



Interconexión de Redes: Repetidores y Bridges

Departamento de Ingeniería Telemática

Universidad Carlos III de Madrid

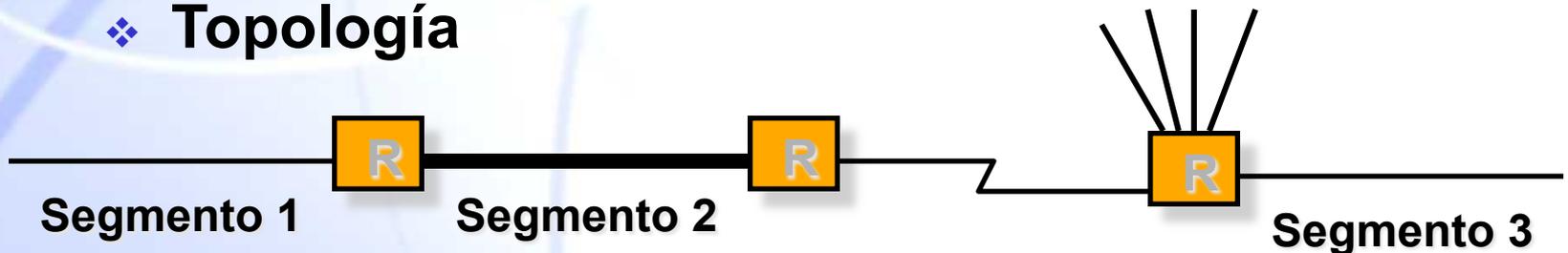
Indice

- ◆ **Interconexión a Nivel Físico**
 - ❖ **Repetidores**

- ◆ **Interconexión a Nivel Enlace**
 - ❖ **Bridges**

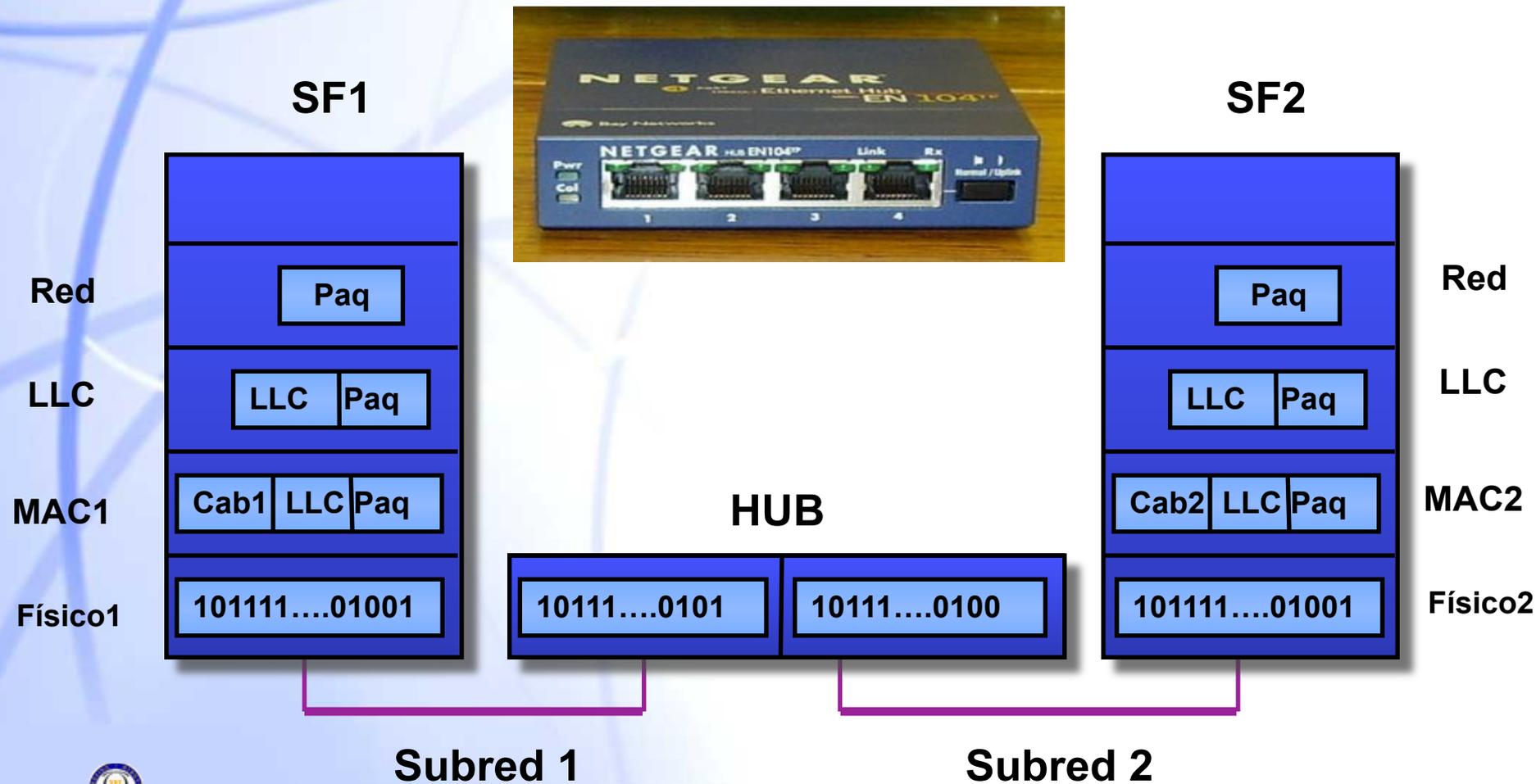
Interconexión a Nivel Físico

- ◆ Repetidor: dispositivo de interconexión de redes a nivel físico.
- ◆ ¿Por qué usar repetidores?
 - ❖ Distancia
 - ❖ Conversión de medio físico
 - ❖ Topología



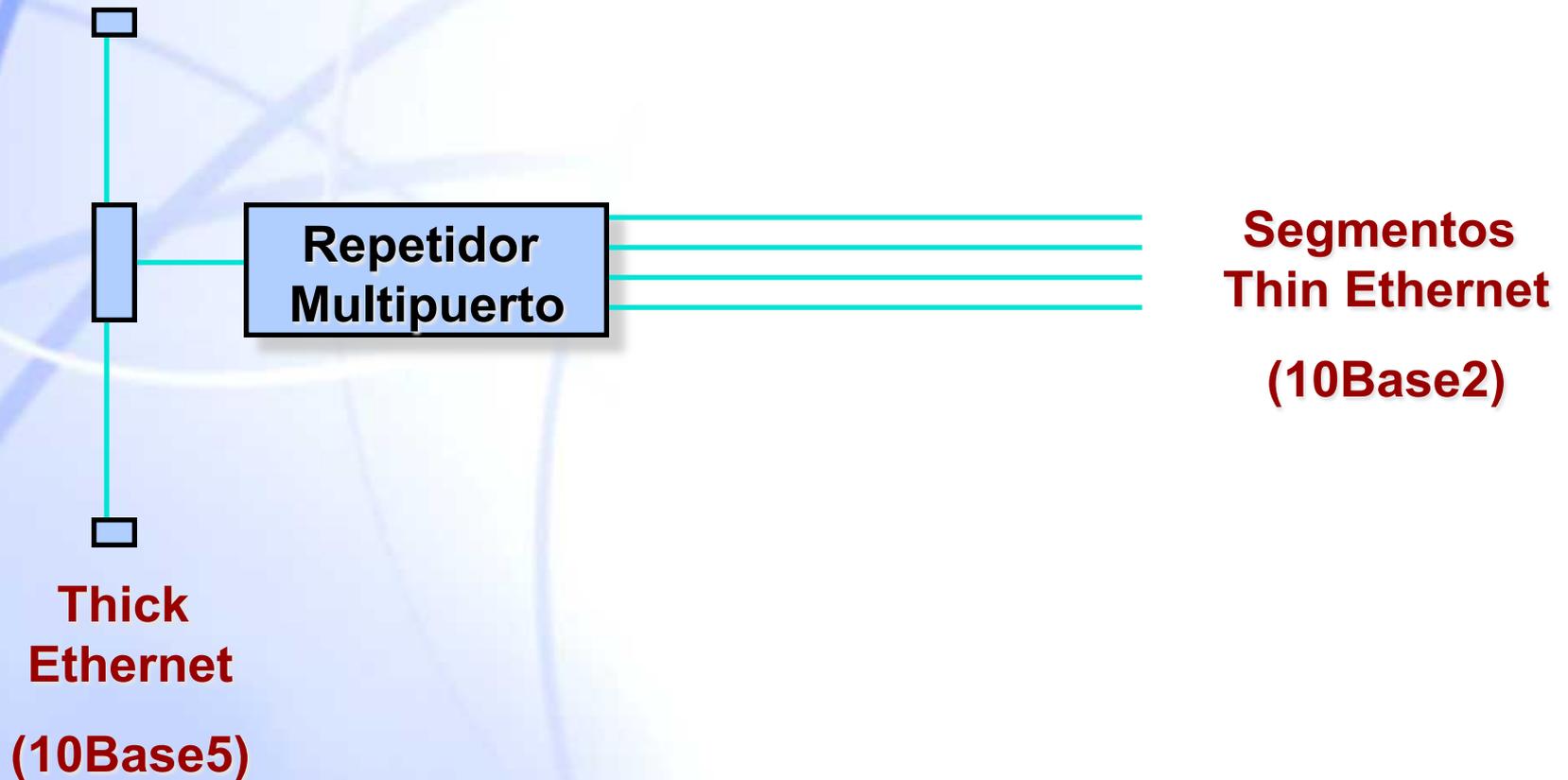
✓ Unico Dominio de Colisión

Repetidores: Modo de Operación



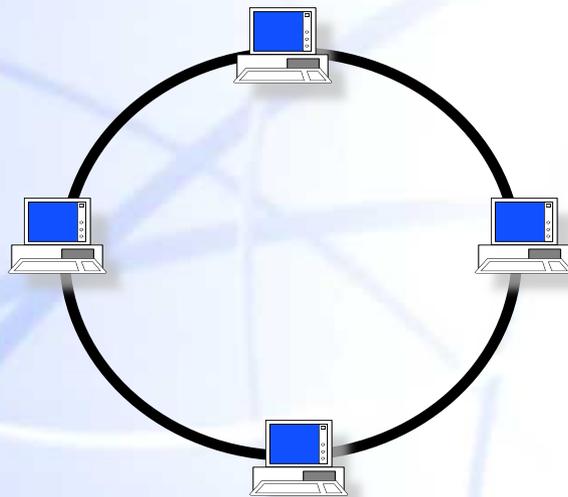
Interconexión a Nivel Físico

◆ Repetidor multipuerto:

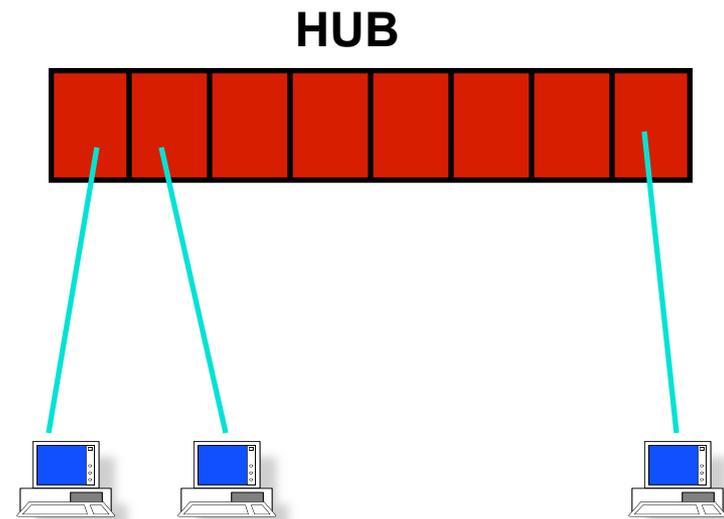


Interconexión a Nivel Físico

◆ Token-ring:



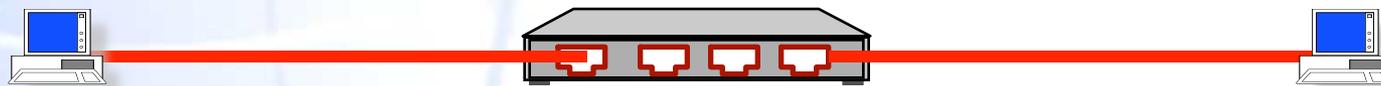
Anillo Lógico



Estrella Fisica

Tipos de Repetidores

- ◆ **Clase I: Latencia $\leq 0,7 \mu\text{seg}$.**
- ◆ **Clase II: Latencia $\leq 0,46 \mu\text{seg}$**
- ◆ **En 100BaseT es un aspecto crítico.**
 - ❖ **1 sólo hub clase I**
 - ❖ **2 hubs clase II**



1 Hubs Clase I/ 200 m



2 Hubs Clase II/ 210 m

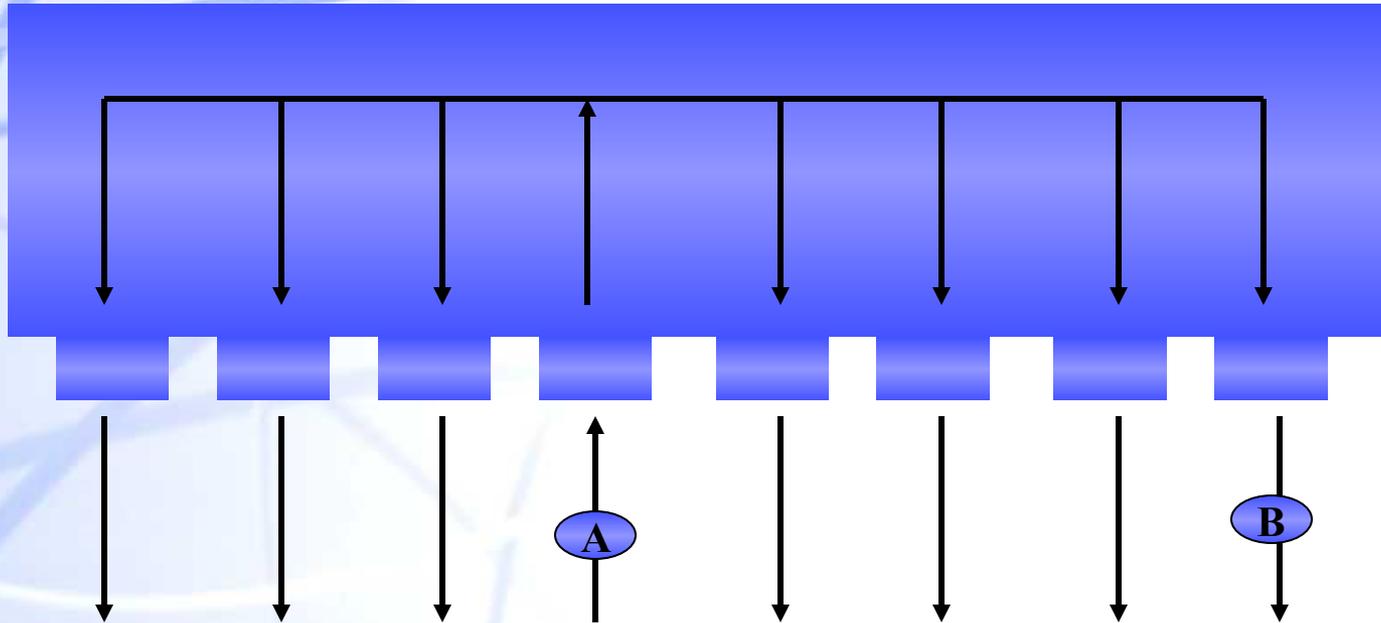


Interconexión a Nivel Físico

- ◆ Problemas:
 - ❖ Restricciones de configuración
 - ❖ Límites de distancia
 - ❖ Separación de Tráfico
 - ❖ Seguridad
 - ❖ Gestión de red

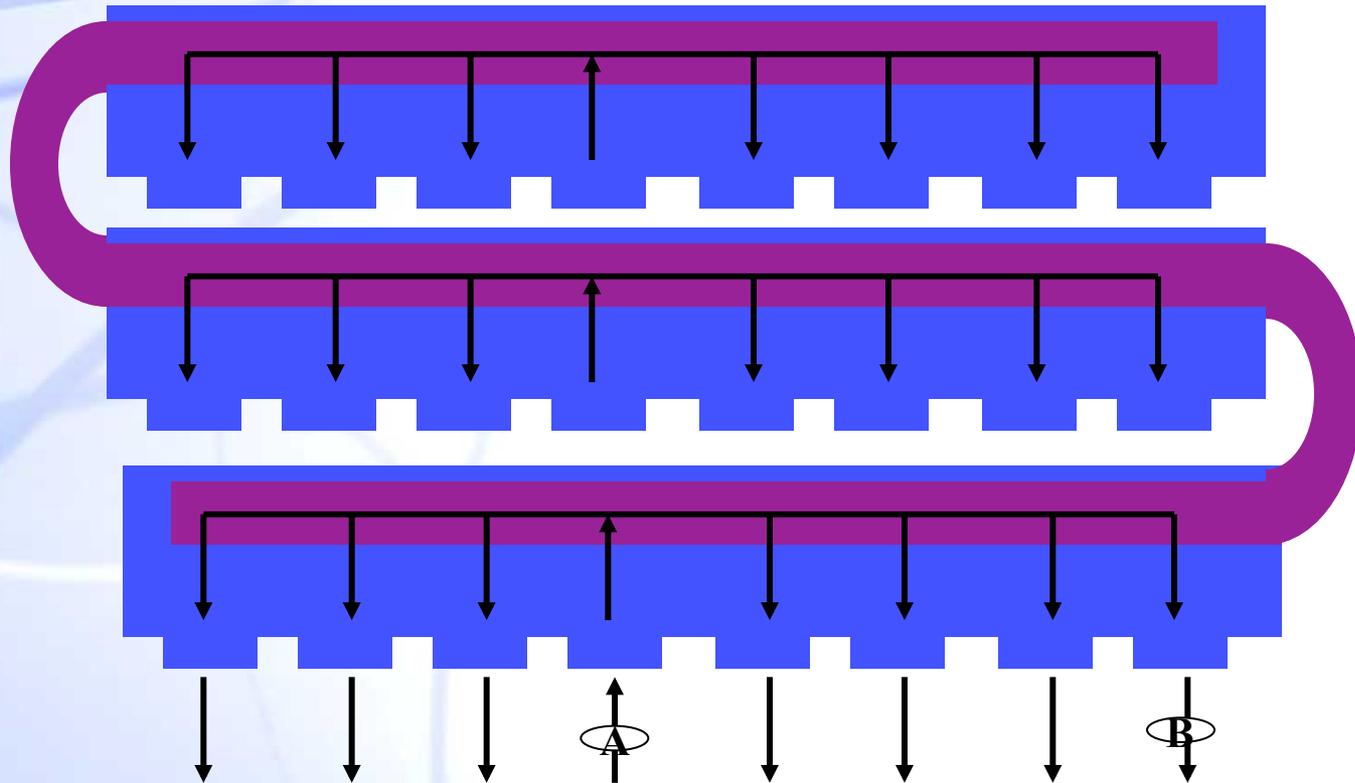


HUB



PROPAGA COLISIONES

HUB en Cascada



PROPAGA COLISIONES

HUBs

- ◆ Los primeros HUBs aparecieron como repetidores para cableado estructurado.
- ◆ Los actuales han sufrido una evolución enorme.
- ◆ Diseñados inicialmente para aprovechar las ventajas del cableado estructurado, tuvieron que dar soporte a la tecnología existente además de estar preparados para el futuro.
- ◆ Permiten la interconexión de diferentes tipos de cableados, con la ventaja de poder utilizar puentes o routers sobre un mismo chasis.



HUBs

- ◆ Los primeros HUBs eran meros repetidores 10BaseT que permitían la conexión de un determinado número de máquinas a la red principal, utilizando para ello una conexión 10Base5 o 10Base2. No incluían funciones de gestión.
- ◆ Aparecen los *HUBs multimedia*, que permiten la conexión a diferentes medios físicos: 10BaseF, Foirl, 10BaseT, 10Base2, 10Base5. Añaden una función mínima de gestión para control del HUB.



HUBs

- ◆ **Posteriormente aparecen los *HUBs* de tercera generación que son capaces de:**
 - ❖ **Mantener sobre una misma máquina un determinado número de redes, de tipo Ethernet, Token Ring y FDDI.**
 - ❖ **Realizar encaminamiento entre cada una de esas redes (utilizando puentes o encaminadores)**
 - ❖ **Utilizar diferentes medios físicos.**
 - ❖ **Gestión mucho más potente, basada en protocolos estándar de gestión, para su control.**



HUBs

- ◆ **La arquitectura de los Hubs de 3^a generación difiere, aunque se pueden considerar tres tipos:**
 - ❖ ***Arquitectura multicanal.*** Emplea varios canales que definen como redes diferentes, tipo Ethernet, Token Ring o FDDI.
 - ❖ ***Arquitectura monocanal.*** De alta velocidad, con comunicación síncrona o asíncrona.
 - ❖ ***Arquitectura mixta.*** Soporta los dos tipos de arquitecturas anteriores.



HUBs

- ◆ **Para proporcionar redes de alta disponibilidad y tolerantes a fallos los Hubs disponen de:**
 - ❖ Fuentes de alimentación redundantes
 - ❖ Módulos de control redundantes
 - ❖ Enlaces redundantes
 - ❖ Port Switching
 - ❖ Concentradores redundantes
 - ❖ Inserción de módulos en caliente
 - ❖ Gestión SNMP

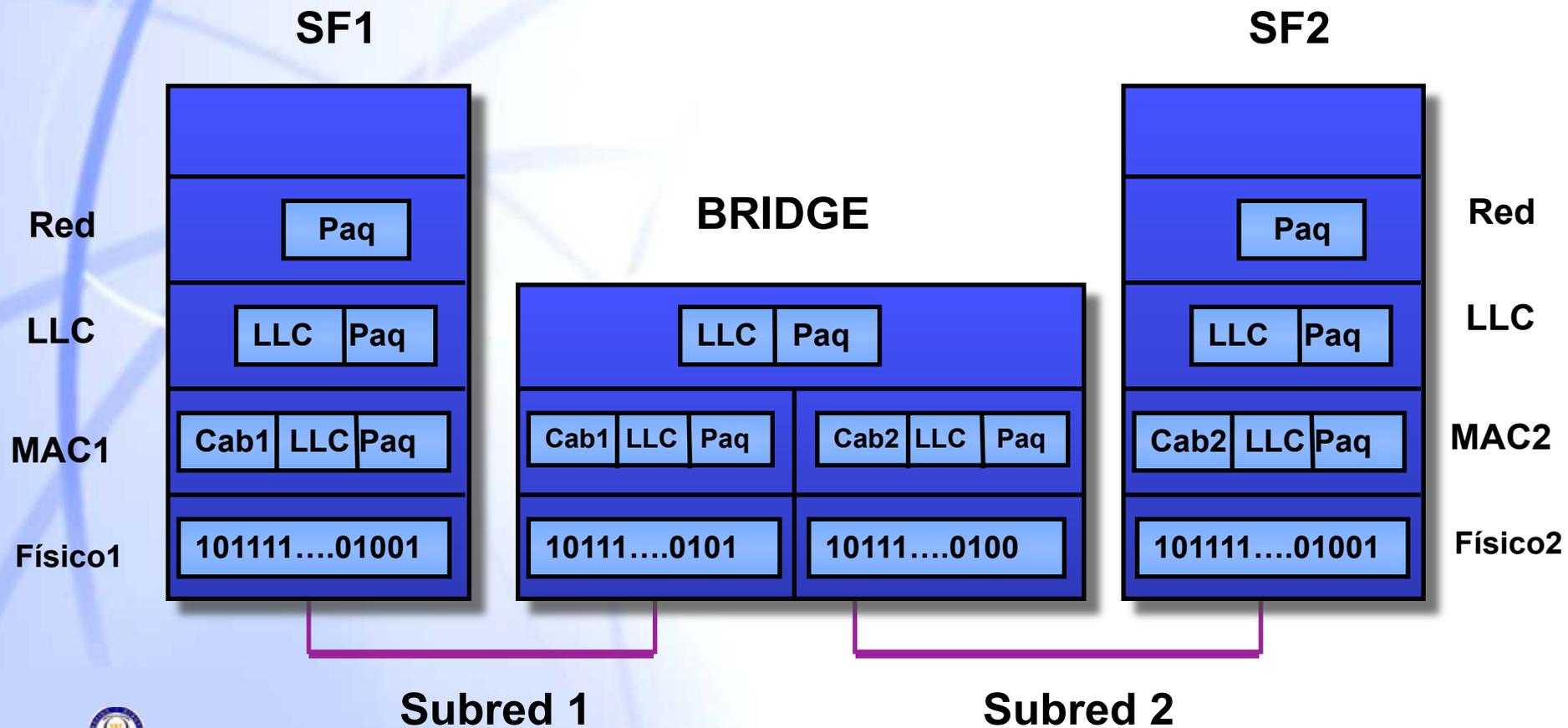
Interconexión a Nivel de Enlace

- ◆ **BRIDGE: dispositivo de interconexión de redes que opera a nivel de enlace. Almacena y reenvía las tramas de unos segmentos de LAN a otros.**



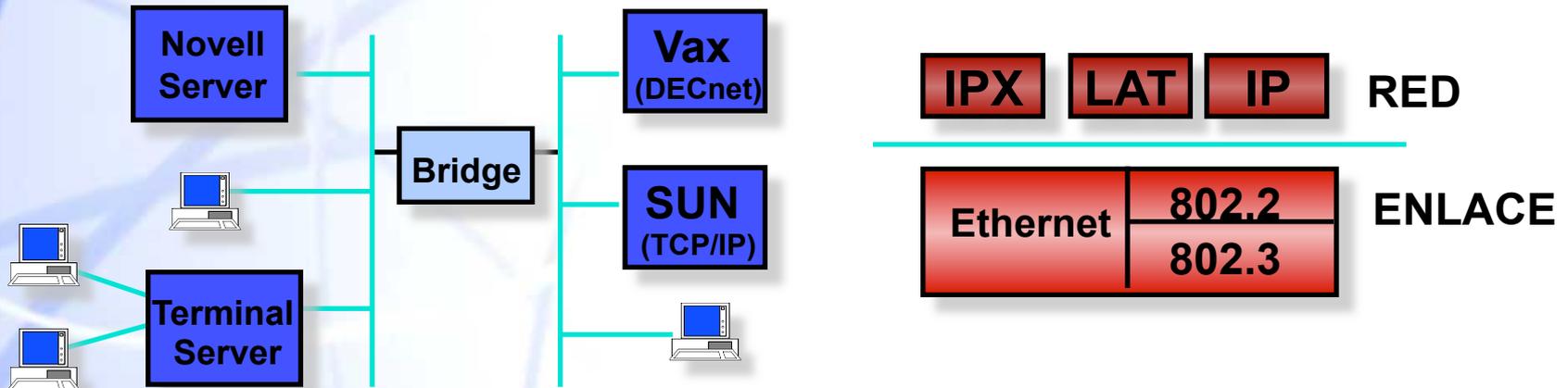
✓ Separan Dominios de Colisión

Bridges: Modo de Operación



Bridges

- ◆ Los Bridges son independientes del protocolo de red:



- ◆ Un bridge puede filtrar tráfico, pero siempre en función de campos del nivel de enlace: *dir. origen, dir. destino, protocolo, tamaño de trama, etc.*

Bridges

- ◆ **Características Típicas:**
 - ❖ **Soportan varios medios físicos: thinnet, thicknet, UTP, etc.**
 - ❖ **Capacidad de filtrado**
 - ❖ **Capacidad de conmutación muy alta**
 - ❖ **Gestionables de forma local o remota. Normalmente con gestión abierta (SNMP)**

Bridges

- ◆ **Utilidad:**
 - ❖ **Interconexión de subredes diferentes**
 - ❖ **Interconexión de LANs distantes**
 - ❖ **Separación de tráfico**
 - ❖ **Aumento de fiabilidad**
 - ❖ **Seguridad**



Bridges entre LANs IEEE 802

- ◆ **Problemática:**
 - ❖ **Diferentes formatos de trama**
 - ❖ **Diferentes velocidades de transmisión**
 - ❖ **Diferentes tamaños máximos de trama**



Bridges entre LANs IEEE 802

Parameters assumed:
 802.3: 1518-byte frames
 802.4: 8191-byte frames
 802.5: 5000-byte frames

		Destination LAN		
		802.3 (CSMA/CD)	802.4 (Token bus)	802.5 (TokenRing)
Source LAN	802.3		1,4	1,2,4,8
	802.4	1,5,9,8,10	9	1,2,3,8,9,10
	802.5	1,2,5,6,7,10	1,2,3,6,7	6,7

Actions:

1. Reformat the frame and compute new checksum.
2. Reverse the bit order.
3. Copy the priority, meaningful or not.
4. Generate a fictitious priority.
5. Discard priority.
6. Drain the ring (somehow).
7. Set A and C bits (by lying).
8. Worry about congestion (fast LAN to slow LAN).
9. Worry about token handoff ACK being delayed or impossible.
10. Panic if frame is too long for destination LAN.

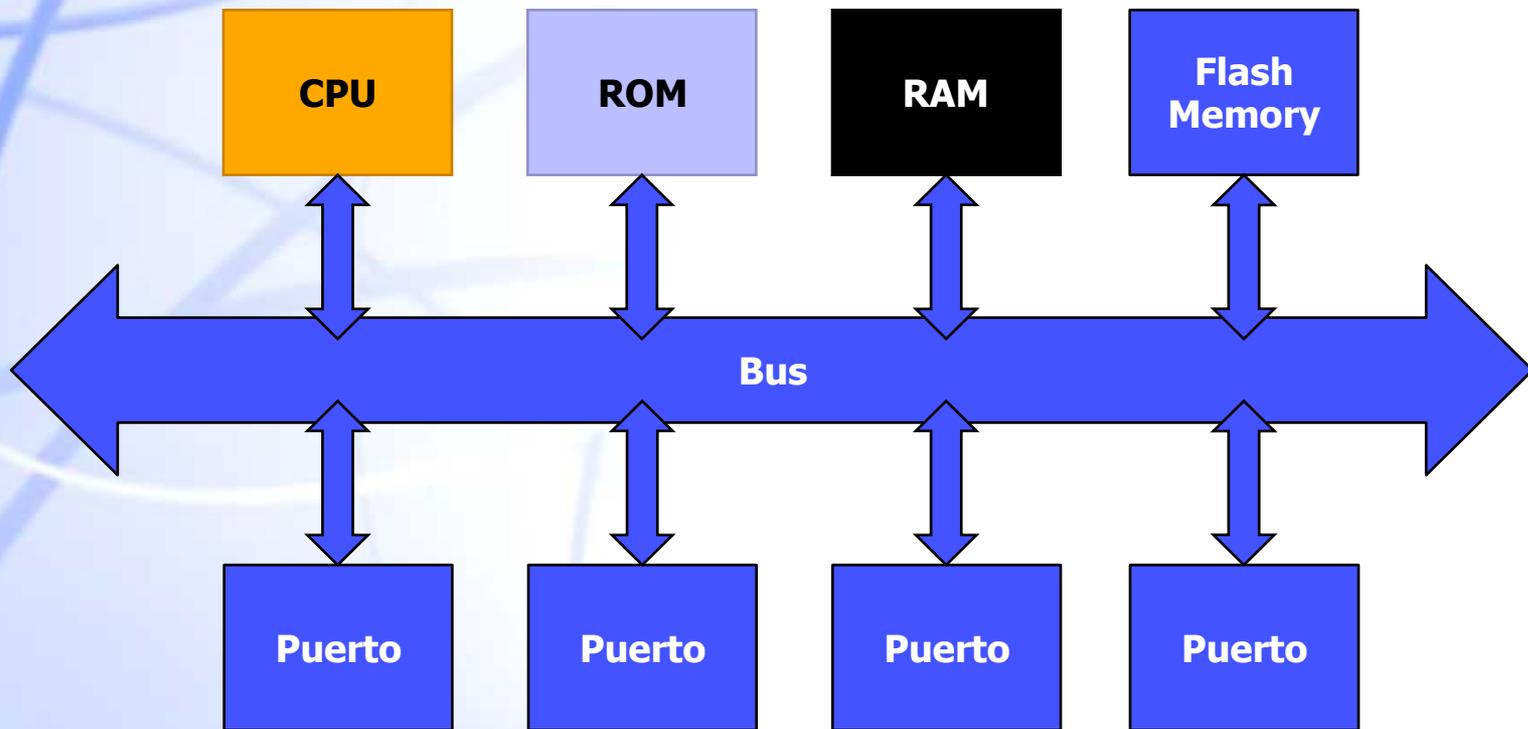


Bridges Transparentes

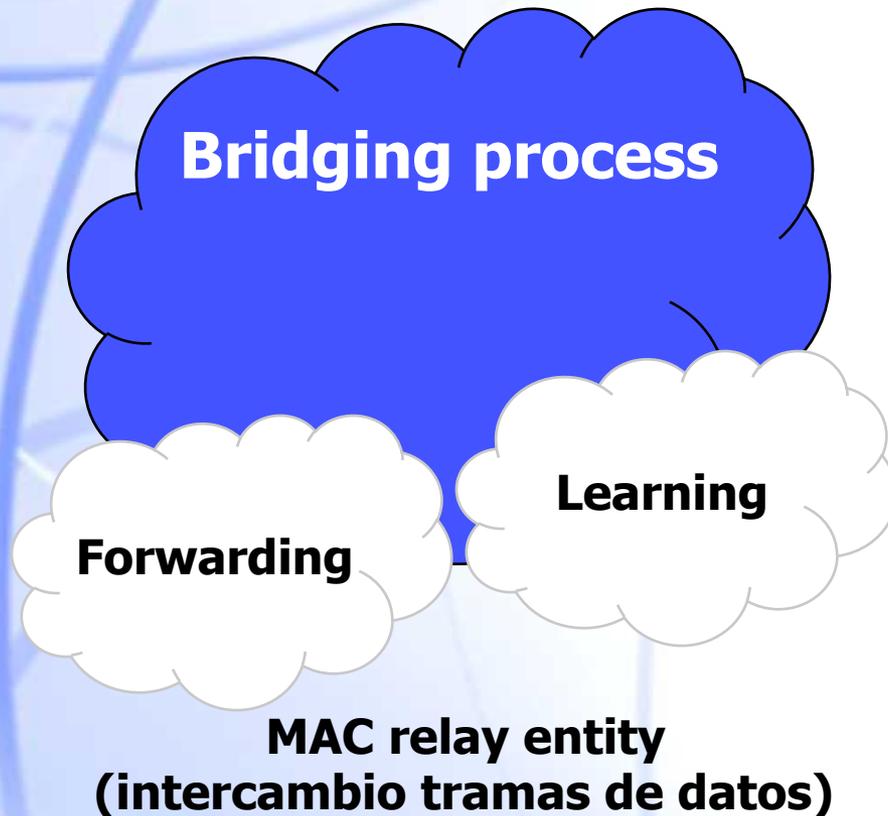
- ◆ No requieren ningún tipo de configuración (PLUG AND PLAY).
- ◆ Funciones soportadas
 - ❖ Reenvío de tramas básico: **funcionalidad mínima**
 - ✓ **Singlecast**: basado en dirección destino
 - ✓ **Broadcast**
 - ❖ Learning Bridge
 - ❖ Spanning tree Bridge
- ◆ Estandarizado en **IEEE 802.1D**



Arquitectura física

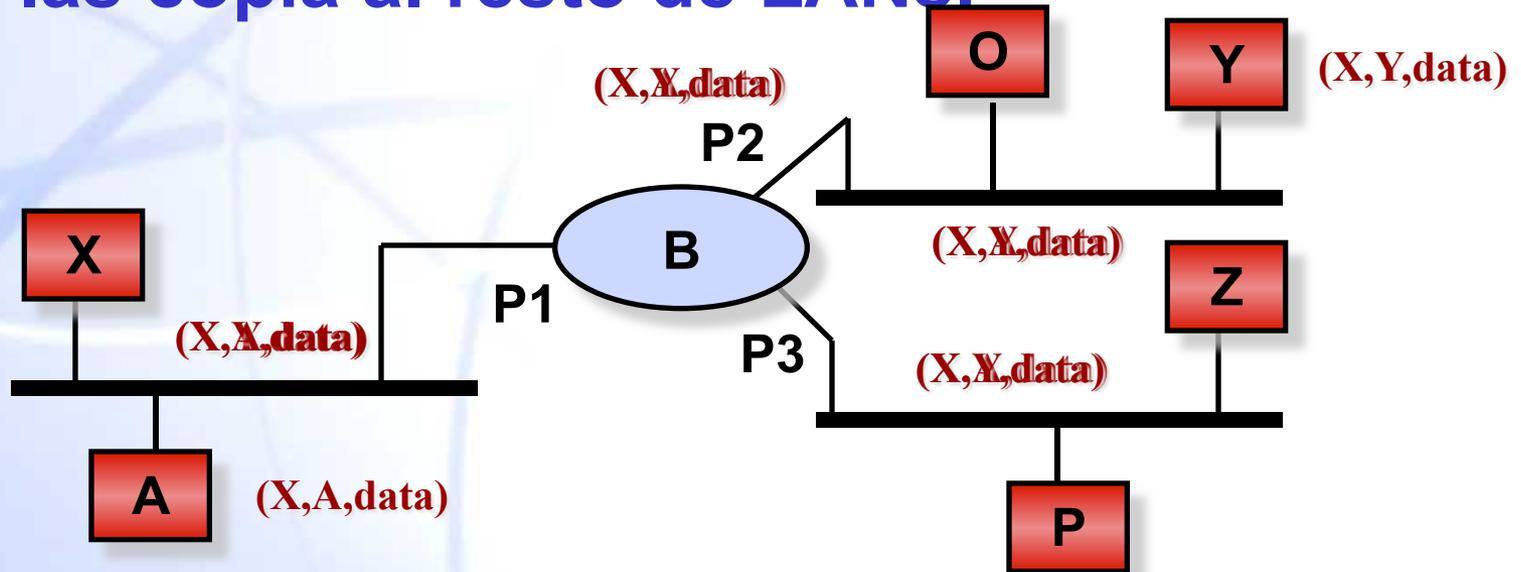


Arquitectura lógica



Bridges Transparentes

- ◆ Bridge Básico: escucha en modo “promiscuo” todas las tramas en cada LAN y las copia al resto de LANs.



☞ **INEFICIENTE !!!**

Bridges Transparentes

- ◆ Bridge con Autoaprendizaje:
 - ❖ escucha en modo “promiscuo” en todas sus interfaces.
 - ❖ Por cada trama recibida, se almacena en una tabla (cache) la dirección origen y el puerto por el que llego.

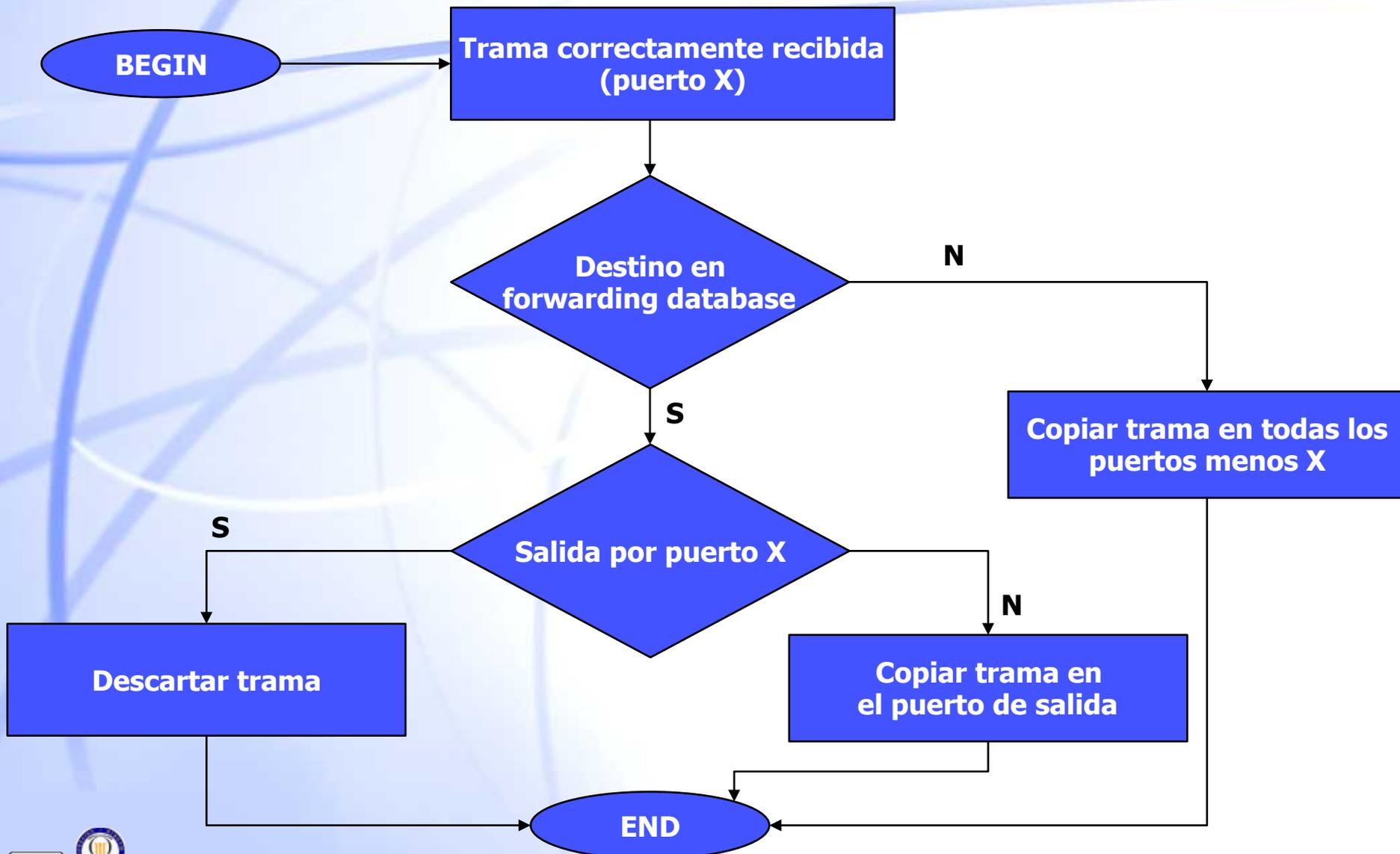


Bridges Transparentes

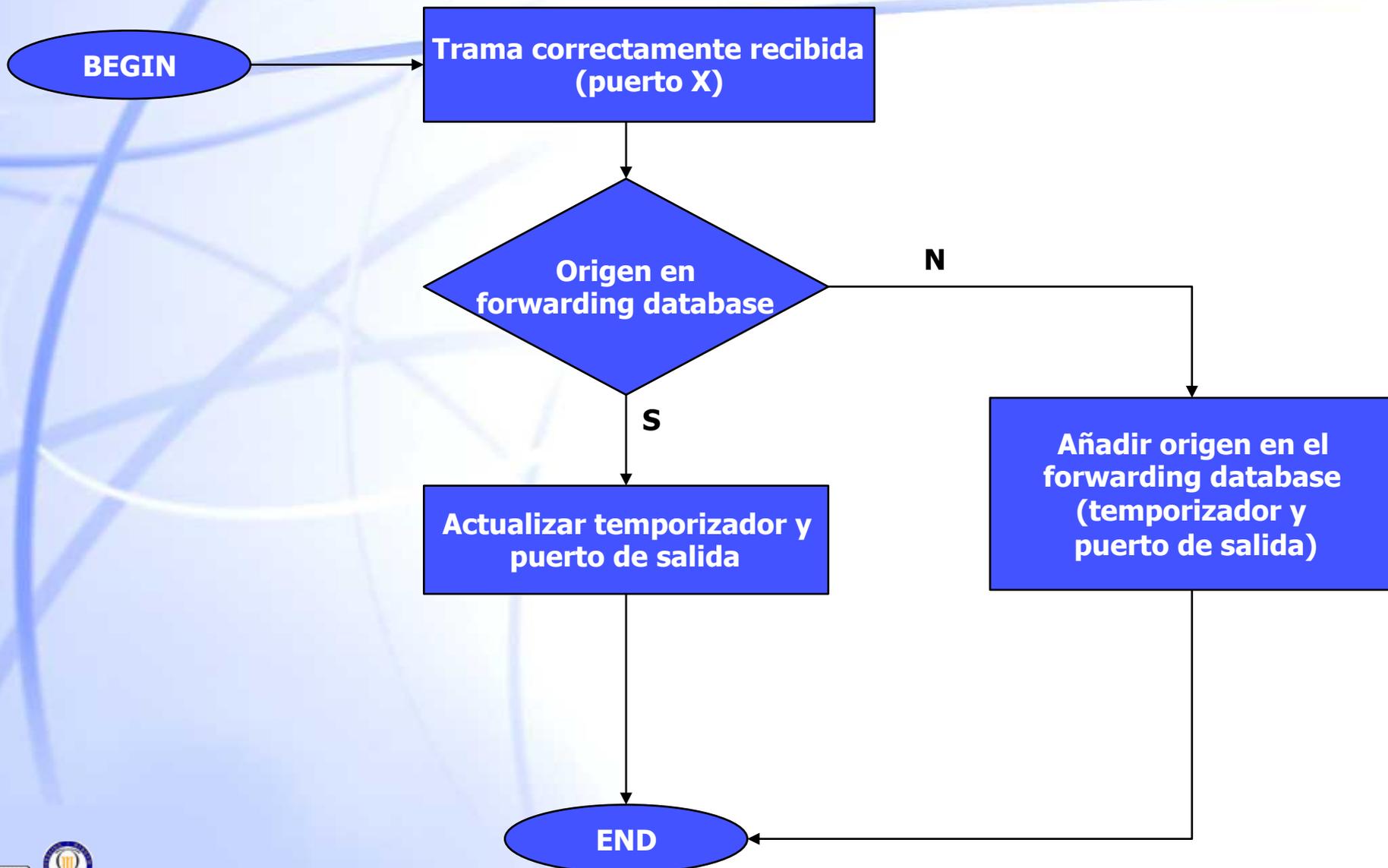
- ◆ **Cuando llega una trama, se consulta si la dirección destino esta en la tabla:**
 - ❖ **SI: se envía a través del interface que indica la tabla (si es el mismo por el que llegó, se descarta)**
 - ❖ **NO: se envía por todos los interfaces salvo por el que llegó.**
- ◆ **Las entradas en la tabla se borran transcurrido un período de tiempo (aging time) si no se actualizan.**



Forwarding process



Learning process



Filtering (forwarding) database

◆ Formato entrada

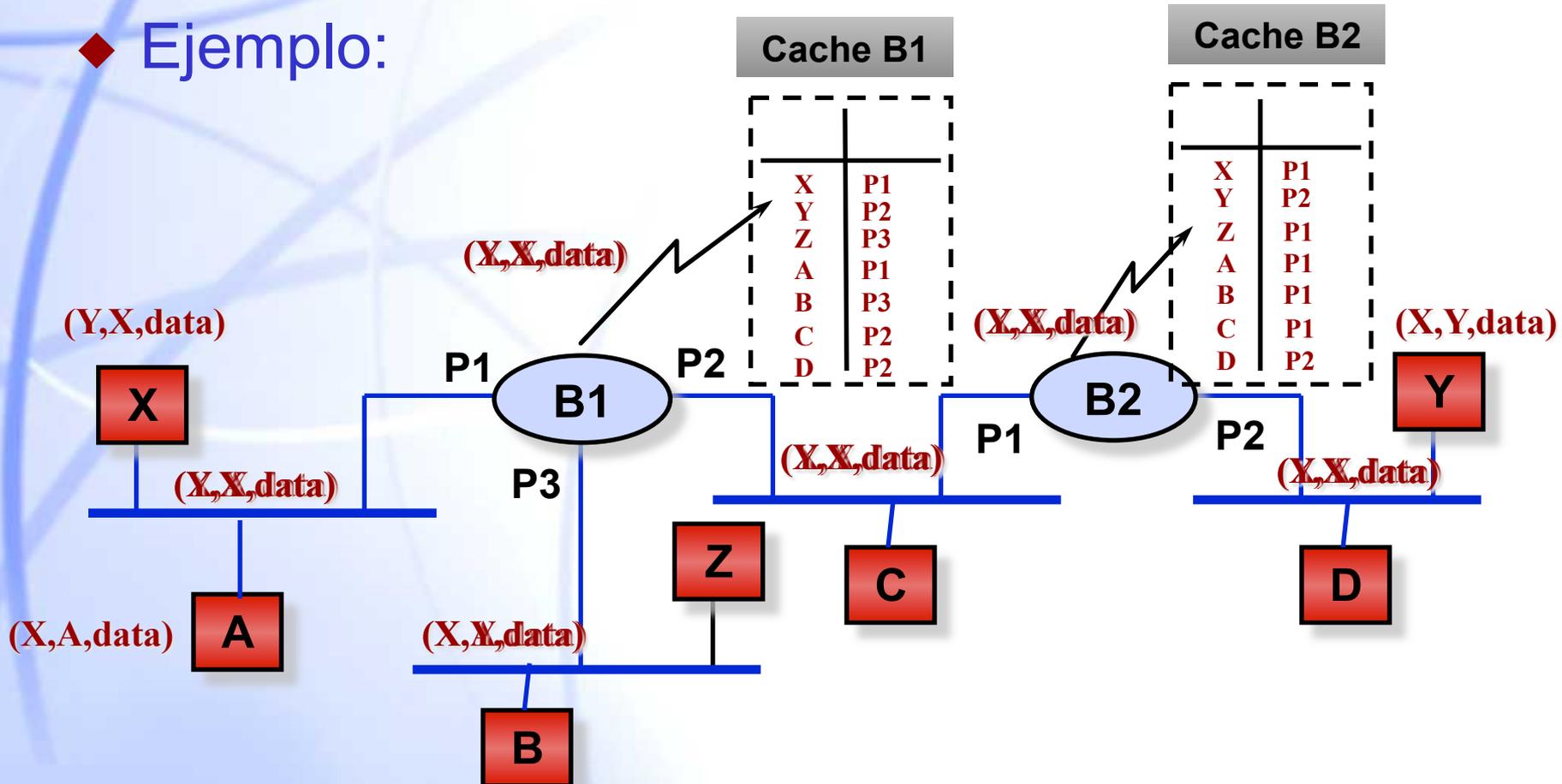
- ❖ MAC
- ❖ Puerto destino
- ❖ Ageing time (temporizador, default 300s)
- ❖ Estado del puerto (Spanning Tree)
- ❖ Tipo
 - ✓ Dinámica (learning process, 1024-65535)
 - ✓ Estática

◆ Implementado con memoria de contenido direccionable (Content-addressable memory, CAM)



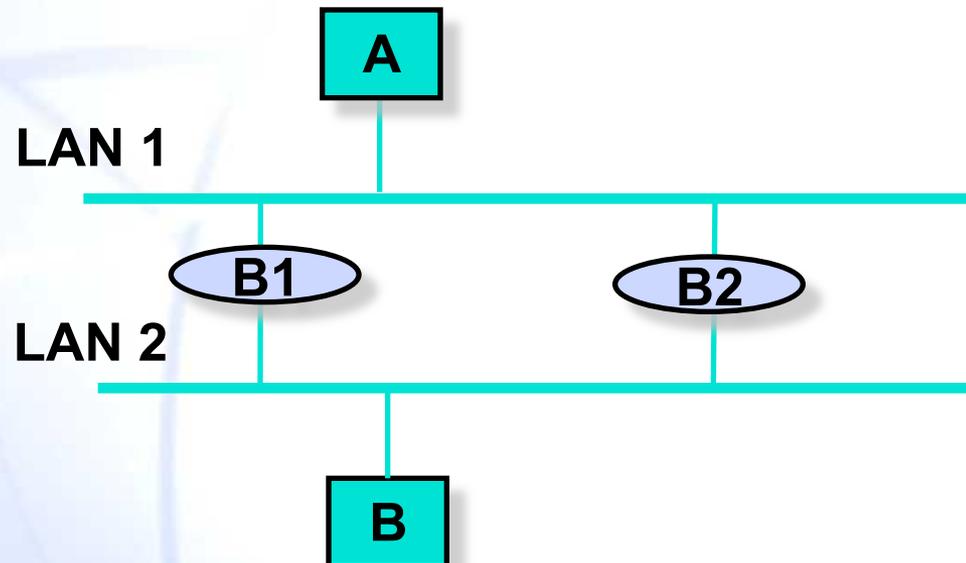
Bridges Transparentes

◆ Ejemplo:



Bridges Transparentes

- ◆ Problema: sólo funcionan en topologías en árbol (sin bucles).
- ◆ Información incongruente en B1 y B2



- ◆ Broadcast Storm

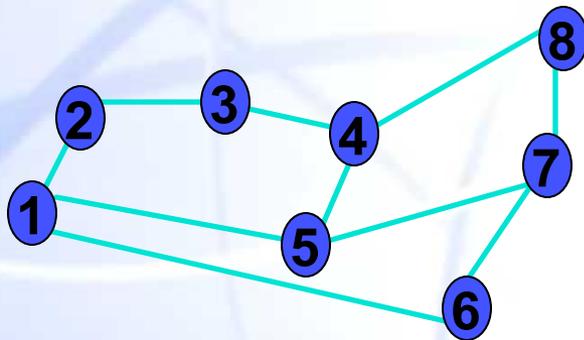
Bridges Transparentes

- ◆ Bridges con Spanning Tree:
 - ❖ Utilizan un algoritmo (Spanning Tree) para evitar los problemas de los “learning bridges” en topologías con bucles.
- ◆ Características:
 - ❖ Topología arbitraria
 - ❖ Plug-and-play
 - ❖ Bridges Transparentes
- ◆ Pasos fundamentales:
- ◆ Elección bridge raíz

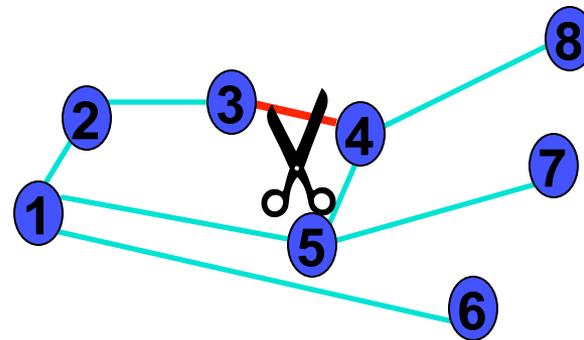


Bridges Transparentes

- ◆ Todos los Bridges se ponen de acuerdo en utilizar una topología sin bucles *Spanning Tree*, sobre la cual funcionan como bridges transparentes.



Ejemplo de Red



Spanning Tree con raíz en 1

- ◆ Buscamos el Spanning Tree (ST) de coste mínimo.

STP se puede resumir en un poema

*I think that I shall never see
a graph more lovely than a tree.
A tree whose crucial property
is loop-free connectivity.
A tree that must be sure to span
so packets can reach every LAN.
First, the root must be selected.
By ID, it is elected.
Least-cost paths from root are traced.
In the tree, these paths are placed.
A mesh is made by folks like me,
then bridges find a spanning tree.*

Radia Perlman



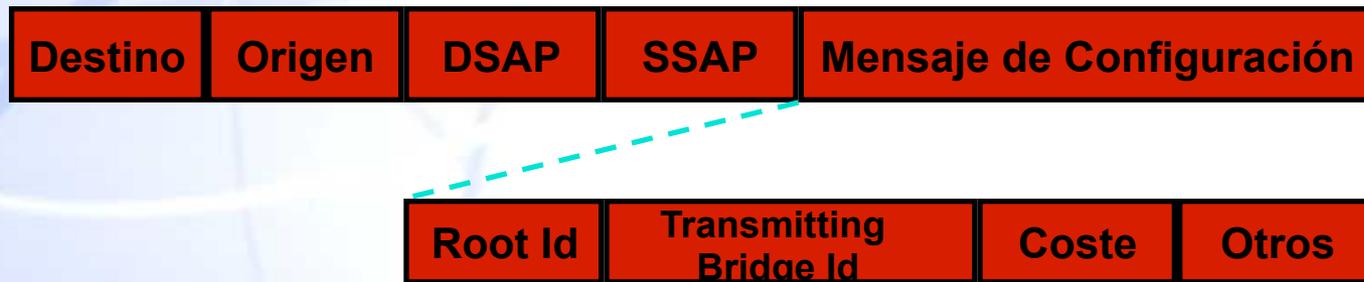
Bridges Transparentes

- ◆ **Algoritmo de construcción del ST:**
 - ❖ Se elige la raíz
 - ❖ Se calcula la distancia mínima desde cada Bridge (B) a la raíz.
 - ❖ En cada LAN se selecciona un B designado (el más próximo a la raíz), que se utilizará para enviar tramas desde la LAN a la raíz.
 - ❖ En cada B se elige un puerto para enviar tramas a la raíz.
 - ❖ Una vez elegido el ST, cada bridge utiliza sus puertos incluidos en él para enviar tramas. El resto quedan inactivas.



Spanning Tree Bridges

- ◆ Para construir el ST los Bridges intercambian mensajes de configuración denominados Configuration Bridge Protocol Data Units (CBPDU).



- ◆ Cada Bridge posee un identificador único
- ◆ Existe una dirección multicast a la cual se envían los mensajes de configuración.

Estado de los puertos

◆ Forwarding

- ❖ Puerto raíz y puerto designado
- ❖ Las tramas recibidas por puertos en estado *forwarding* son procesadas y reenviadas según el *forwarding process*

◆ Blocking

- ❖ Los demás puertos
- ❖ Las tramas recibidas por puertos en estado *blocking* son rechazadas



Spanning Tree Bridges

- ◆ Al inicio, cada Bridge supone que él es la raíz y envía sus CBPDUs de acuerdo a ello.
- ◆ Cada Bridge recibe CBPDUs por todos sus puertos. Selecciona en cada uno de ellos el “mejor” según el algoritmo siguiente.

Una CBPDU C1 es mejor que otra C2 si:

- ❖ $C1.rootID < C2.rootID$
- ❖ $C1.cost < C2.cost$
- ❖ $C1.transmID < C2.transmID$
- ❖ $C1.port < C2.port$



Spanning Tree Bridges

◆ Ejemplo de CBPDUs:

C1

Root ID	Cost	Trans. Id
---------	------	-----------

A.	29	15	35
B.	35	80	39
C.	35	15	80

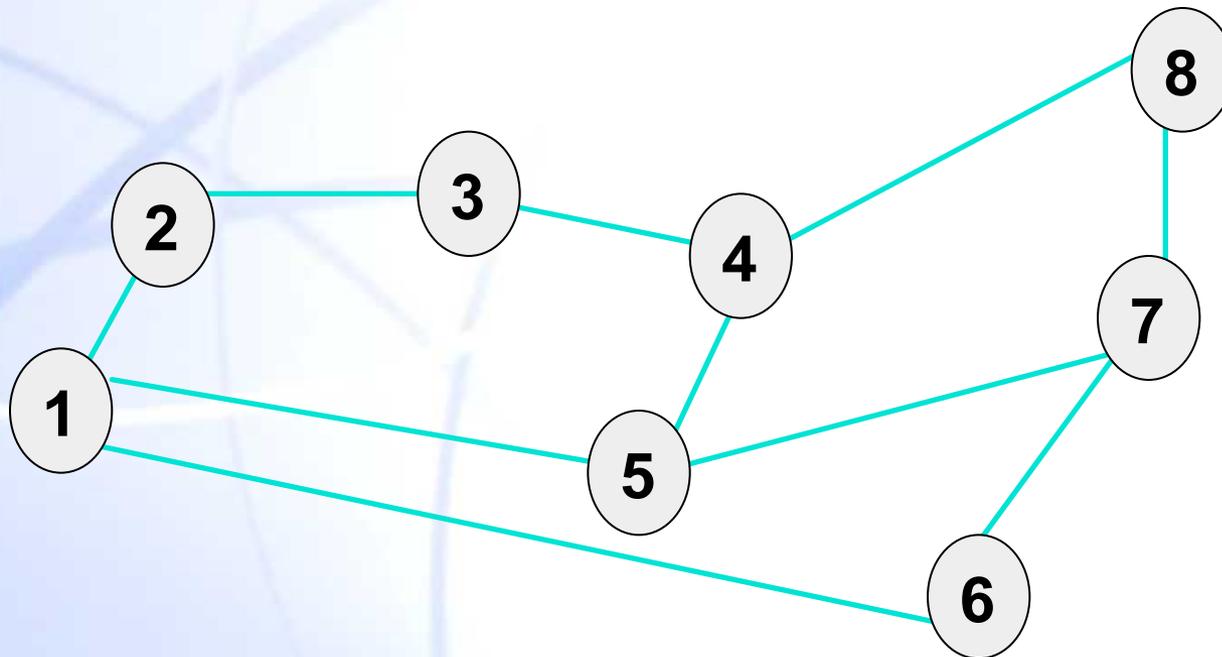
C2

Root ID	Cost	Trans. Id
---------	------	-----------

29	12	32
35	80	40
35	18	38

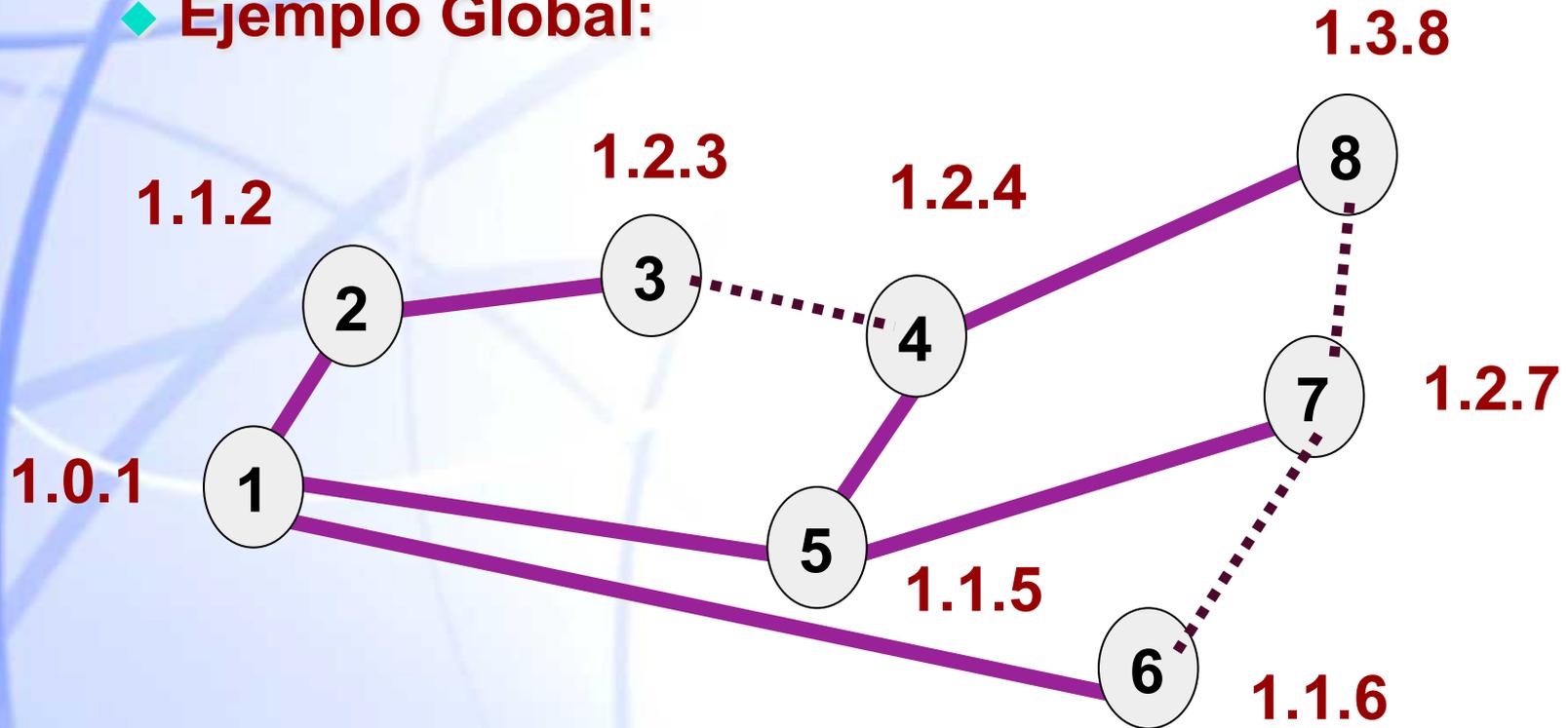
Spanning Tree Bridges

◆ Ejemplo Global:



Spanning Tree Bridges

◆ Ejemplo Global:



Spanning Tree Bridges

- ◆ **Funcionamiento en caso de fallos.**
 - ❖ los mensajes tienen “fecha de caducidad”
 - ❖ cada “hello time” la raíz transmite sus CBPDUs
 - ❖ al recibirlas, los bridges transmiten sus CBPDUs a través de los puertos en que son Bridges designados
- ◆ **El algoritmo de construcción del ST se ejecuta cada vez que:**
 - ❖ se recibe una CBPDU mejor o más joven
 - ❖ si una de las CBPDUs almacenadas se pasa de fecha

