

# Capítulo 4

## La Capa de Red Enrutamiento de Estado de Enlace

Application
Transport
Network
Link
Physical

# Enrutamiento de estado de enlace

- **Problema:** se quiere un algoritmo de enrutamiento que se adapte a cambios en la topología y el tráfico de la red.
- **Idea:** Si se usa algoritmo de Dijkstra periódicamente y cada vez que cambia la topología en la red,
  - ❑ se va a tener en cuenta la topología y el tráfico.
  - ❑ En cada una de esas veces se necesita conocer la topología de la subred y los retardos de las líneas.
- **Solución:** El algoritmo de **enrutamiento de estado de enlace** implementa esta idea.

# Enrutamiento de estado de enlace

- **Enrutamiento de estado de enlace (Link state routing -LSR)**
  - ❑ En cada enrutador usar **algoritmo de Dijkstra** para encontrar la ruta más corta de un enrutador a los demás enrutadores.
  - ❑ La topología y retardos en las líneas se distribuyen a cada enrutador.
  - ❑ Este algoritmo es valioso porque:
    - **Responde rápido** frente a cambios en la topología de la red.
    - Es **ampliamente usado en Internet** (como parte del protocolo OSPF)
  - ❑ **¿Qué tareas debe hacer un enrutador LSR?**
    1. Descubrir sus vecinos
    2. Medir el costo a cada uno de sus vecinos
    3. Construir un paquete diciendo lo que ha aprendido
    4. Enviar este paquete a todos los demás enrutadores
    5. Computar el camino más corto a cada uno de los otros enrutadores

# Enrutamiento de estado de enlace

- ¿Cómo se puede averiguar quiénes son los vecinos de un enrutador?
  - Se envía **paquete Hello** a cada línea punto a punto
  - Se espera que el enrutador del otro extremo regrese una respuesta indicando quién es.

# Enrutamiento de estado de enlace

- ¿Cómo se puede hacer para que enrutador conozca retardo a sus vecinos?
  - enviar un **paquete ECHO** especial a través de la línea
  - Una vez que llegue al otro extremo, éste debe regresarlo inmediatamente
  - Uso de temporizadores para medir el tiempo.
  - **Método:** Se mide el tiempo de ida y vuelta y se divide por 2.
- **Problema del método:**
  - Asume implícitamente que los retardos son simétricos

# Enrutamiento de estado de enlace

- Cada enrutador construye un **paquete de estado de enlace (LSP)**

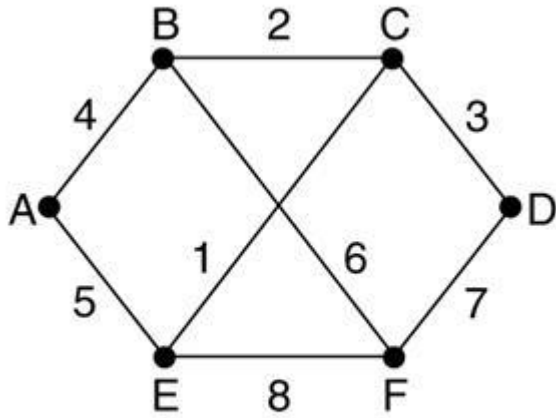
- **¿Qué datos poner en el LSP?**

- Identidad del emisor (para saber de quien se trata)
    - Número de secuencia (para distinguir entre distintos LSP de un enrutador)
    - Edad
    - Lista de <vecino, retardo al vecino>

- **¿Cuándo se pueden construir los LSP?**

- Construirlos a intervalos regulares.
    - Construirlos cuando ocurra un evento significativo, como la caída o la reactivación de la línea o de un vecino, o el cambio apreciable de sus propiedades.

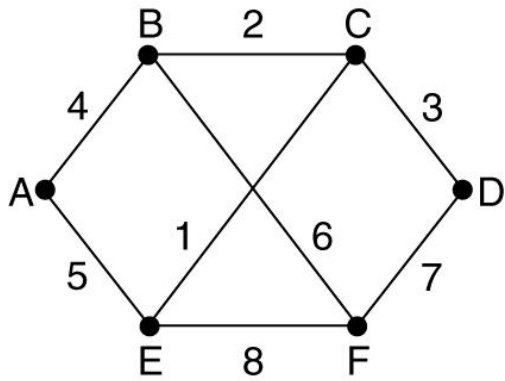
# Enrutamiento de estado de enlace



(a)

¿Qué LSPs se construyen para nodos A y B?

# 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).
  - **Forma de este registro:**
  - 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,

- 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

- Se puede construir la tabla de enrutamiento de un enrutador
  - ❑ una vez que el enrutador ha acumulado un grupo completo de paquetes de estado del enlace
- Primero usando los LSP construir el grafo de la subred completa.
  - ❑ Cada enlace se representa dos veces, una para cada dirección.
  - ❑ Los dos valores pueden promediarse o usarse por separado.
- Se ejecuta el **algoritmo de Dijkstra** para construir la ruta más corta a todos los destinos posibles.
- Con los resultados del mismo se actualiza la tabla de enrutamiento.

# Enrutamiento de estado de enlace

- El algoritmo de inundación de paquetes de estado de enlace tiene algunos problemas que mencionamos a continuación.
- **Problema 1:** Si los números de secuencia vuelven a comenzar, reinará la confusión
- **Solución:** usar un número de secuencia de longitud suficiente para que el problema anterior no suceda. Por ej. de 32 bits.
  - ❑ P.ej. Si un enrutador produce un paquete de estado de enlace cada segundo, llevará 137 años antes de volver a empezar.

# Enrutamiento de estado de enlace

- **Problema 2:** *Si llega a corromperse un número de secuencia y se escribe 65540 en lugar de 4 (un error de un bit), los paquetes 5 a 65540 serán rechazados como obsoletos, dado que se piensa que el número de secuencia actual es 65540.*
- **Solución:** Como protección contra los errores en las líneas enrutador-enrutador, se confirma la recepción de todos los paquetes de estado del enlace.
  - Antes de actualizarse el número de secuencia más grande, el enrutador manda una confirmación de recepción al transmisor y luego espera una respuesta afirmativa o negativa del transmisor.
    - En el primer caso se actualiza el número de secuencia más grande.
    - En el segundo caso se descarta el paquete que se recibió por estar errado.

# Enrutamiento de estado de enlace

- **Problema 3:** *Si llega a caerse un enrutador (de origen), perderá el registro de su número de secuencia. Si comienza nuevamente en 0, se rechazará el siguiente paquete.*
- **Solución:** La información de los enrutadores sólo expira (a lo largo de la red) cuando el enrutador está caído.
  - ❑ **¿Cuándo se puede detectar que un enrutador está caído?**
  - ❑ Cuando se actualicen las tablas de enrutamiento y se manden los paquetes Hello.
  - ❑ **Una vez identificado que un enrutador está caído se hace lo siguiente:**
    - Se propaga la información de este hecho por toda la red.
    - Se hace que la información asociada al enrutador caído expire (paquetes pendientes a enviar, número de secuencia más grande recibido, etc.).
    - Así que cuando ese enrutador vuelva a la vida, puede comenzar con número de secuencia 0.

# Enrutamiento de estado de enlace

- **Problema 4:** ¿Cómo hacer para asegurar que no pueda perderse ningún paquete y sobrevivir durante un período indefinido?
- **Solución:** incluir un campo de **edad en cada paquete**
  - ☐ Disminuir la edad una vez cada segundo.
  - ☐ Los enrutadores también decrementan el campo de edad durante el proceso inicial de inundación.
  - ☐ Se descarta el paquete cuya edad sea 0.

# Enrutamiento de estado de enlace

- Ahora vemos cómo hacer el algoritmo de inundación de paquetes de estado de enlace más eficiente:
  - ❑ Una vez que un paquete de estado del enlace llega a un enrutador para ser inundado, no se encola para transmisión inmediata. En vez de ello, entra en un **búfer de almacenamiento** donde espera un tiempo breve.
  - ❑ Si antes de transmitirlo, llega otro paquete de estado del enlace proveniente del mismo origen, se comparan sus números de secuencia.
    - Si son iguales, se descarta el duplicado.
    - Si son diferentes, se desecha el más viejo.



# Enrutamiento de estado de enlace

- El **buffer de paquetes para un enrutador** contiene una celda por cada paquete de estado de enlace recién llegado, pero aún no procesado por completo.
- Una fila de la tabla del búfer de paquetes de un enrutador contiene:
  - ❑ Origen del paquete, número de secuencia, edad, datos de los estados de enlaces.
  - ❑ **Banderas** que pueden ser:
    - **Banderas de confirmación de recepción**: indica a dónde tiene que enviarse la confirmación de recepción del paquete.
    - **Banderas de envío**: significan que el paquete debe enviarse a través de las líneas indicadas.
    - Si llega un duplicado mientras el original aún está en el búfer, los bits de las banderas tienen que cambiar.

# Enrutamiento de estado de enlace

- Número de secuencia y edad son usados para inundación confiable.
- Paquetes LSP nuevos son confirmados en las líneas que son recibidos y enviados por todas las demás líneas.

Source	Seq.	Age	Send flags			ACK flags			Data
			A	C	F	A	C	F	
A	21	60	0	1	1	1	0	0	
F	21	60	1	1	0	0	0	1	
E	21	59	0	1	0	1	0	1	
C	20	60	1	0	1	0	1	0	
D	21	59	1	0	0	0	1	1	

Búfer para el enrutador B de la filmina 8