

# Capítulo 4

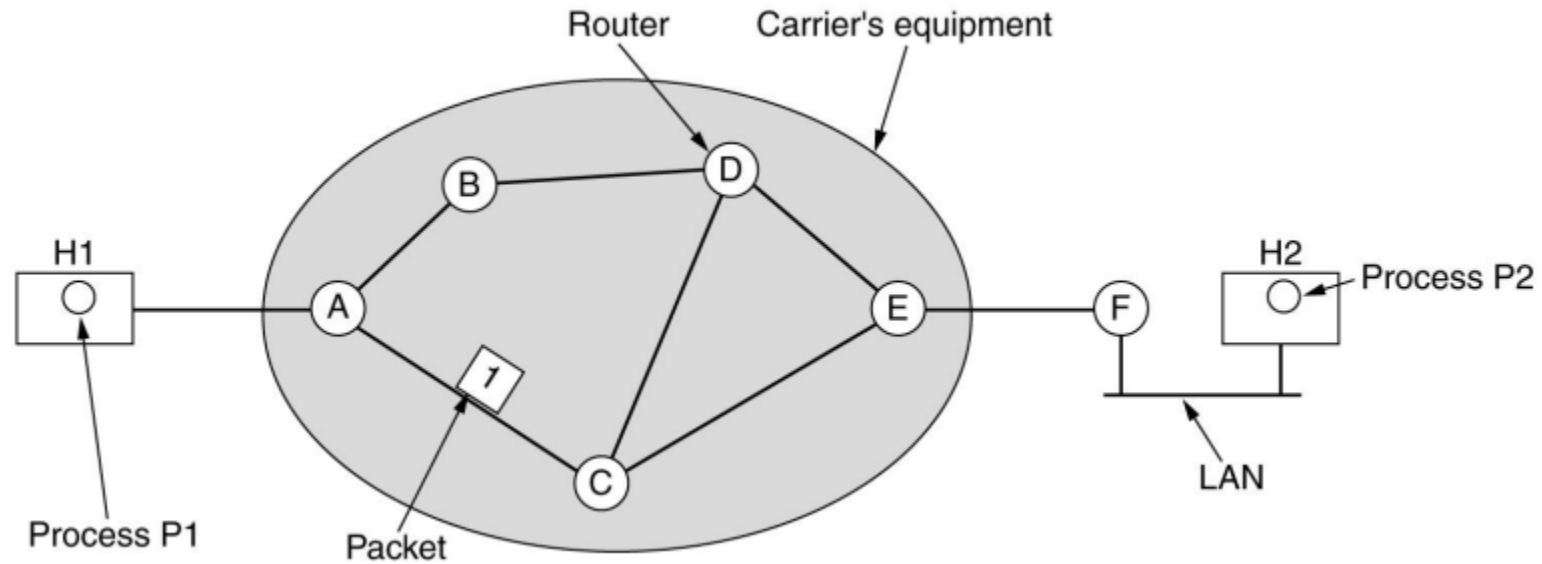
## Capa de red Parte 1

Application
Transport
Network
Link
Physical

# Capa de red

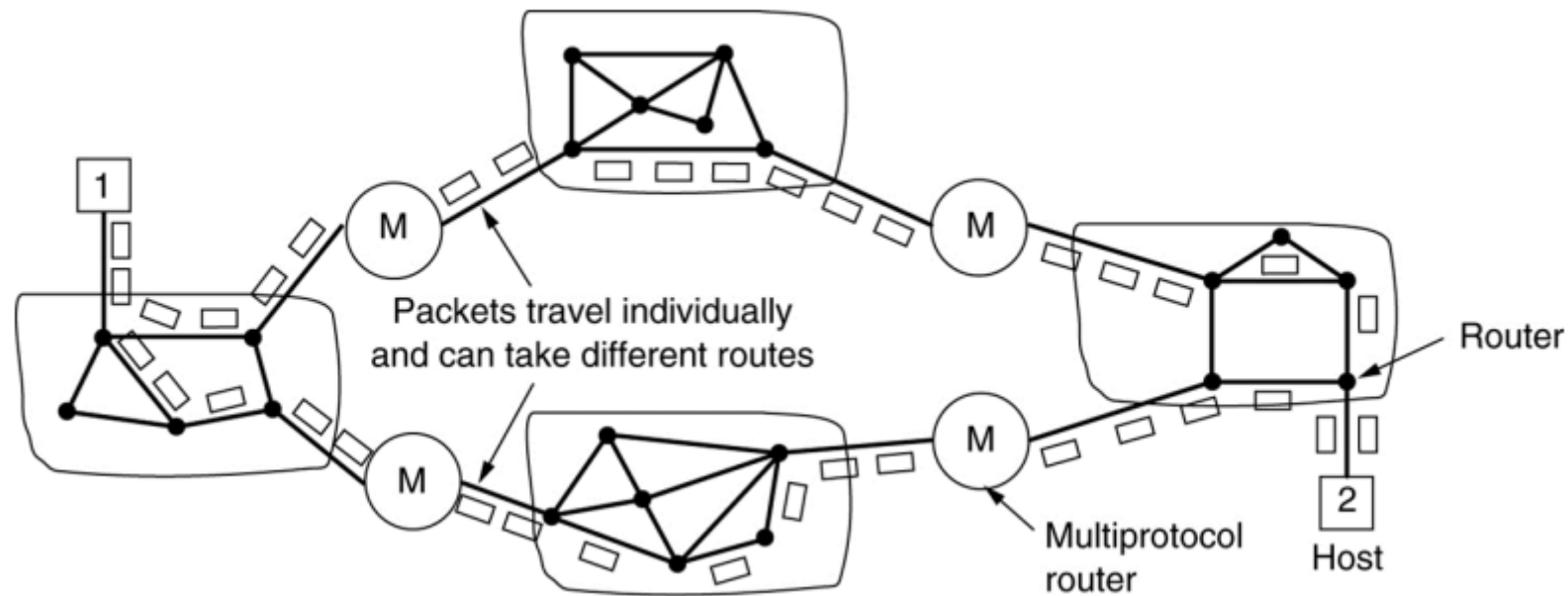
- **Repaso:**
  - ¿Cuál es el propósito de la capa de red?
  - ¿De qué asuntos se ocupa la capa de red?

# Capa de red: hardware subyacente



subred

# Capa de red: hardware subyacente



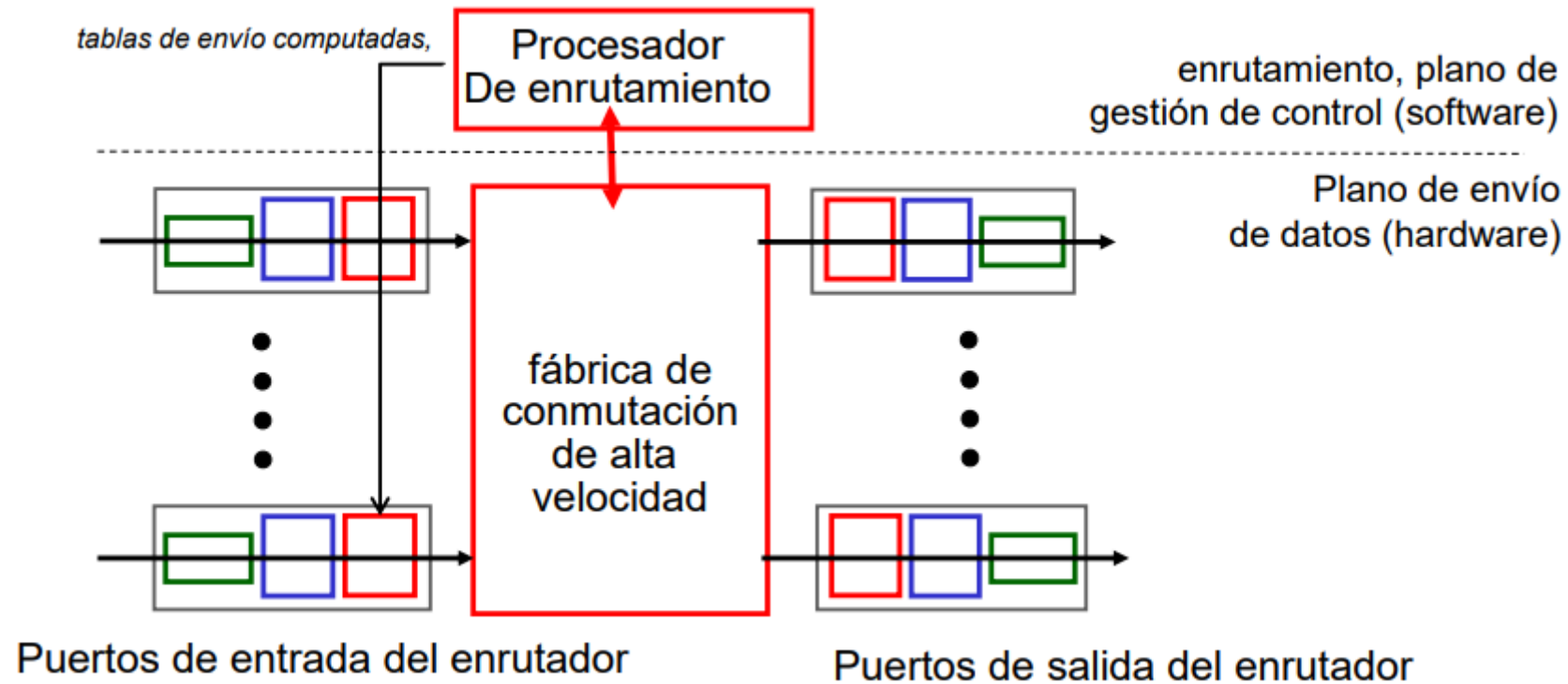
Interred

# Capa de red: Enrutadores

- **Repaso:**

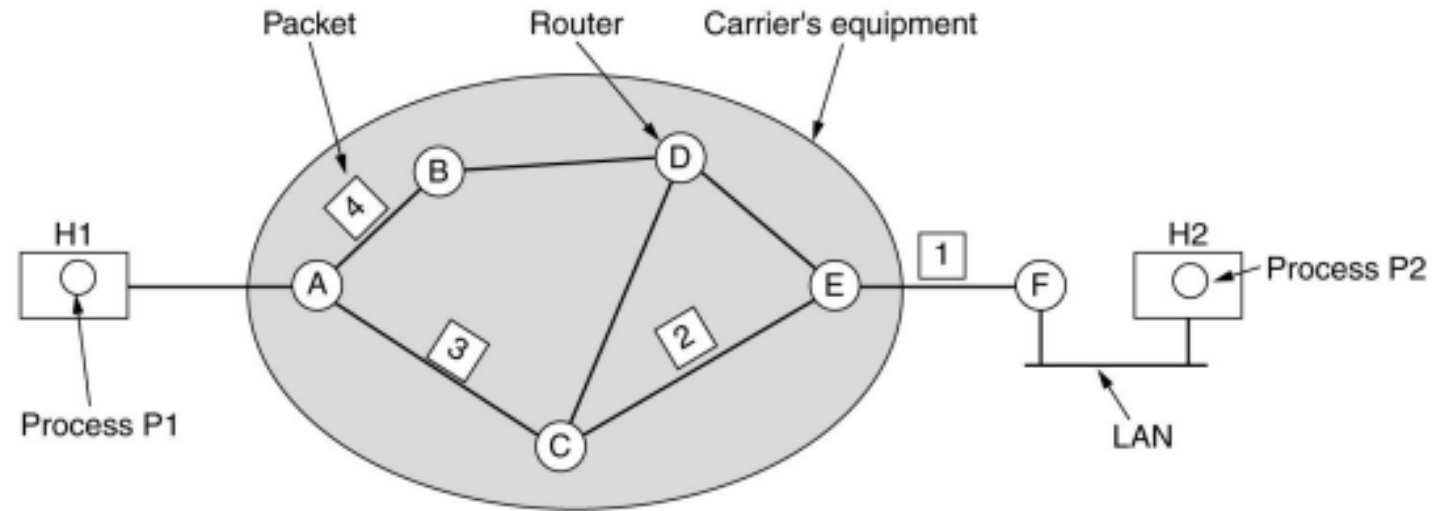
- ¿De qué tareas se encarga un enrutador?

# Capa de red: arquitectura de un enrutador



# Capa de red: servicio no orientado a la conexión

- Enrutadores de origen y de destino
- **Mapeo** de host de destino a enrutador de destino
- Espacios de direcciones para las máquinas de las redes locales.



A's table			
initially	later	C's table	E's table
A   -	A   -	A   A	A   C
B   B	B   B	B   A	B   D
C   C	C   C	C   -	C   C
D   B	D   B	D   D	D   D
E   C	E   B	E   E	E   -
F   C	F   B	F   E	F   F
<div>Dest. Line</div>			

¿Qué se hace cuando llega un paquete a un enrutador?

# Capa de red

- **Tareas**

- Leer servicio orientado a la conexión:
  - no lo vamos a estudiar en clase ni usar en la materia, pero es bueno conocerlo.
- Leer tabla comparativa en las filmillas de servicios orientado a la conexión y no orientado a la conexión.



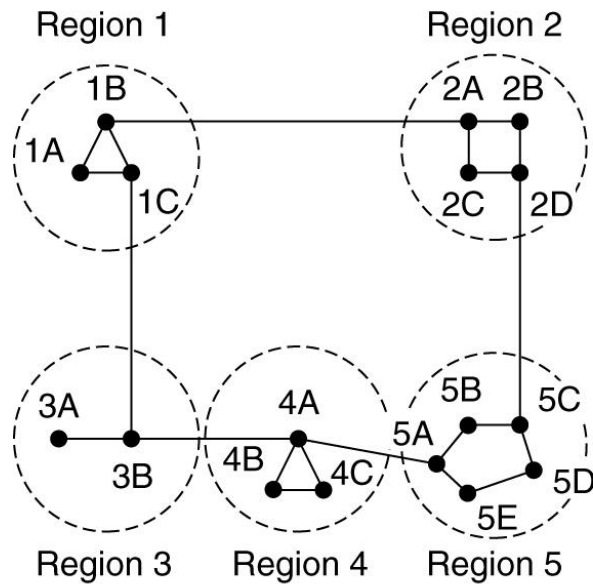
# Capa de red

- ¿Qué pasa a medida que crece mucho el tamaño de una subred? ¿Qué problemas surgen?
- **Ayuda:** Examinar costo para los enrutadores, costo de actualizar tabla de reenvío, etc.

# Capa de red: Enrutamiento jerárquico

- **Solución:** Usar **enrutamiento jerárquico**.
  - Dividir la red en regiones
  - Un enrutador sabe de la topología de la región donde se encuentra.
  - Un enrutador sabe de la existencia de otras regiones y cómo ir hacia ellas, pero no sabe la topología interna de una región.
- **¿Cómo sería la tabla de un enrutador cuando se hace enrutamiento jerárquico?** Usar la idea de la solución.

# Capa de red: Enrutamiento jerárquico



(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

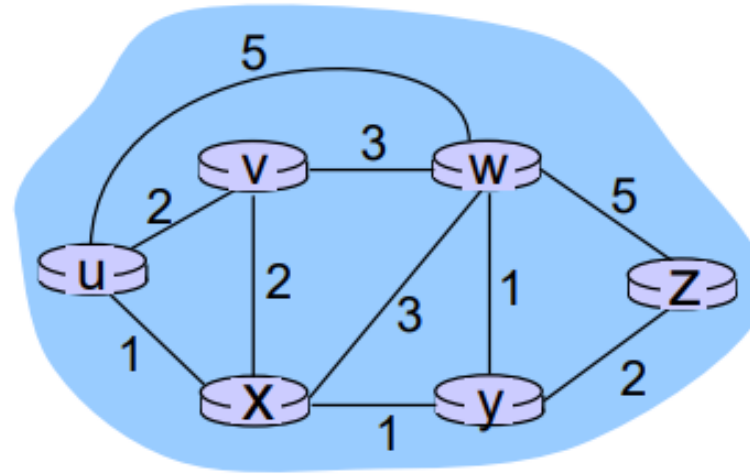
Dest.	Line	Hops
1A	—	—
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

(c)

# Capa de red: Grafo de una red

- **Sugerir cómo definir el grafo de una red.**
  - ¿Qué representan los nodos?
  - ¿Qué representan los lados?

# Capa de red: Grafo de una red



**Grafo:**  $G = (N, E)$

$N$  = conjunto de enrutadores =  $\{ u, v, w, x, y, z \}$

$E$  = conjunto de enlaces =  $\{ (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) \}$

Los arcos tienen **etiquetas** para el **costo** de atravesarlos

costo del camino  $(x_1, x_2, x_3, \dots, x_p) = c(x_1, x_2) + c(x_2, x_3) + \dots + c(x_{p-1}, x_p)$

# Capa de red: algoritmos de enrutamiento

- **Repaso:**

- ¿Para qué sirven los algoritmos de enrutamiento?
- ¿Idealmente qué debería lograr un algoritmo de enrutamiento?
- ¿Qué es lo que hace un algoritmo de enrutamiento al ejecutarse en un enrutador?

# Capa de red: enrutamiento de caminos más cortos

- **Suposiciones:**

- Se ingresa en cada enrutador el grafo de la red.
- Esto se hace una sola vez o cada tanto.
- En base a ese grafo se caminan los caminos más cortos.

- **Cosas que no hacen estos algoritmos**

# Capa de red: enrutamiento de caminos más cortos

- **Tarea:**

- Repasar el algoritmo de caminos más cortos de algoritmos 2.
- Leerlo del libro de Tanenbaum.

- Algoritmos de cálculo de la ruta más corta entre dos nodos.

- Uno de ellos es el algoritmo de **Dijkstra** (1959).

- Dado grafo conexo con costos en los enlaces, y nodo  $n$  en el grafo, obtiene **árbol de caminos más cortos** desde  $n$  hacia todos los demás nodos.
    - El árbol de caminos más cortos se representa con un **mapeo** donde para cada nodo del grafo de la subred asigna su padre (en el árbol de caminos más cortos).
    - Repasar los detalles del algoritmo de Dijkstra visto en algoritmos 2.



## Capa de red: enrutamiento de caminos más cortos

- ¿Qué pasos debería realizar el enrutador para actualizar las tablas de reenvío usando el algoritmo de Dijkstra?

# Capa de red: enrutamiento de caminos más cortos

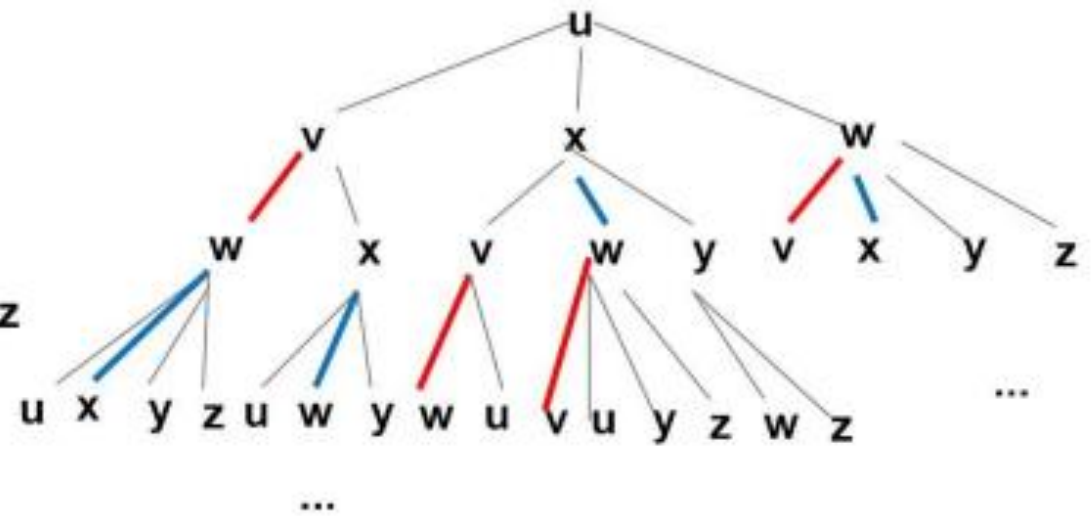
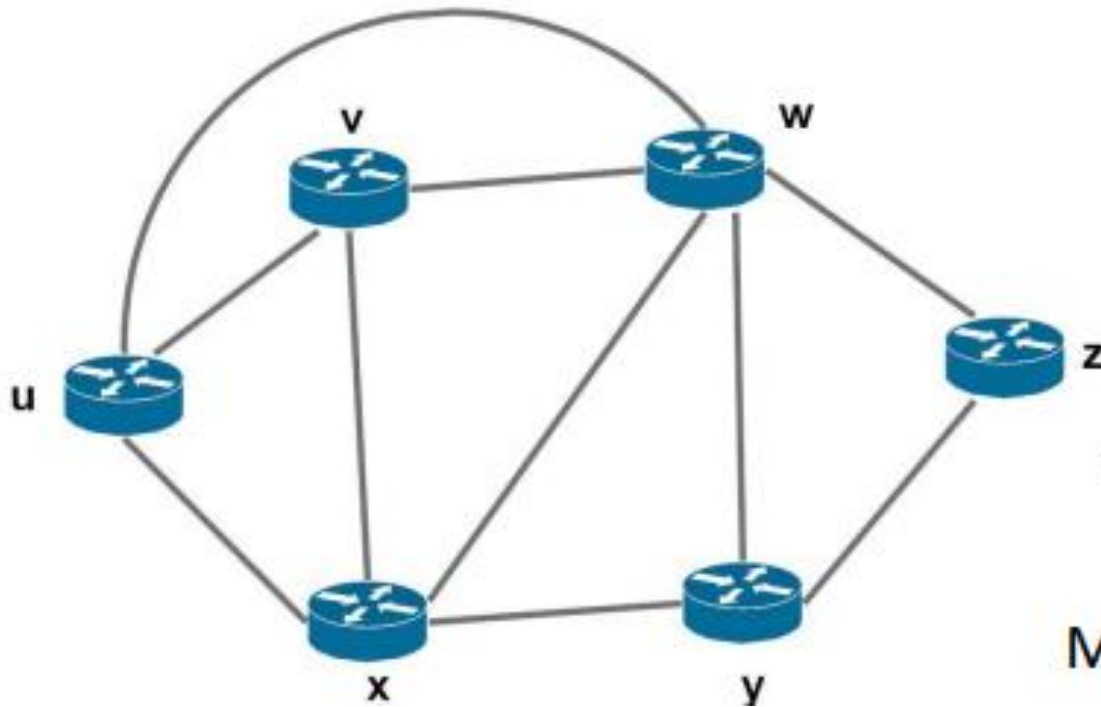
- **Procedimiento para calcular tablas de reenvío en redes de datagramas usando algoritmo de Dijkstra.**
  1. Construir grafo de la subred con costos.
  2. Ingresar grafo de la subred con costos en los enrutadores.
  3. En cada enrutador construir tabla de enrutamiento; para eso:
    - a. Ejecutar algoritmo de Dijkstra en el enrutador
    - b. A partir de árbol de caminos más cortos con raíz en el enrutador obtenido generar la tabla de reenvío del enrutador.

# Capa de red: Inundación

- **Idea de inundación:** para enviar un paquete de un origen  $u$  a un destino  $v$  los caminos usados son aquellos que respetan las siguientes reglas:
  - $u$  manda el mensaje por todas las líneas de salida.
  - Cada paquete que llega a un enrutador distinto de  $v$  se reenvía por cada una de las líneas excepto aquella por la que llegó.

# Capa de red: Inundación

Problema de la idea anterior



Mandar paquete de u a z.

Árbol de envío de paquetes

# Capa de red: Inundación

- **Sugerir ideas para limitar el proceso anterior.**
  - O sea que el problema no suceda,
  - o que sea mitigado enormemente.

# Capa de red: Inundación con registro de paquetes difundidos

- **Implementación:** Para cada enrutador usar **tabla de registro de paquetes difundidos**.



Esta tabla sirve para averiguar si un paquete que llega necesita ser reenviado.

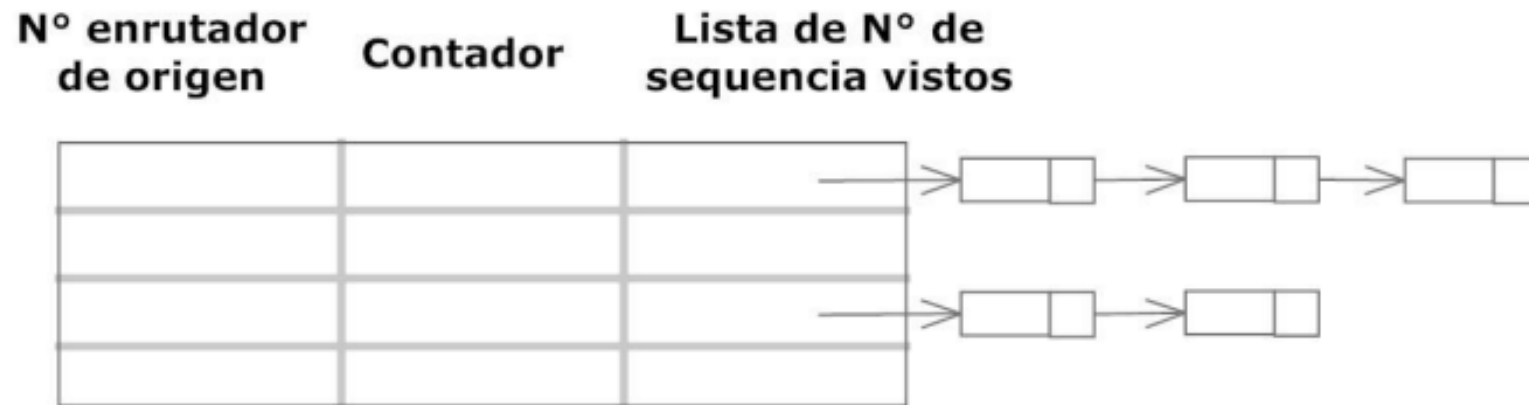
¿indicar qué se hace cuando llega un paquete a un enrutador que tiene esta tabla?

# Capa de red: Inundación con registro de paquetes difundidos

- **Criticar la estructura de datos anterior y sugerir una solución.**
- **Ayuda:** ¿qué pasa cuando enrutadores de origen inundan con más y más paquetes?

# Capa de red: Inundación con registro de paquetes difundidos

**Solución:** agregar Contador. ¿Qué significa?





# Capa de red: algoritmos de enrutamiento

- **Para reflexionar:**

- ¿El algoritmo de enrutamiento de Dijkstra se adapta a cambios en la topología de la red y de tráfico?
- ¿Cuándo debería ejecutarse un algoritmo de enrutamiento que se adapta a esos cambios?

# Capa de red: algoritmos de enrutamiento

- **Idea:** Ejecutar Dijkstra periódicamente y cada vez que cambia topología de la red
  - **Algoritmo de estado de enlace** implementa esta idea.
- ¿Antes de ejecutar Dijkstra en un enrutador qué tarea haría falta hacer?
- ¿Alguna idea cómo se la podría hacer?

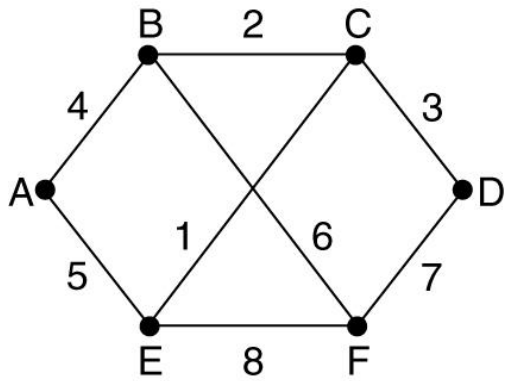
# Capa de red: protocolo de estado de enlace

- **Solución:** distribución de retardos de las líneas de un enrutador a sus vecinos a toda la subred.
  - ¿Qué pasos hacen falta para protocolo de estado de enlace para desglosar esta solución?

# Capa de red: protocolo de estado de enlace

- ¿Cómo averiguar quiénes son los vecinos de enrutador?
- ¿Cómo conocer retardo a los vecinos?
- Cada enrutador construye paquete de estado de enlace (a sus vecinos). **¿Qué información (campos) debe contener?**
- ¿Cómo puedo distribuir paquetes de estado de enlace a toda la subred?

# Enrutamiento de estado de enlace



(a)

Subred

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

(b)

LSP para cada nodo

# Enrutamiento de estado de enlace

- **Distribución confiable de los LSP.**

- ☐ usar **inundación** para distribuir los LSP.
- ☐ se lleva **registro de los paquetes difundidos**.
  - Cada paquete contiene un *número de secuencia* que se incrementa con cada paquete nuevo enviado (desde su enrutador de origen).
  - Los enrutadores llevan el registro de todos los pares *<enrutador de origen, secuencia>* que ven.

# Enrutamiento de estado de enlace

- Cuando llega un LSP a un enrutador, ¿Qué se hace con él?
  - Si es nuevo (nuevo número de secuencia mayor que los anteriores),
    - se reenvía a través de todas las líneas, excepto aquella por la que llegó.
  - Si es un duplicado (número de secuencia mayor visto, pero repetido),
    - se descarta.
  - Si llega un paquete con número de secuencia menor que el mayor visto hasta el momento,
    - se rechaza como **obsoleto** debido a que el enrutador tiene datos más recientes.

# Enrutamiento de estado de enlace

- **Ejercicio:** Supongamos que tenemos una subred con forma de anillo (ciclo) de  $N$  enrutadores (i.e. cada enrutador está conectado con 2 enrutadores vecinos) y que se usa el protocolo de estado de enlace;
  - cada enrutador tiene dos líneas con un vecino: una para enviar y una para recibir;
  - suponiendo que un paquete que atraviesa una línea se cuenta como una carga de 1;
  - ¿cuál es la carga total en la subred para el proceso entero para la actualización de las tablas de enrutamiento?