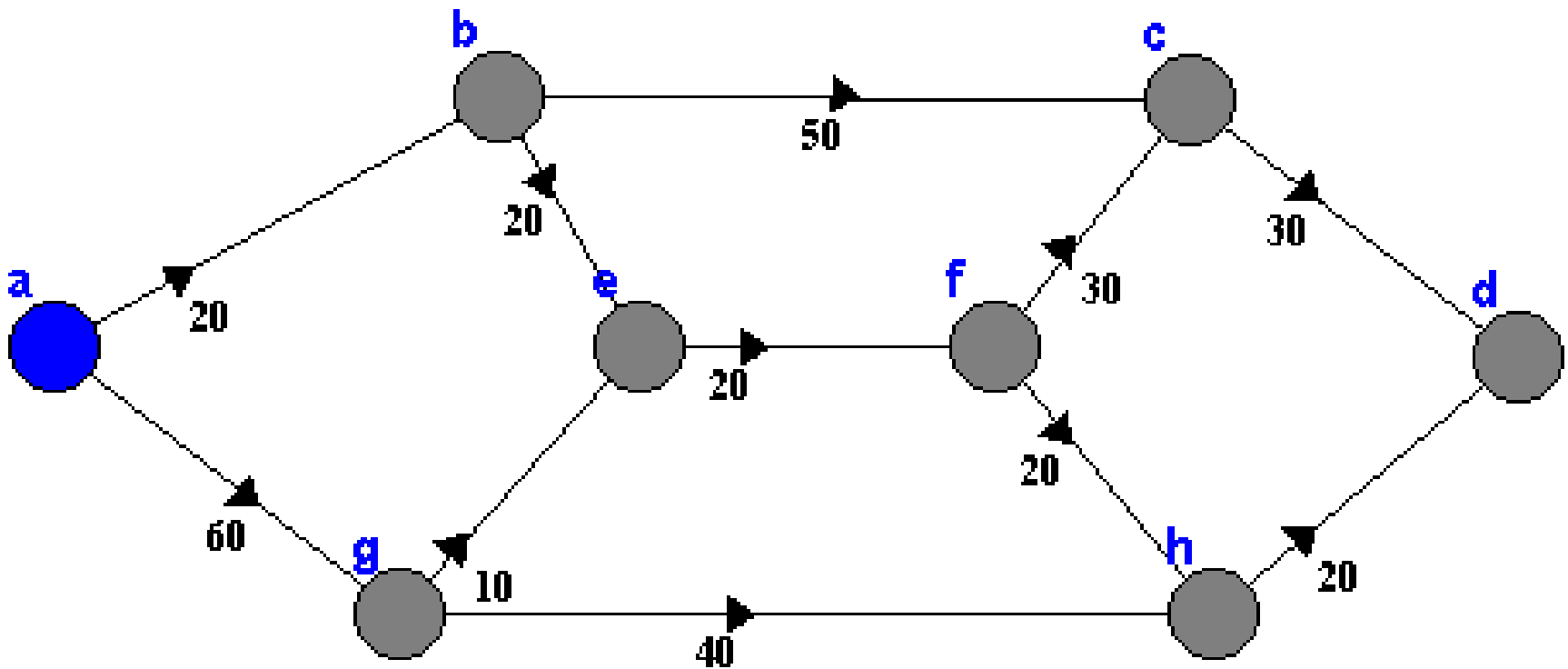
## Algoritmo del Camino más corto o del Mínimo Costo

- Es uno de los algoritmos más utilizados en la práctica
- Se asocia un “Peso” o “Costo” a cada enlace de la red
- Para determinar los pesos o costos de cada enlace, se recurre al uso de métricas, como por ejemplo:
  - Distancia
  - Retardo Promedio
  - Longitud de las filas
  - Tráfico Promedio
  - entre otros.



# Algoritmo de Dijkstra

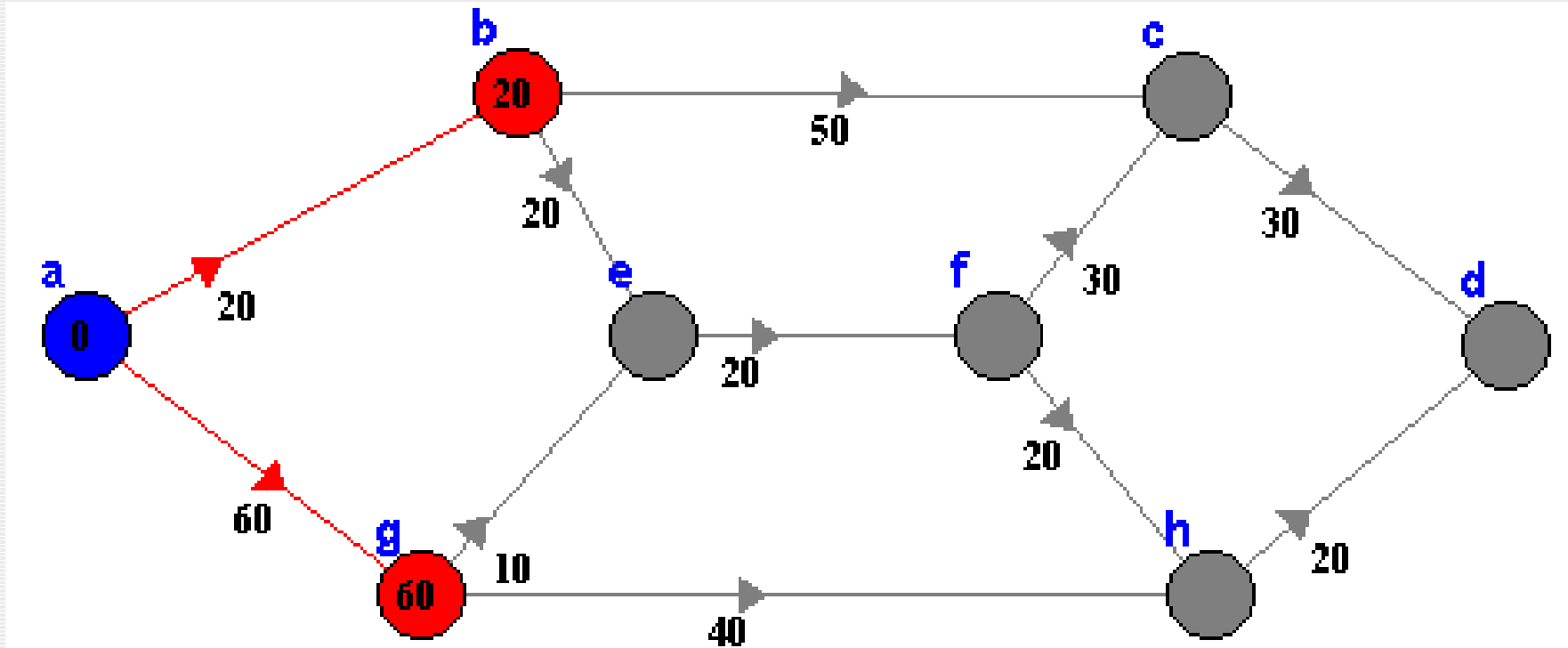
- Objetivo: Encontrar el camino más corto de un nodo a otro
- Ejemplo: nodo A ---> nodo D





# Algoritmo de Dijkstra

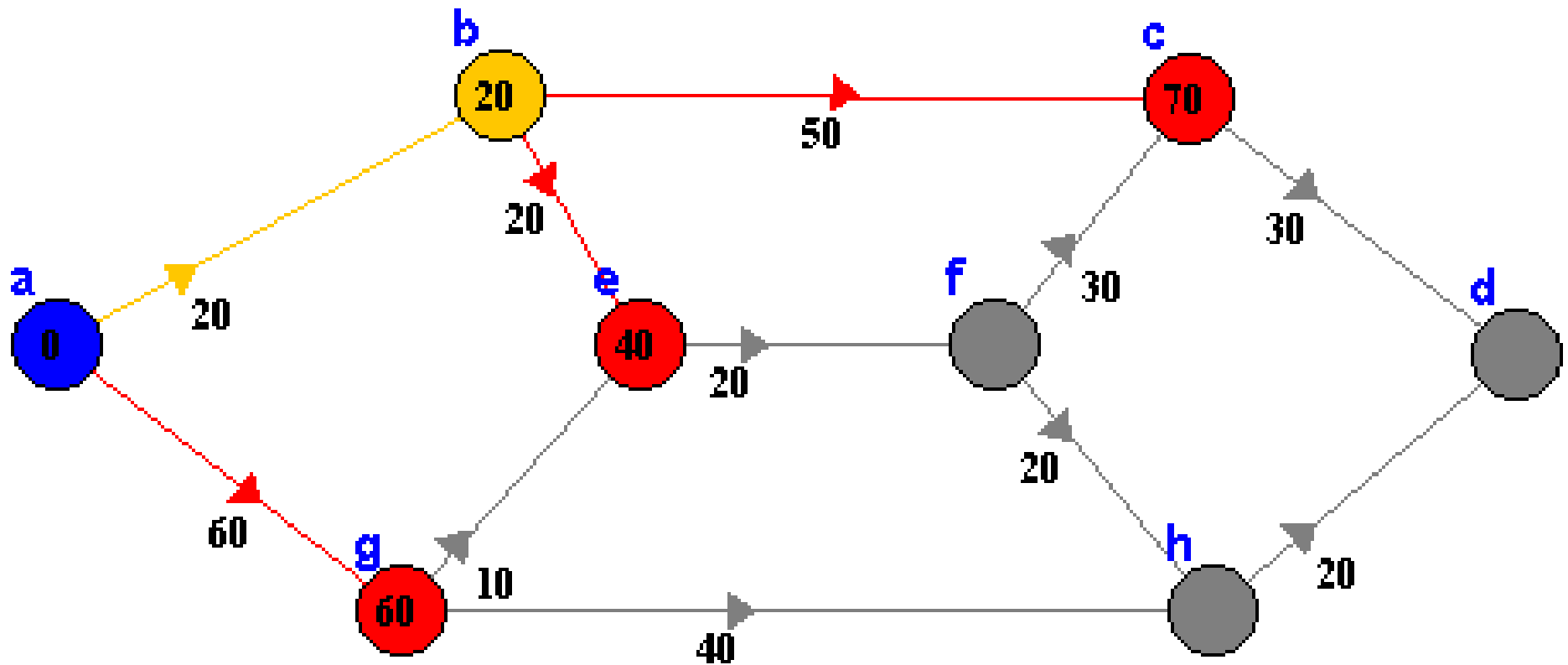
- Se analizan los nodos directamente conectados
- B es el de menor distancia
- B no posee otros nodos apuntando a él, por lo que se sigue con B.





# Algoritmo de Dijkstra

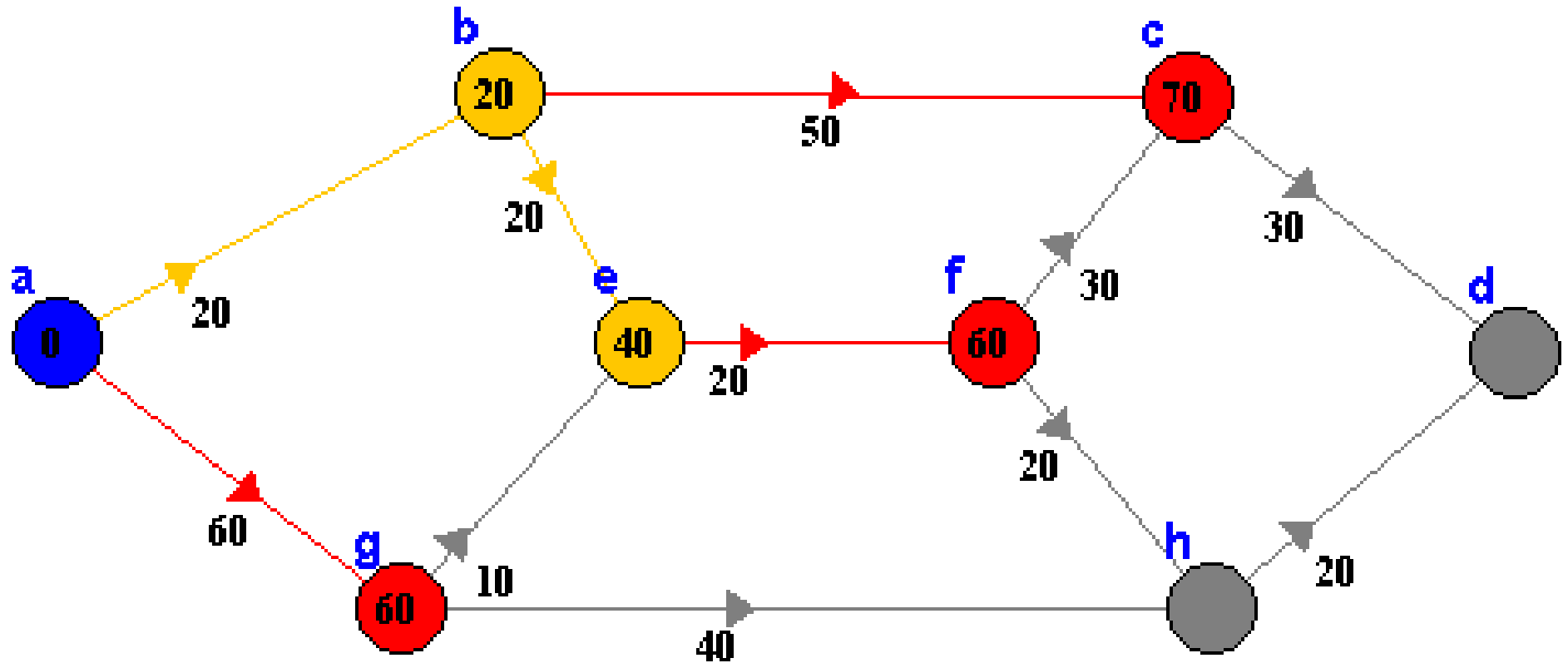
- Se repite la operación ahora desde el nodo B
- Se escoge al nodo E quien posee la menor distancia a A.





# Algoritmo de Dijkstra

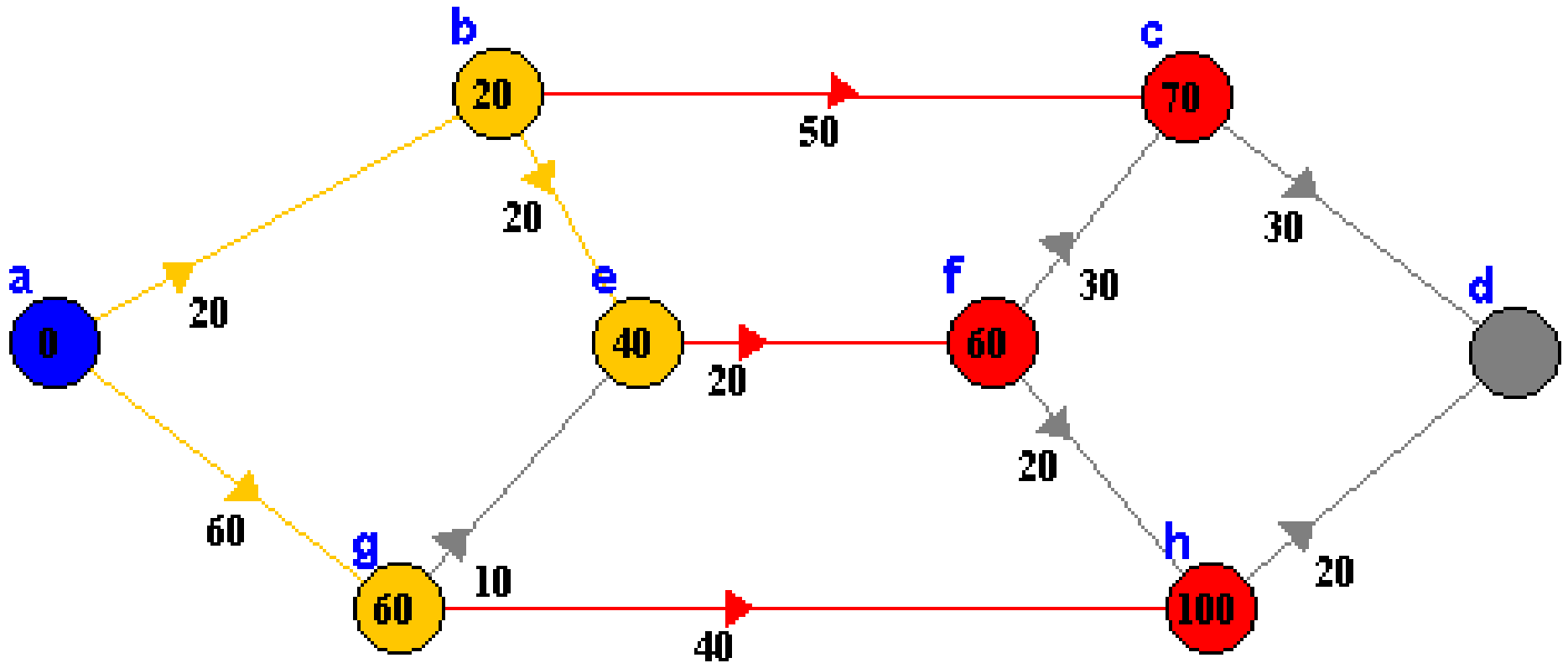
- G y F poseen la menor distancia, pero G con menos saltos
- Se escoge a G para continuar.





# Algoritmo de Dijkstra

Ahora F posee la menor distancia.

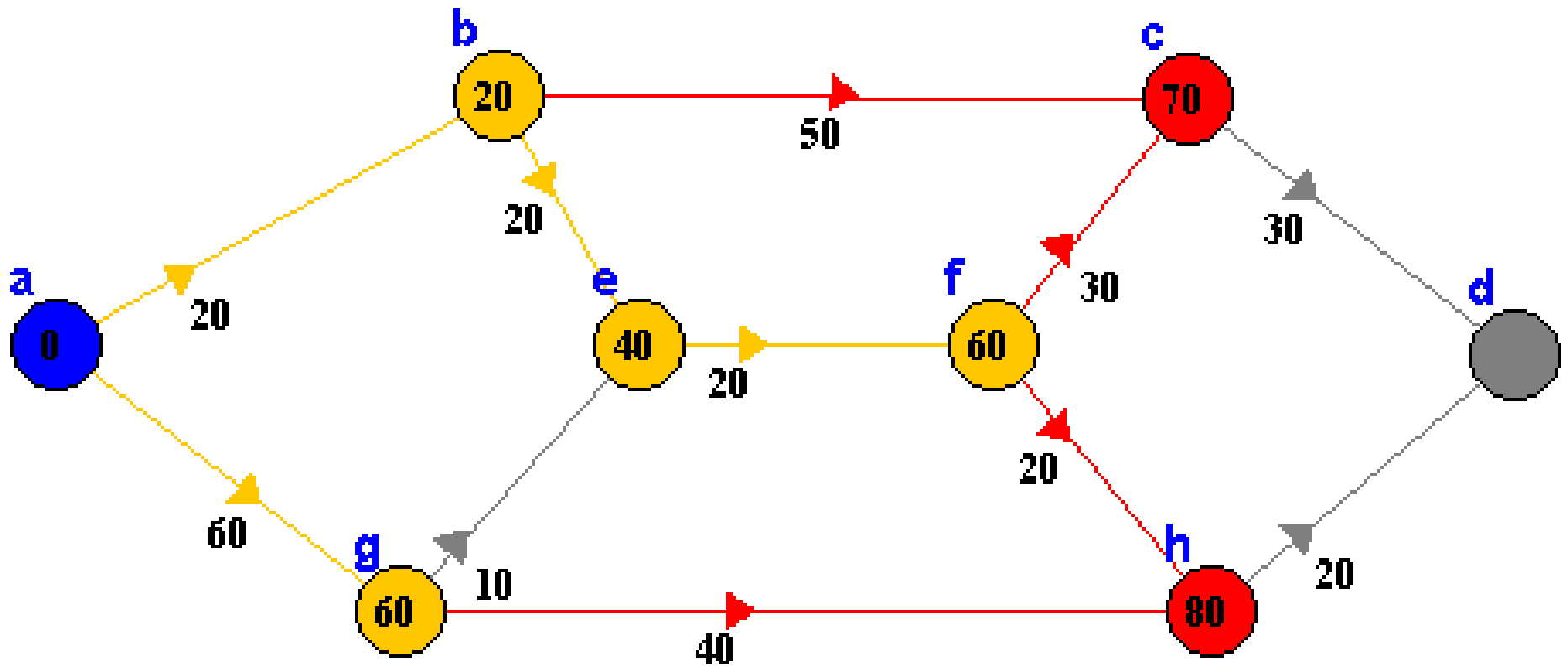






# Algoritmo de Dijkstra

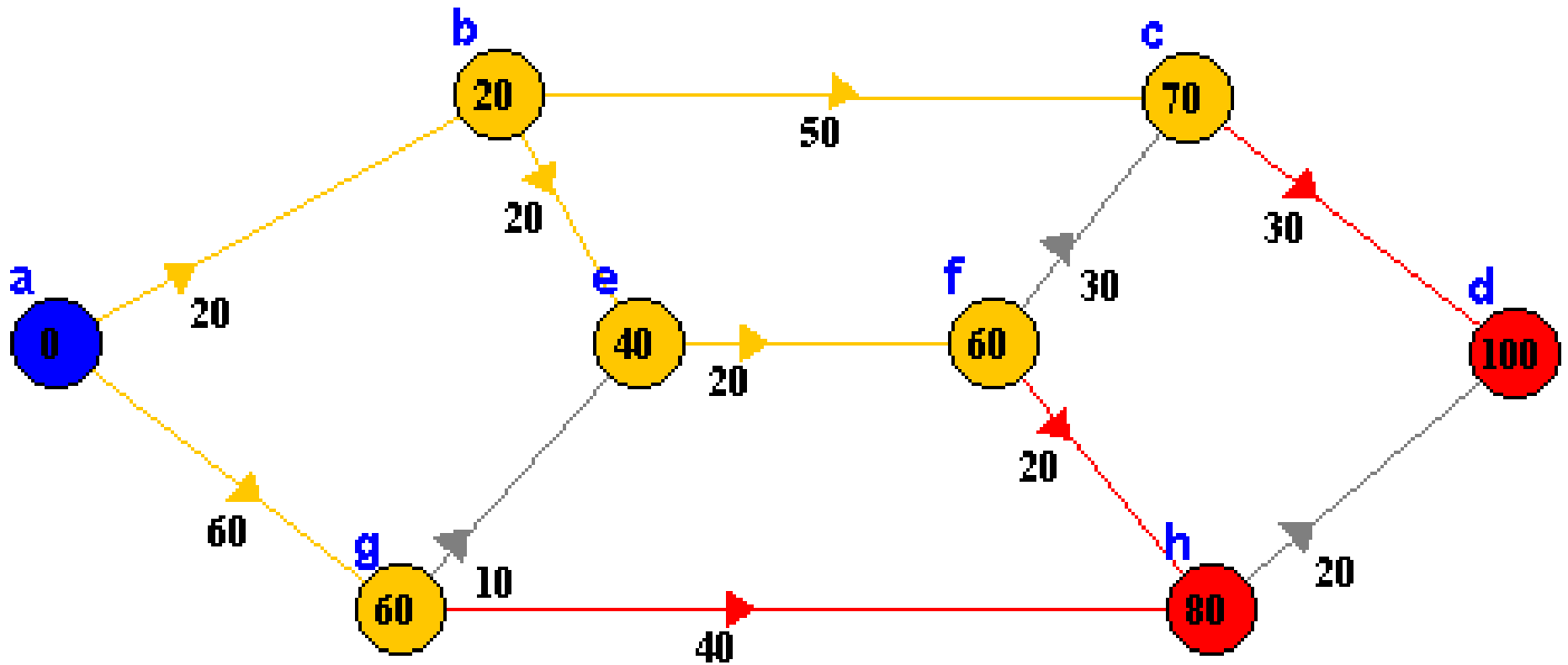
C con distancia 70 es el de menor distancia.





# Algoritmo de Dijkstra

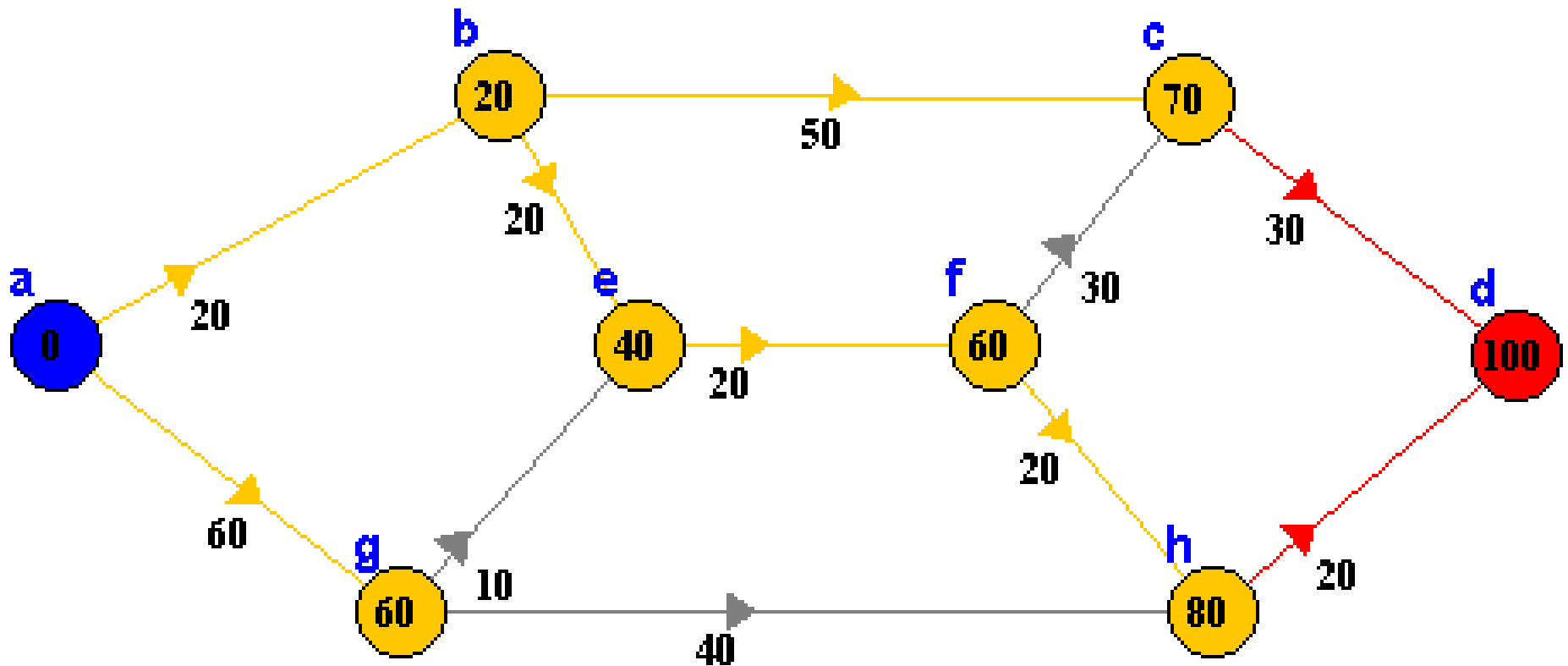
- H con distancia 80 posee menor distancia que D.





# Algoritmo de Dijkstra

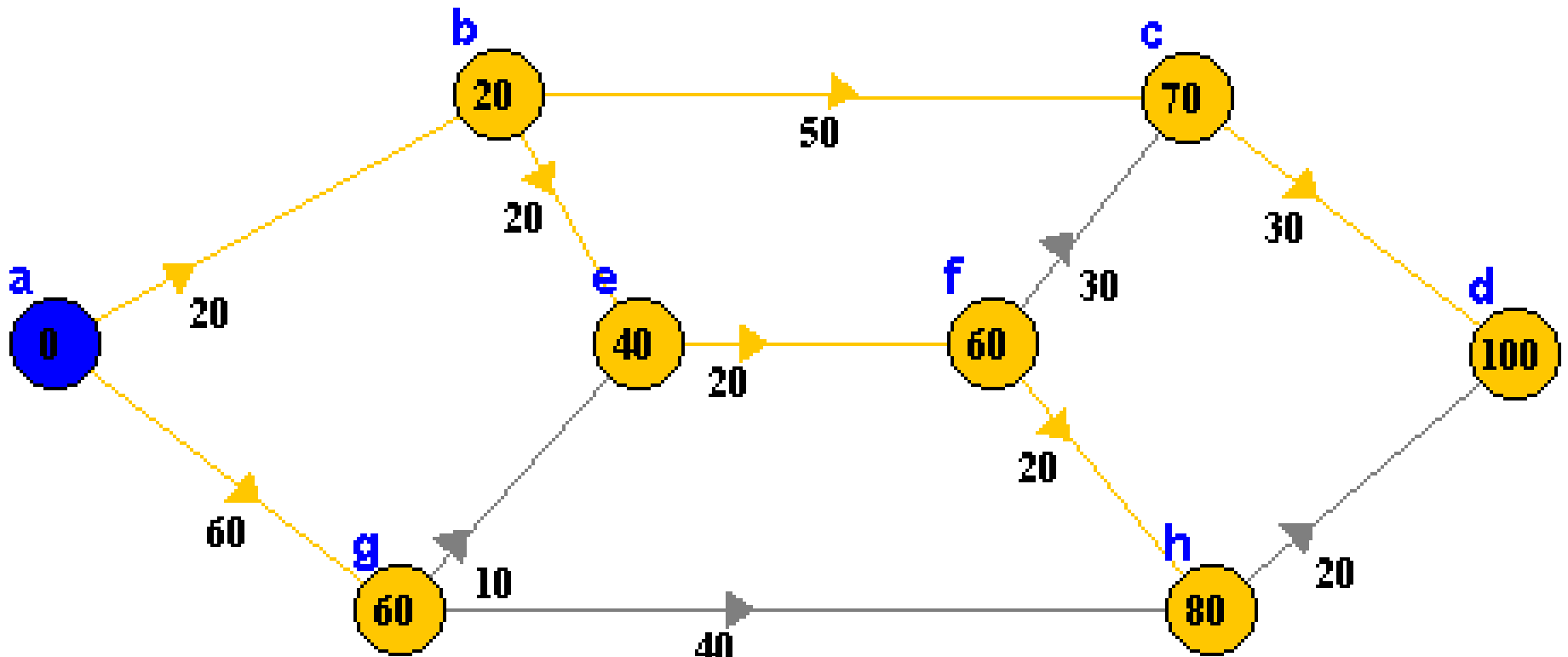
- D puede ser alcanzado con una distancia de 100 a través de C o de H.....cuál escoger?





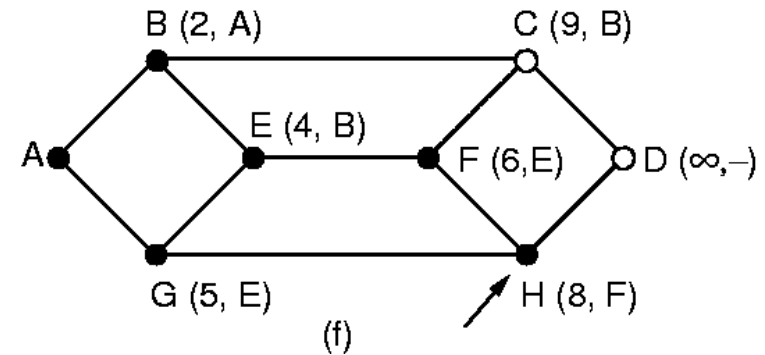
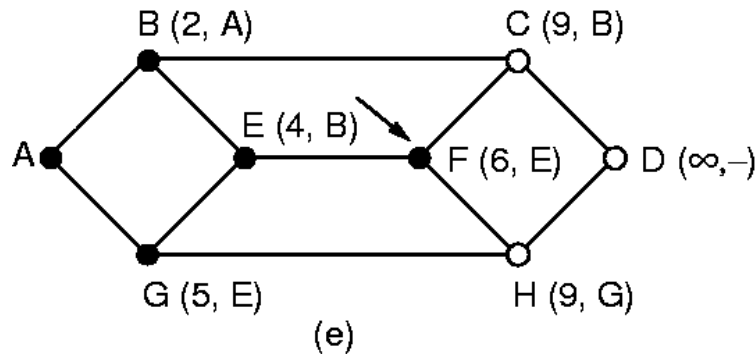
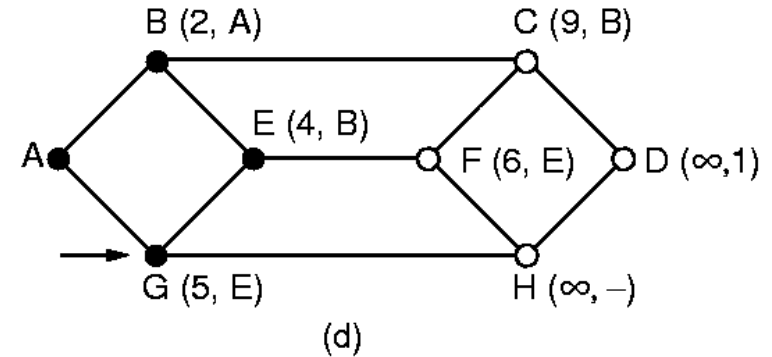
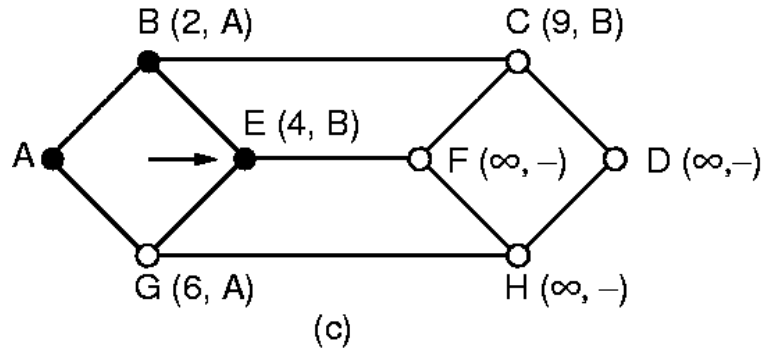
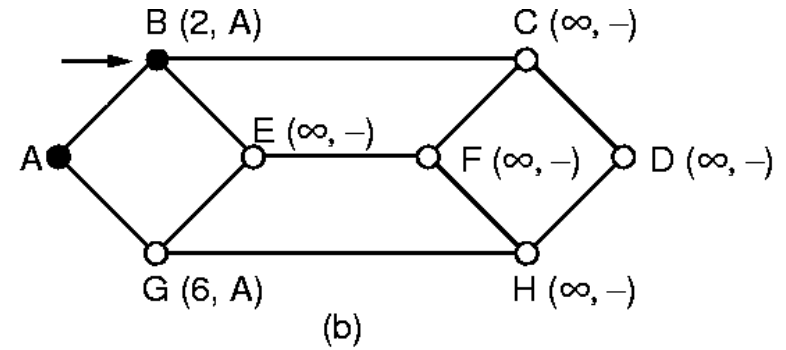
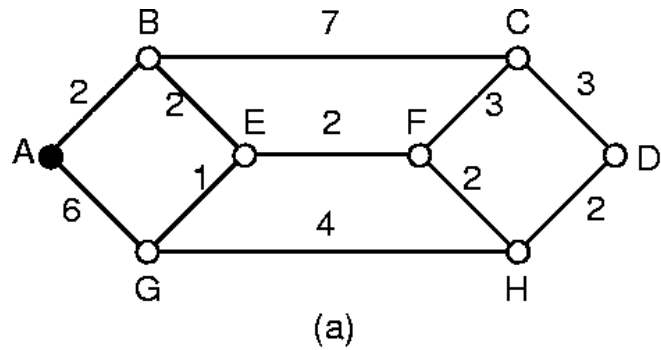
# Algoritmo de Dijkstra

- Se escoge C
- Ruta de menor distancia de A a D es: ABCD
- línea amarilla indica la ruta óptima desde A a cualquier nodo.





# Algoritmo de Dijkstra

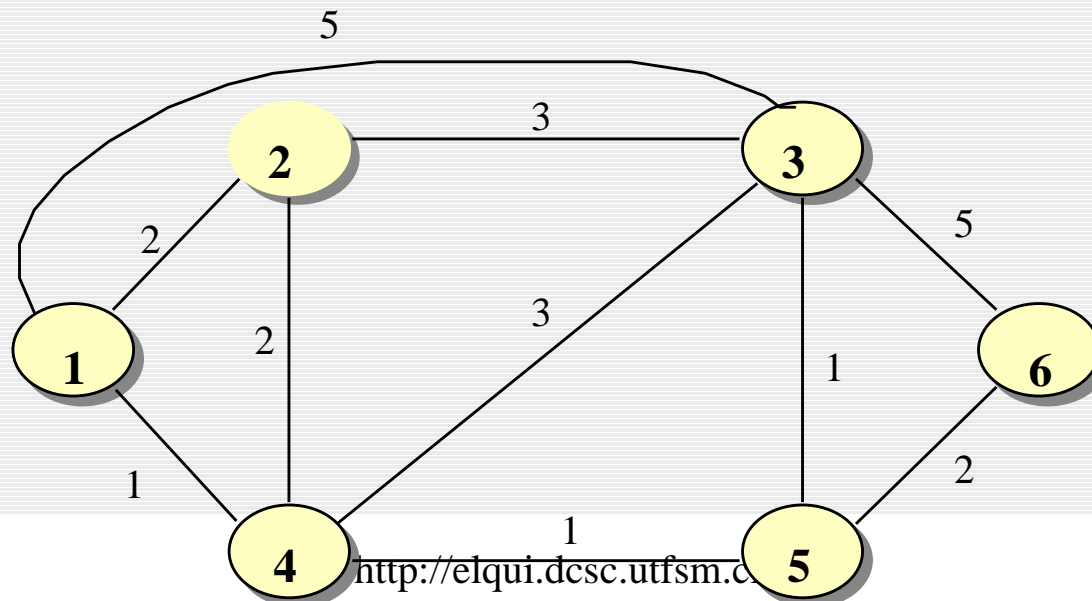




# Algoritmo de Dijkstra

## ALGORITMO DE DIJKSTRA, TOMANDO COMO FUENTE EL NODO 1

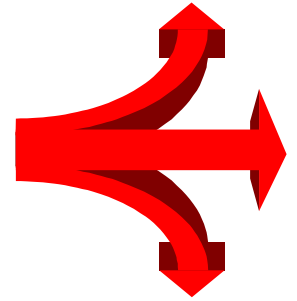
ITER.	M	D <sub>2</sub>	PATH	D <sub>3</sub>	PATH	D <sub>4</sub>	PATH	D <sub>5</sub>	PATH	D <sub>6</sub>	PATH
1	{1}	2	1-2	5	1-3	1	1-4	∞	-----	∞	-----
2	{1, 4}	2	1-2	4	1-4-3	1	1-4	2	1-4-5	∞	-----
3	{1, 2, 4}	2	1-2	4	1-4-3	1	1-4	2	1-4-5	∞	-----
4	{1, 2, 4, 5}	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6
5	{1, 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





# Algoritmos de Enrutamiento

- Ruteamiento del Camino más Corto
- **Ruteamiento Multitrayecto**
- Ruteamiento Centralizado
- Ruteamiento Aislado
- Inundación
- Ruteamiento basado en el Flujo
- Ruteamiento Distribuido (Vector Distancia)
- Ruteamiento por estado de enlace
- Ruteamiento Jerárquico
- Ruteamiento por Difusión



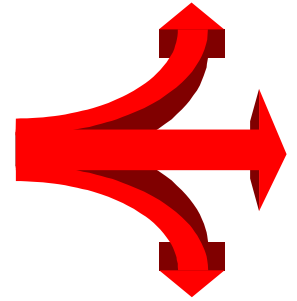
# Ruteamiento Multitrayecto

- El ruteamiento se realiza vía múltiples links de salida que unen a nodos distintos en la red
- Usado cuando existen varios trayectos entre un par Fuente-Destino
- Se aplica tanto en redes con servicio Datagrama, como en redes con servicio de Circuitos Virtuales
- Divide el tráfico entre los diversos trayectos que unen a dos nodos, mejorando el desempeño de la red.
- Se conoce también como Ruteamiento Bifurcado





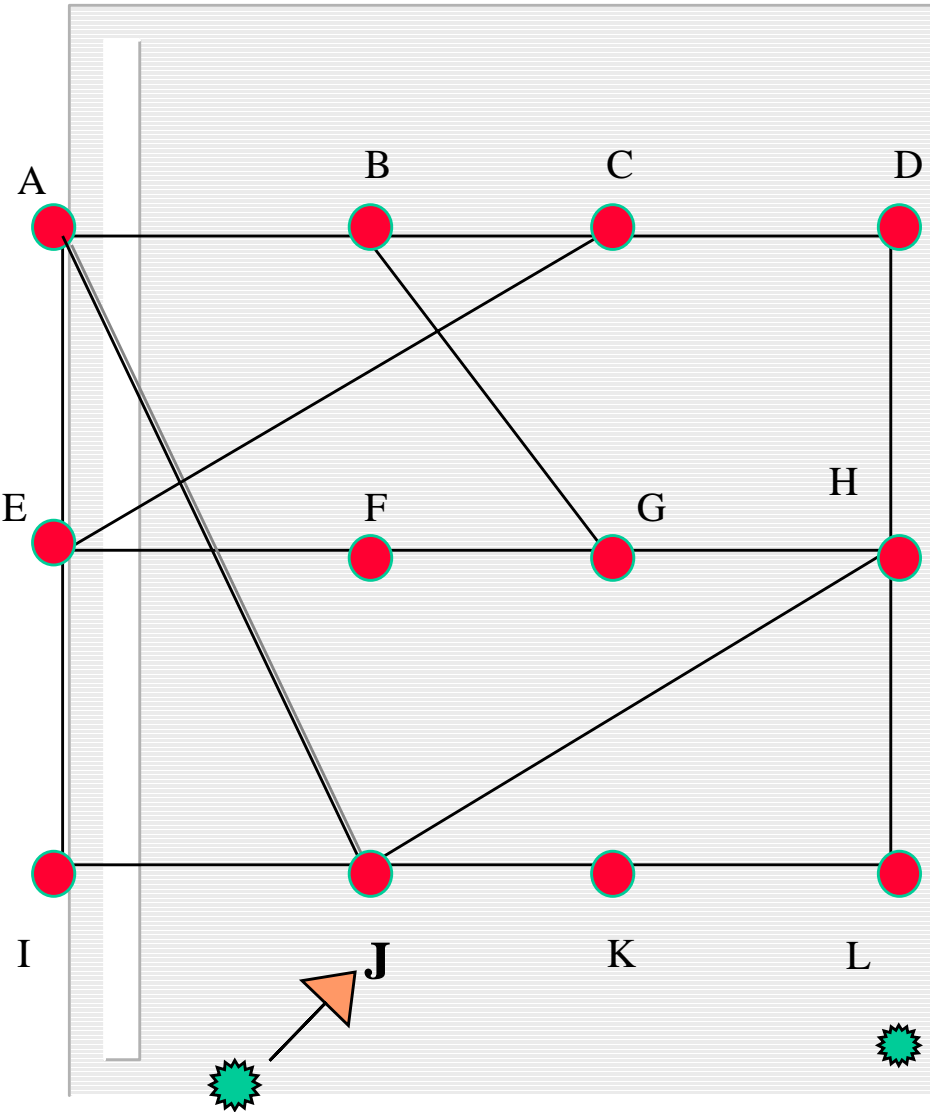
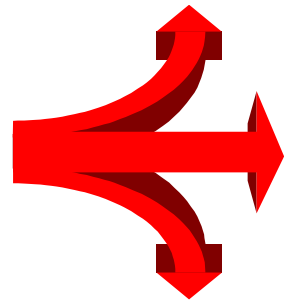
# Ruteamiento Multitrayecto



- Cada nodo mantiene una tabla con una entrada por cada posible destino
- Cada entrada en la tabla ofrece la 1<sup>a</sup>, 2<sup>a</sup>, 3<sup>a</sup>,... mejor ruta de salida para un destino en particular, además, junto a cada ruta se adjunta un peso que es usado como probabilidad de elección de la respectiva ruta
- Antes de enviar un paquete el nodo genera un número aleatorio (0.00 - 0.99), a continuación escoge una de las rutas alternativas usando los pesos de la tabla



# Ruteamiento Multitrayecto



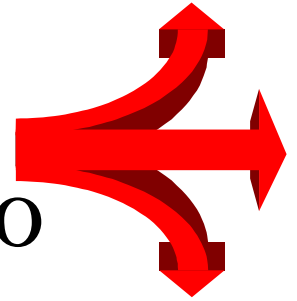
NODO DESTINO

	1 <sup>a</sup> OPCIÓN		2 <sup>a</sup> OPCIÓN		3 <sup>a</sup> OPCIÓN	
A	A	0.63	I	0.21	H	0.16
B	A	0.46	H	0.31	I	0.23
C	A	0.34	I	0.33	H	0.33
D	H	0.50	A	0.25	I	0.25
E	A	0.40	I	0.40	H	0.20
F	A	0.34	H	0.33	I	0.33
G	H	0.46	A	0.31	K	0.23
H	H	0.63	K	0.21	A	0.16
I	I	0.65	A	0.22	H	0.13
-						
K	K	0.67	H	0.22	A	0.11
L	K	0.42	H	0.42	A	0.16

**TABLA de RUTEAMIENTO para el NODO J**



# Ruteamiento Multitrayecto



## ■ Ventajas

- Posibilidad de Tx diferentes clases de tráfico sobre diferentes caminos
- Mejora el Rendimiento
- Da mayor fiabilidad
  - Al poseer  $n$  rutas independientes entre cada par de nodos, la red es capaz de “soportar” la pérdida de  $n-1$  enlaces, sin que sea dividida en dos partes.



# Algoritmos de Enrutamiento

- Ruteamiento del Camino más Corto
- Ruteamiento Multitrayecto
- **Ruteamiento Centralizado**
- Ruteamiento Aislado
- Inundación
- Ruteamiento basado en el Flujo
- Ruteamiento Distribuido (Vector Distancia)
- Ruteamiento por estado de enlace
- Ruteamiento Jerárquico
- Ruteamiento por Difusión



# Ruteamiento Centralizado

- Las decisiones de Ruteamiento las toma una entidad central llamada Centro de Control de Ruteamiento (RCC).
- Cada nodo Tx información de su estado actual al RCC. Esta información puede ser :
  - Lista de sus vecinos activos
  - Longitud actual de las colas de espera
  - Tráfico procesado
  - Entre otros
- El RCC, utilizando algún algoritmo de ruteamiento, determina las nuevas tablas de ruteamiento para todos los nodos de la red



# Ruteamiento Centralizado

## Ventajas

- Al conocer la información completa de la Red, el RCC puede tomar decisiones de ruteamiento óptimas
- Libera a los nodos del trabajo de calcular las tablas de ruteamiento

## Desventajas

- Aumento considerable del tráfico, sobretodo si los cambios de topología y/o de tráfico son frecuentes
- La red queda muy vulnerable a los problemas que puedan suceder en el RCC
- Alta concentración de tráfico en los enlaces cercanos al RCC
- Los nodos cercanos conmutarán primero a las nuevas rutas, antes que los localizados a mayor distancia, provocando desfases entre los nodos.



# Algoritmos de Enrutamiento

- Ruteamiento del Camino más Corto
- Ruteamiento Multitrayecto
- Ruteamiento Centralizado
- **Ruteamiento Aislado**
- Inundación
- Ruteamiento basado en el Flujo
- Ruteamiento Distribuido (Vector Distancia)
- Ruteamiento por estado de enlace
- Ruteamiento Jerárquico
- Ruteamiento por Difusión



# Ruteamiento Aislado

- Las decisiones de ruteamiento son tomadas en cada nodo, basándose en la información disponible localmente
- No se intercambia información de ruteamiento con otros nodos (vecinos)
- Algoritmo de la “Papa Caliente” (Baran, 1964)
- Algoritmo del Aprendizaje hacia Atrás

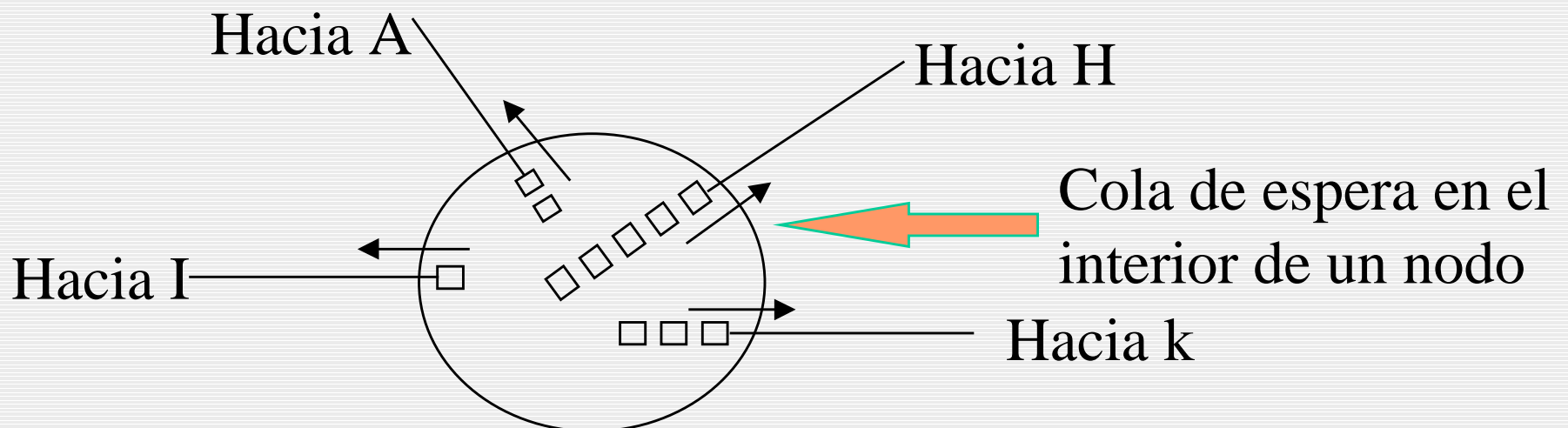




# Ruteamiento Aislado

## Algoritmo de la “Papa Caliente” (Baran, 1964)

- Cuando llega un nuevo paquete a un nodo, éste trata de deshacerse lo más rápido posible de él, retransmitiéndolo por la línea que posea la cola de espera más corta, sin importar que la línea no se dirija hacia el destino





# Ruteamiento Aislado

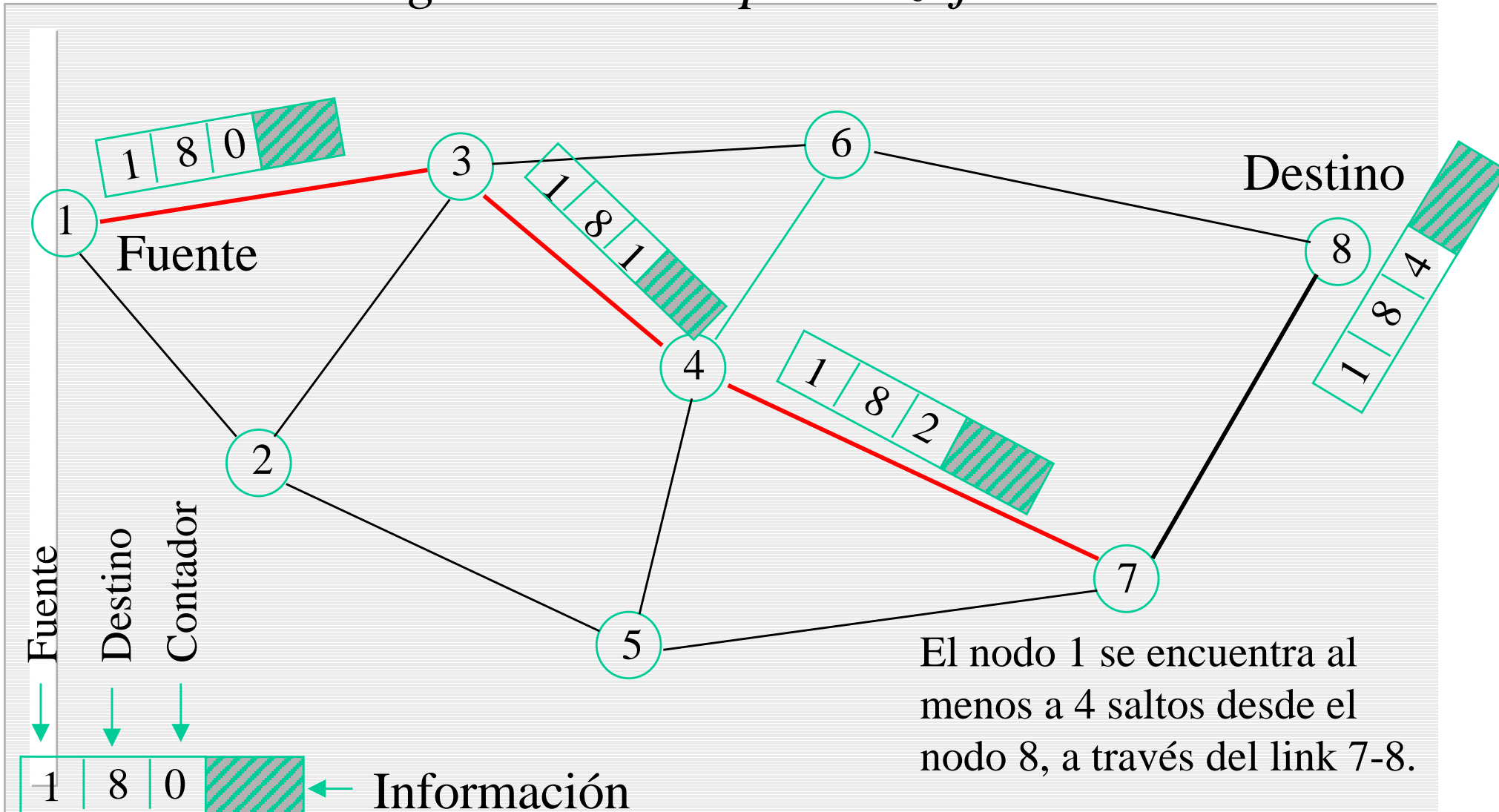
## Algoritmo del Aprendizaje hacia Atrás

- Se incluye, en cada paquete, la identificación del nodo fuente, además de un contador que se incrementa después de pasar por cada salto
- Si un nodo recibe un paquete procedente del nodo  $X$ , por la línea  $k$ , con su contador marcando  $Y$  saltos....
  - Determina que  $X$  no puede estar más lejos que  $Y$  saltos.
  - Si su mejor ruta a  $X$  es mayor que  $Y$  saltos, concluye que  $k$  es su mejor ruta hacia  $X$



# Ruteamiento Aislado

## *Algoritmo del Aprendizaje hacia Atrás*





# Algoritmos de Enrutamiento

- Ruteamiento del Camino más Corto
- Ruteamiento Multitrayecto
- Ruteamiento Centralizado
- Ruteamiento Aislado
- **Inundación**
- Ruteamiento basado en el Flujo
- Ruteamiento Distribuido (Vector Distancia)
- Ruteamiento por estado de enlace
- Ruteamiento Jerárquico
- Ruteamiento por Difusión



# Inundación

- Básicamente el método consiste en:
  - Un Nodo fuente envía un paquete a cada uno de sus vecinos
  - En cada nodo vecino, el paquete que llega es retransmitido usando todos los enlaces de salida, exceptuando aquél por el cual entró.
- En su forma natural este algoritmo no es práctico
- Se deben hacer modificaciones que permitan terminar con la inundación y disminuir la cantidad de réplicas.

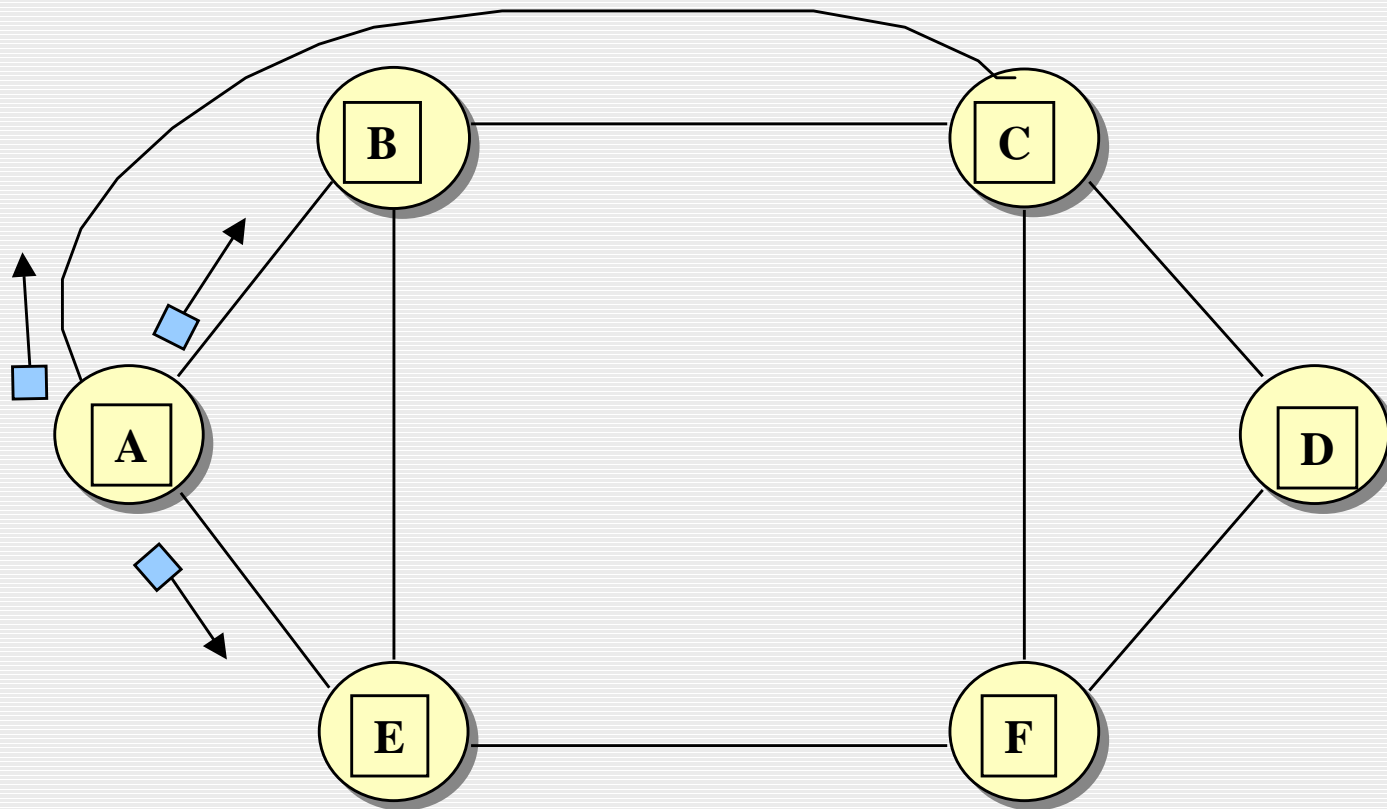


# Inundación

- Es la Técnica de ruteamiento más simple
- No requiere información de la red
- Es inherentemente robusta
- Es ineficiente en el uso de los recursos de la red
- Es robusto ante fallas en los enlaces y en los nodos.



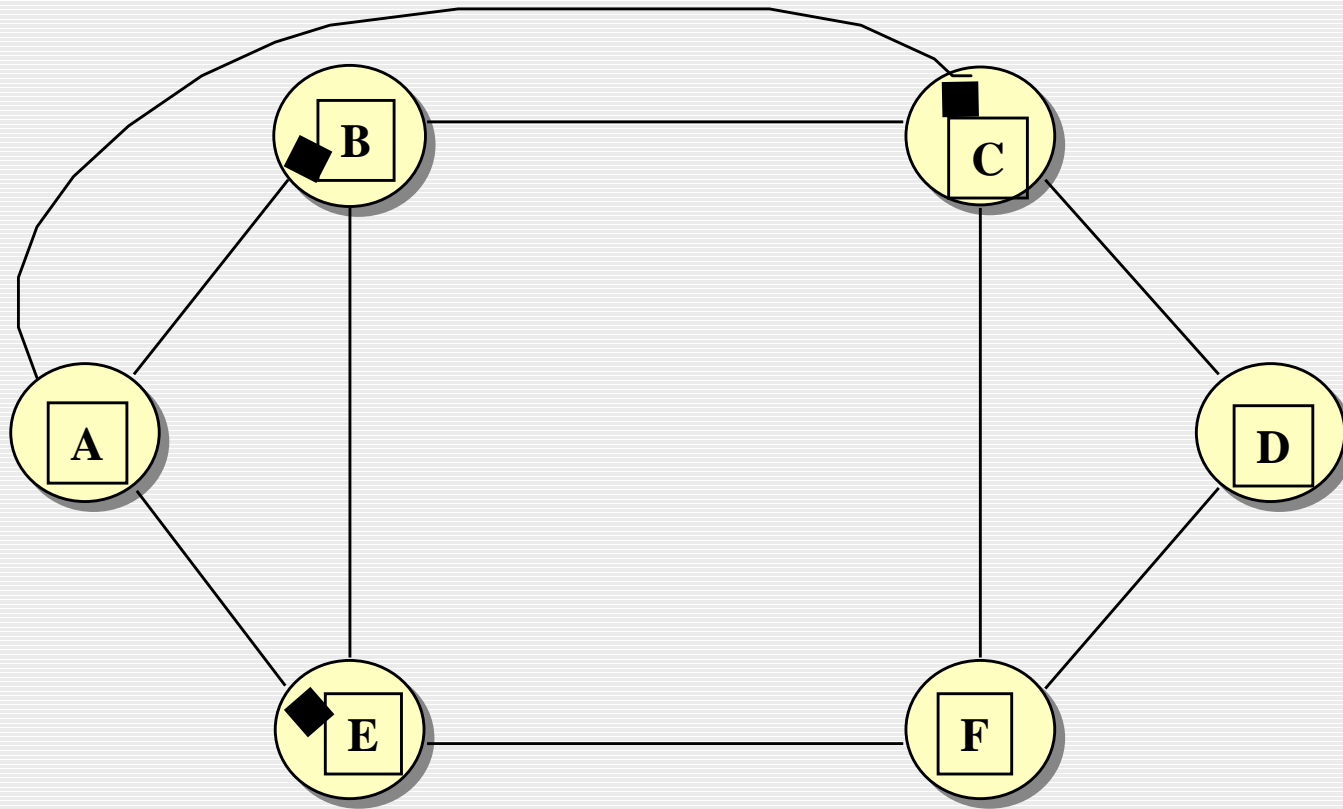
# Inundación



**Primer Salto**



# Inundación

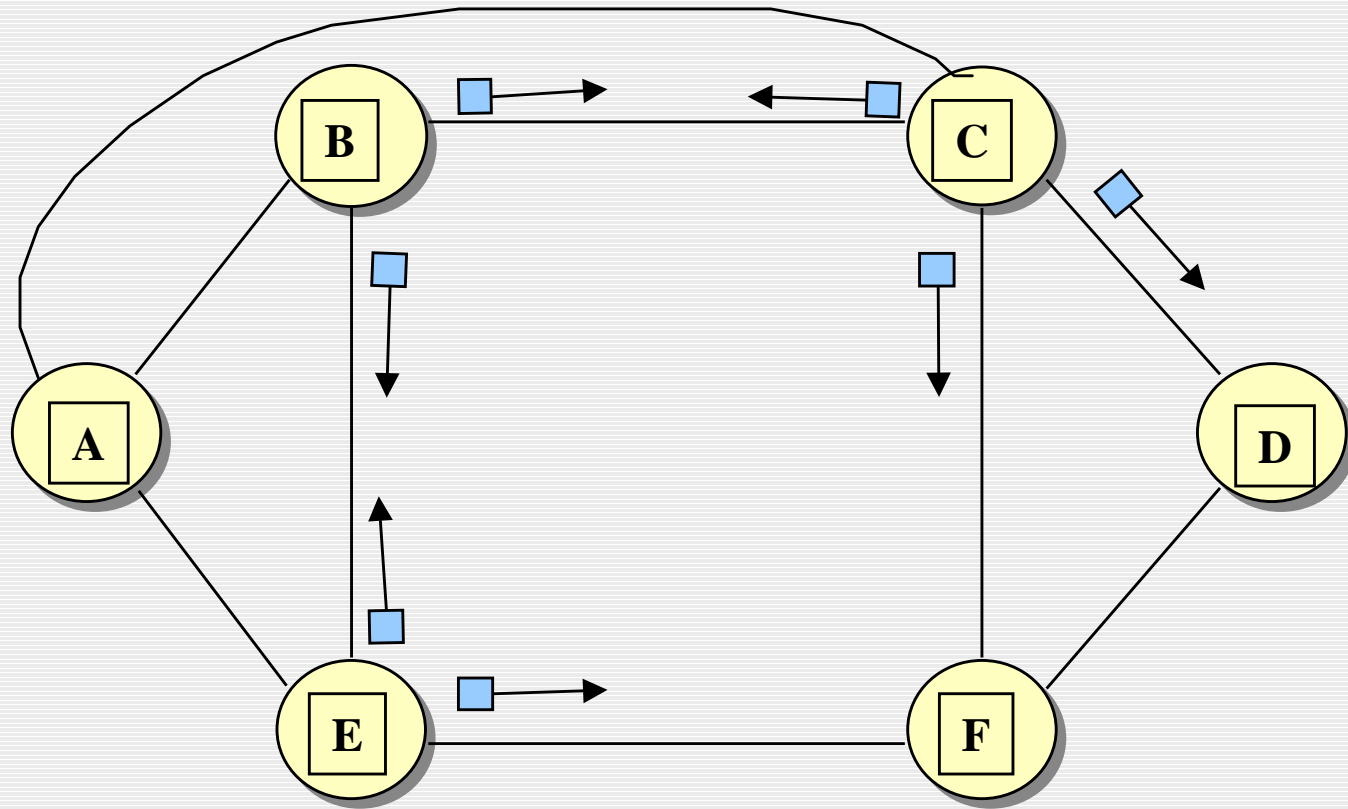


Los paquetes llegan a los nodos vecinos





# Inundación

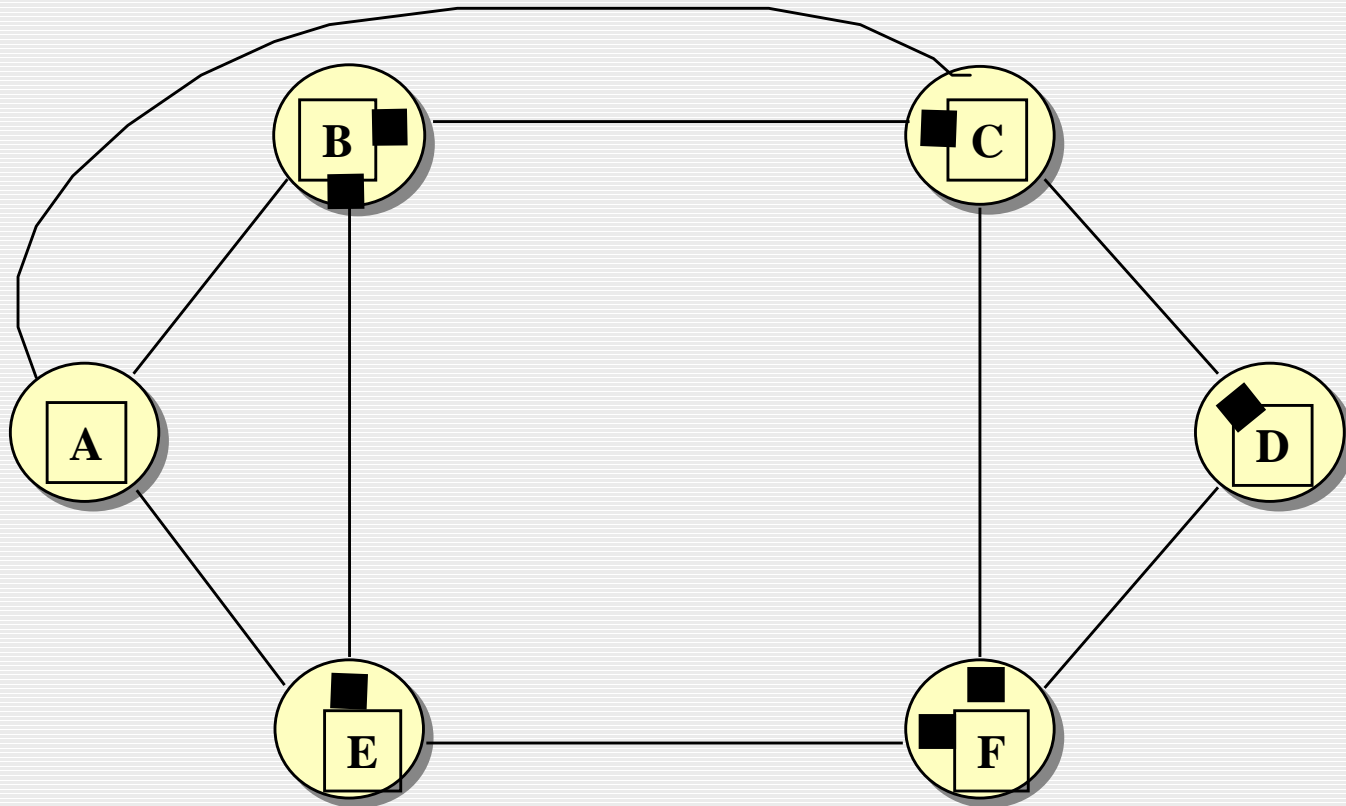


**Segundo Salto**

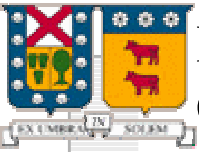
Se generan réplicas en cada nodo y son retransmitidas a todos sus vecinos



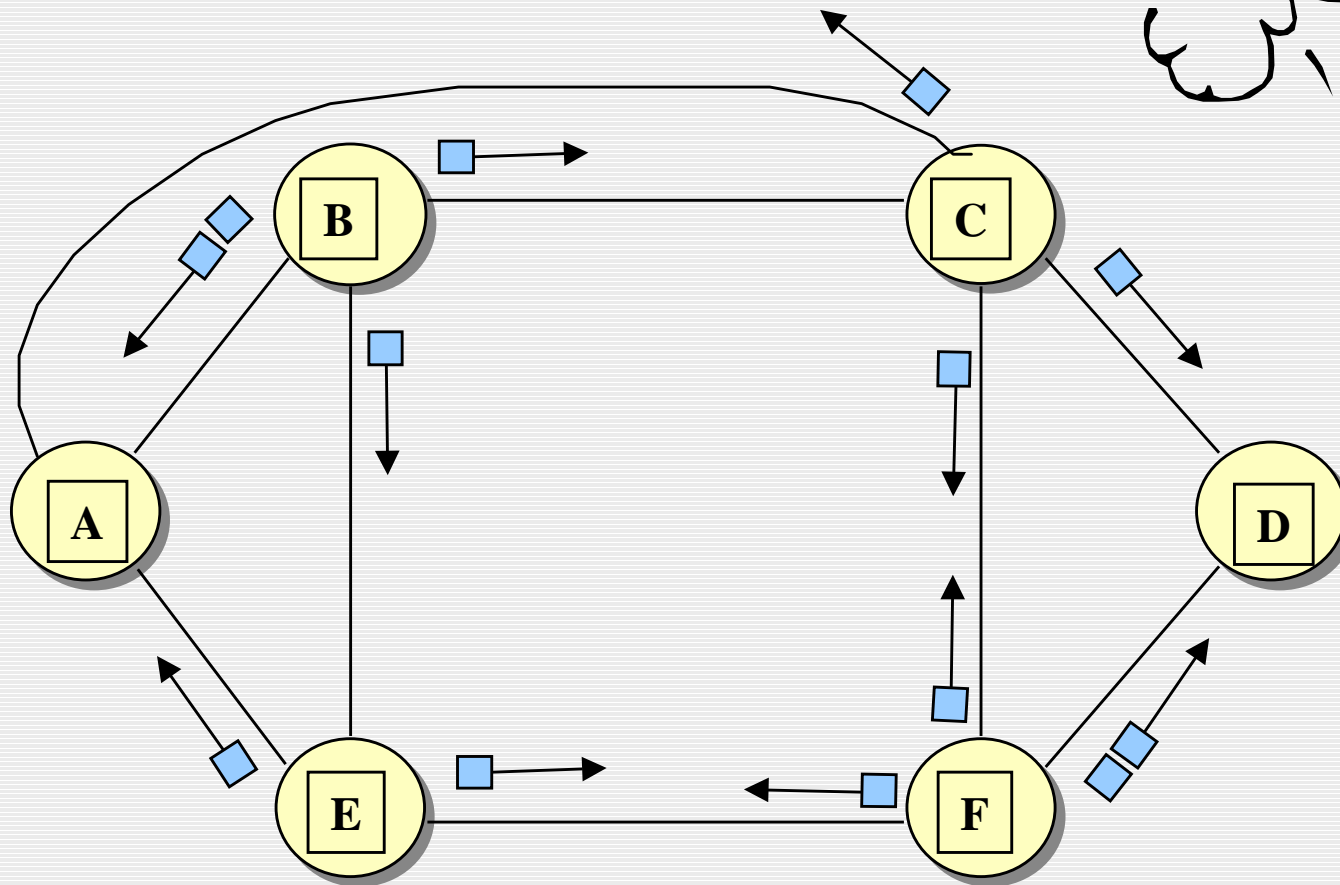
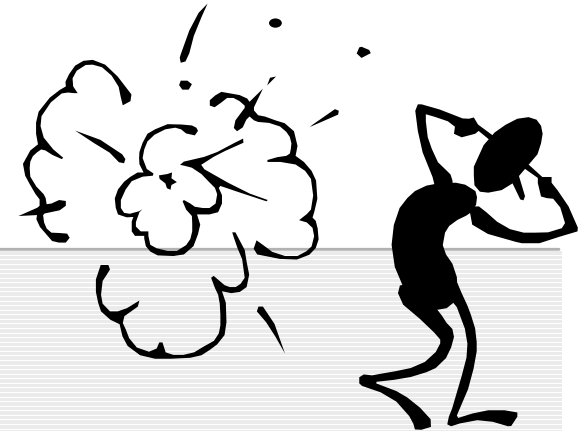
# Inundación



Las réplicas llegan a los nodos vecinos donde serán nuevamente duplicadas y retransmitidas



# Inundación



## Tercer Salto

La situación se vuelve incontrolable. Es necesario establecer un mecanismo de control que permita frenar la inundación de paquetes.



## Inundación (Método Práctico)

- Dado que el número de paquetes duplicados crece rápidamente (exponencialmente), es necesario establecer un mecanismo que controle el número de réplicas.
- Una forma de frenar el “overflow”, es agregar a cada paquete un contador de saltos
- Generalmente el contador es inicializado con el “diámetro” de la red
- Cada vez que un paquete pasa por un nodo su contador es decrementado en una unidad
- Cuando el contador llega a cero el paquete es desechado por el router